

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS  
INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA



***Proyecto Fin de Carrera***

**Aurebot, sistema didáctico para el  
aprendizaje de electrónica basado en una  
plataforma robótica**

(Aurebot, system for electronics learning based  
on a robotic platform)

Para acceder al Título de

**INGENIERO TÉCNICO DE TELECOMUNICACIÓN**

Autor: Luis E. Sainz Nieto

Julio - 2013



E.T.S. DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

## INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN

### CALIFICACIÓN DEL PROYECTO FIN DE CARRERA

**Realizado por: Luis E. Sainz Nieto**

**Director del PFC: Olga M<sup>a</sup> Conde Portilla, Luis Rodríguez Cobo**

**Título:** “Aurebot, sistema didáctico para el aprendizaje de electrónica basado en plataforma robótica”

**Title:** “Aurebot, a system for electronics learning based on a robotic platform”

**Presentado a examen el día: 31/07/2013**

para acceder al Título de

### INGENIERO TÉCNICO DE TELECOMUNICACIÓN, ESPECIALIDAD EN SISTEMAS ELECTRÓNICOS

Composición del Tribunal:

Presidente: Cobo García, Adolfo

Secretario: Rodríguez Cobo, Luis

Vocal: Quintela Incera, Antonio

Este Tribunal ha resuelto otorgar la calificación de: .....

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del PFC

Vº Bº del Subdirector

Proyecto Fin de Carrera Nº

AGRADECIMIENTOS: trabajo en parte realizado con los medios aportados por los proyectos DA2TOI (Proyecto FIS2010-19860) y TFS (Proyecto TEC2010-20224-C02-02).

## Agradecimientos

---

A mi familia, por todo y en especial por permitir tomarme esto con “relativa” tranquilidad y darme la oportunidad de perseguir otros proyectos.

A Luis, porque esto es, al menos, tan suyo como mío.

A mis amigos: astronautas frustrados, políticos, ingenieros/as con y sin trabajo, personas “normales” y resto, por la diversión durante el viaje.

Al Grupo de Ingeniería Fotónica por permitirme realizar este proyecto y la ayuda prestada.

*The most likely way for the world to be destroyed, most experts agree, is by  
accident.  
That's where we come in. We're computer professionals. We cause accidents.*  
**Nathaniel Borenstein**

## Índice

---

Índice .....	1
1. Introducción .....	1
1.1. Origen y motivación .....	1
1.2. Evolución .....	2
1.3. Objetivos .....	4
1.4. Organización del documento .....	4
2. Metodología: ¿Un robot dando clase? .....	6
2.1. Uso de la robótica en la docencia.....	6
2.1.1. Cantabria .....	7
2.1.2. España .....	10
2.1.2.1. Competiciones a nivel local .....	10
2.1.2.1.1. Concurso nacional de robots “Ciutat de Mataró” .....	10
2.1.2.1.2. Robolid .....	10
2.1.2.1.3. Alcabot .....	10
2.1.2.2. Competiciones nacionales .....	10
2.1.2.2.1. Hispabot .....	10
2.1.2.2.2. Ceabot .....	11
2.1.2.2.3. Cosmobot .....	11
2.2. Plataformas robóticas para docencia .....	11
2.2.1. Lego Mindstorms .....	11
2.2.2. Moway .....	12
2.2.3. Skybot .....	13
2.2.4. Boeobot.....	13
2.2.5. Arduino y Raspberry Pi.....	14
3. El AureBot .....	16
3.1. Funcionalidad .....	16
3.2. Hardware.....	17
3.2.1. Aureboard, descripción general .....	17
3.2.2. Chasis .....	17
3.2.3. Unidad de proceso.....	19
3.2.4. Jumpers de configuración .....	20
3.2.5. Interfaces con el exterior .....	21
3.2.5.1. Diodos LED .....	21

3.2.5.2.	Puertos de conexión .....	22
3.2.5.3.	Puerto serie .....	23
3.2.5.4.	U.S.B.....	23
3.2.5.5.	L.C.D.....	23
3.2.5.6.	Placa de prototipos .....	24
3.2.6.	Motorización .....	24
3.2.7.	Alimentación .....	25
3.2.7.1.	Alimentación por baterías .....	26
3.3.	Software .....	28
3.3.1.	Bootloader.....	28
3.3.2.	Interfaz serie.....	29
3.3.3.	Interfaz “gráfica” (L.C.D.) .....	29
3.3.4.	Librerías desarrolladas .....	30
3.3.4.1.	Librería principal.....	30
3.3.4.2.	Librería “2 leds” .....	30
3.3.4.3.	Librería “barra leds” .....	30
3.3.4.4.	Librería “bumpers” .....	31
3.3.4.5.	Librería “CNY70” .....	31
3.3.4.6.	Librería “flex_lcd” .....	31
3.3.4.7.	Librería “motores” .....	31
3.3.4.8.	Librería “servos” .....	32
3.3.4.9.	Librerías adicionales .....	32
3.3.4.9.1.	Librería protocolo I2C .....	32
3.3.4.9.2.	Librería protocolo 1-Wire .....	32
3.3.4.9.3.	Librería dispositivo DS 18B20 .....	33
3.3.4.9.4.	Librería dispositivo DS 1307 .....	33
3.3.4.9.5.	Librería dispositivo DS 1624 .....	33
3.3.4.9.6.	Librería WiiMotion Plus .....	33
3.3.5.	Interfaz gráfica (Aurebot GUI).....	34
4.	Resultados y valoración .....	37
5.	Conclusiones y líneas de futuro .....	40
5.1.	Baterías .....	40
5.2.	Programación .....	40
5.3.	Plataforma más actual .....	41
6.	Bibliografía .....	42

7.	Anexos.....	43
7.1.	Presupuesto construcción robots V1 – Jornadas 2008.....	43
	Componentes electrónicos .....	43
	Resto de materiales .....	44
7.2.	Esquema de la Aureboard .....	45
7.3.	Esquema del cargador de baterías de dos celdas LiPo .....	46
7.4.	Resumen básico de C para PICs .....	47
7.5.	Valoración actividad AureBot 2011-12 .....	51

## 1. Introducción

---

### 1.1. Origen y motivación

---

Al iniciar nuestros estudios de la Ingeniería de Telecomunicaciones, algunos compañeros nos dimos cuenta de que la mayor parte de los conocimientos que se impartían -especialmente en los primeros años- eran meramente teóricos.

Aunque toda esta teoría fuera la base de todo lo que fuéramos a aprender y utilizar después, es muy fácil perder la perspectiva y el interés si no se conoce qué aplicación práctica tienen.

Por esto, por coincidencia con nuestros hobbies y por divulgar entre la gente que esa serie de instrucciones abstractas y algoritmos que veíamos en clase tenían una aplicación muy fácil y muy real, nos embarcamos en la cruzada de organizar unas Jornadas de robótica y electrónica en la Escuela de Ingenieros Industriales y de Telecomunicación de la Universidad de Cantabria.

En este contexto se nos ocurrió que una llamativa manera de cumplir con nuestros objetivos sería diseñar un robot muy simple, con unos componentes muy fáciles de identificar y emplear, con conocimientos técnicos o sin ellos.

¿Por qué un robot? Es una interfaz con el mundo real, algo más atractivo que los elementos individuales de los que se compone. Nosotros estudiamos los componentes electrónicos y fotónicos como simples "bloques" con los que montar algo mayor,, cuya única interfaz es la variación en la pantalla de un multímetro del voltaje o de la corriente.

Un robot proporciona una visión más práctica. En vez de comprobar que cada bloque aislado como una fotorresistencia (LDR) varía su resistencia al cambiar la luz incidente empleando un multímetro, se puede proponer un reto: utilizar los conocimientos de programación adquiridos en otras asignaturas para que el robot, en base a las lecturas de voltaje presentes en un par de LDRs situadas a ambos lados del robot se dirija al lado con más o menos luz.

Con este reto estaremos impulsando el trabajo en equipo y ayudando a los alumnos a aprender a resolver problemas reales. El alumno retendrá los conocimientos adquiridos en este caso durante más tiempo y los comprenderá mejor que en el primero por estar asociados a una experiencia práctica.

Cuatro años después, con los compañeros finalizando los estudios y tres exitosas jornadas a las espaldas surge la problemática de la falta de tiempo para seguir organizándolas, así como la búsqueda de un nuevo objetivo para la plataforma robótica que habíamos desarrollado.

En el Grupo de Ingeniería llevaban tiempo también trabajando con robots, específicamente con la plataforma BoeBot desde 2006, utilizándola con objetivos similares a los nuestros aunque mucho más orientados a la docencia y al empleo de manera práctica de electrónica personalizada. Usando los componentes que se habían visto de manera teórica en clase para solucionar problemas “reales” con la ayuda de la plataforma robótica. Aunque con el problema, común para ambos, de disponer de un bajo número de unidades como para poderlos usar con efectividad.

La colaboración con el Grupo de Ingeniería Fotónica de la UC durante las Jornadas y la coincidencia de gran parte de nuestros objetivos con los del reenfoque de una de sus asignaturas al plan piloto hizo posible la realización de este proyecto: adaptar nuestra plataforma a un objetivo más didáctico y dirigido a un público específico, alumnos del Grado de Ingeniería de tecnologías de Telecomunicaciones.

## 1.2. Evolución

---

El Aurebot se ha desarrollado en paralelo a las Jornadas de Robótica y Electrónica de la Escuela Superior de Ingenieros Industriales y de Telecomunicación de la UC. La repetición de estos eventos durante tres años consecutivos ha permitido la evolución y mejora de la plataforma, para, posteriormente, ser adaptada con fines didácticos.

La primera y más austera versión del Aurebot (Ilustración 1 Aurebot, primera versión) data de 2008, de las I Jornadas de Robótica y Electrónica.

Teniendo en cuenta los escasos recursos económicos de los que se disponía, se plantea una plataforma básica con unas funcionalidades mínimas. Empleando un microcontrolador PIC de la empresa Microchip (que fue obtenido gracias al programa de muestras de dicha empresa), se desarrolló una placa de circuito impreso que combina capacidad de actuación (motores), de sensado y decisión (Microcontrolador). Dicha placa se fabricó y ensambló a mano para ser incorporada a una estructura simple, realizada en madera de contrachapado cedida por una empresa local.

A pesar del esfuerzo realizado para conseguir recursos y minimizar costes, en esta primera convocatoria sólo se pudieron fabricar 10 unidades, puesto que el coste estaba limitado por los diversos componentes electrónicos de la placa, así como por los motores y sensores. Estos primeros robots disponían de un par de bumpers como sensores de contacto y unos sensores ópticos reflexivos con el objeto de que pudieran realizarse carreras del tipo “sigue líneas”.

En 2010, durante las III Jornadas de Robótica y con su posible adaptación a una plataforma docente, se diseñó un nuevo chasis para el robot, cambiando madera por plástico. Asimismo, se incorporan numerosos cambios en electrónica y funcionamiento como el paso de pilas a baterías recargables, el cambio de ruedas a orugas, mejoras en

la programación y comunicación con el exterior, más sensores y un acceso más directo y simple a los puertos del microcontrolador.

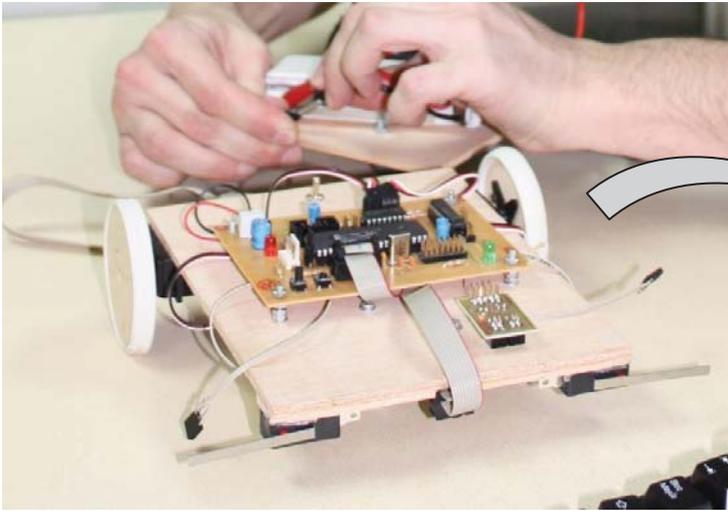


Ilustración 1 Aurebot, primera versión

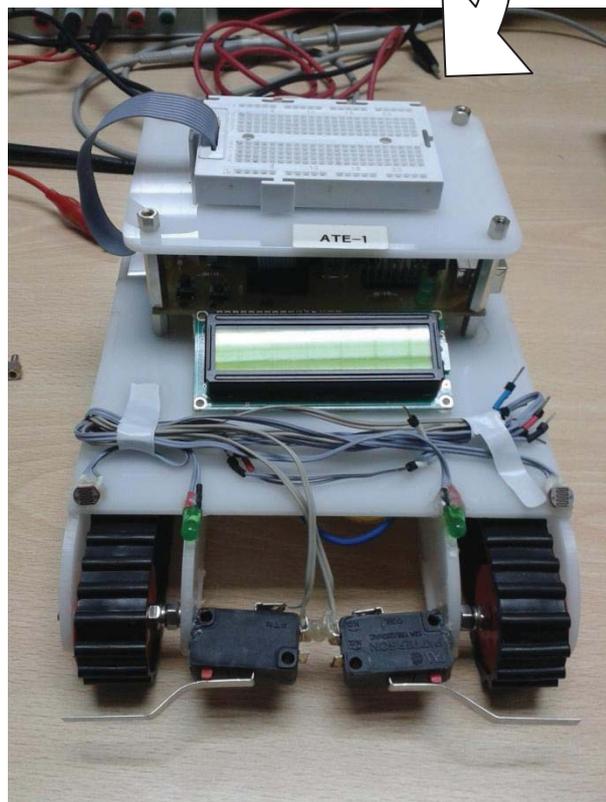


Ilustración 2 Aurebot, versión final

Ambas versiones se pudieron desarrollar y mejorar gracias a la ayuda y a los fondos cedidos por la Escuela Técnica y Superior de Ingenieros Industriales y de Telecomunicación, por la Delegación de alumnos de este mismo centro y por el Grupo de Ingeniería Fotónica de la Universidad durante las tres ediciones de las Jornadas.

### 1.3.Objetivos

---

El objetivo del presente proyecto es adaptar las plataformas originales de los Aurebots a una plataforma robótica didáctica para el aprendizaje en la asignatura de Dispositivos electrónicos y fotónicos.

Como objetivos específicos de la plataforma se han establecido: el desarrollo específico de esta con un claro objetivo docente -simple de usar y mantener y bien documentada- que favorezca el aprendizaje experiencial frente al teórico -permitiendo la resolución de problemas simples mediante la utilización de los componentes electrónicos trabajados en la asignatura y las capacidades del robot- que fomente el trabajo colaborativo y la competitividad –los robots se emplearan en grupos, los alumnos de cada grupo colaboraran para solucionar el problema propuesto y a la vez competirán entre si para ver quien consigue la mejor solución.

Además el proyecto tendrá algunos requerimientos adicionales: la reutilización del máximo posible de las plataformas Aurebot 1.0 con el objeto de ahorrar costes, desarrollar tanto la parte hardware (el robot en sí) como el software asociado (librerías e interfaz gráfica de control), simplificar el uso de la plataforma mediante la creación de nuevas librerías de software que abstraigan las funciones de C para microcontroladores a instrucciones mas amigables para el alumno, una mejora de la estética que le aporte un aspecto más profesional, mejorar la interfaz con el exterior desechando el puerto serie RS232 tradicional en pos del más habitual USB y añadiendo mas diodos LED controlables por el usuario y un panel LCD de texto, construir un mínimo de nueve unidades (8 equipos que estarán siempre en funcionamiento y un robot adicional para las ocasiones en las que uno de los otro falle), facilitar el acceso a las salidas y entradas analógico-digitales del microprocesador añadiendo además la posibilidad de incorporar electrónica personalizada por el usuario mediante una placa de prototipos y la disponibilidad de serie de unos sensores básicos (resistencias LDR y sensores de contacto mediante interruptores de final de carrera) incluidos de serie en el robot.

### 1.4.Organización del documento

---

El presente documento se estructurará en cinco capítulos de la siguiente manera:

- El primer capítulo es la introducción, un resumen del origen, la motivación y la evolución de este proyecto.
- A continuación, en el segundo capítulo, se defenderá el uso de un robot como plataforma didáctica destacando qué ventajas tiene, así como un resumen del

uso de la robótica para docencia en la actualidad y un repaso a algunas de las plataformas didácticas más utilizadas.

- En el capítulo tercero se realizará una descripción detallada de la plataforma, primero en la parte hardware y posteriormente, en la de software.
- Un cuarto capítulo lo conformarán los resultados y la valoración del proyecto funcionando en un entorno real y las impresiones de los alumnos que lo utilizan.
- El último capítulo contendrá las conclusiones y líneas de futuro, donde se evaluará el proyecto y se plantearán mejoras se le podrían realizar al sistema.

## 2. Metodología: ¿Un robot dando clase?

---

El objetivo de toda asignatura es transmitir al alumno una serie de conocimientos, y lo ideal es maximizar esta transferencia. Para esto se necesita generar interés en el alumno, que éste vea el objetivo de esos conocimientos, ya sea global, como base a todo los conocimientos que adquirirá posteriormente, o localmente, como conocimientos que por sí mismos ya tienen una aplicación real.

Un robot es una herramienta con la que el alumno puede aplicar fácilmente conocimientos de matemáticas, de física, de lógica y desarrollar habilidades como el pensamiento abstracto, la manipulación gruesa y fina y el trabajo en equipo al mismo tiempo que observa inmediatamente como estos conocimientos y habilidades tienen un efecto práctico en el mundo real

El robot, aún siendo relativamente simple comparado con los robots a nivel industrial, es un fiel reflejo a nivel de bloques de estas.

Si tenemos en cuenta el nivel del alumno, estudiante de primer curso de grado -por regla general sin muchos conocimientos técnicos- sí que es una maquina complicada, lo que le obliga a cooperar con otros, a subdividir tareas para conseguir el objetivo propuesto y competir con el resto para cumplirlo de la mejor manera.

Estas subtarear que bien podrían ser: calcular la velocidad de desplazamiento, diseñar un circuito que polarice correctamente un componente, adaptar una señal a los niveles aceptados por la plataforma, programar un algoritmo para dirigirse a la zona con mayor o menor luminosidad, requieren la aplicación de toda una serie de conocimientos que el alumno habrá adquirido en la parte teórica de la asignatura, o de otras asignaturas por lo que este las aprenderá y comprenderá mucho mejor.

Además, no todos los conocimientos necesarios tienen porque haberse estudiado, lo que les obliga también a aprender a buscar y encontrar que necesitan para resolver el problema. Dicho en otras palabras: a pensar por sí mismos.

### 2.1. Uso de la robótica en la docencia

---

En los periodos de la educación obligatoria (primaria y secundaria) no hay mucho margen para el aprendizaje práctico. No obstante, la asignatura de Tecnología que se cursa durante la educación secundaria permite introducir algunas herramientas de docencia no tradicionales como la electrónica aplicada y la robótica, que aunque conllevan un gran compromiso y carga de trabajo para el profesorado suelen compensar con creces.

En el ámbito universitario la libertad de incorporar este tipo de experiencias en las asignaturas es mayor. No obstante, como en los casos anteriores la carga adicional que supone para el profesorado y el miedo a incorporar nuevos métodos los relegan a un segundo o tercer plano, a algo en ocasiones meramente puntual.

Los principales problemas para la implantación de este tipo de experiencias son: el desinterés del profesorado que prefiere seguir el itinerario curricular de un libro de texto, los recursos económicos disponibles ya que las plataformas son costosas y no permiten equipar un aula con suficientes equipos por una cifra económica razonable, la falta de tiempo del profesorado para la preparación de las clases y de las plataformas y la falta de formación del equipo docente en áreas en las que no tienen porqué estar formados, pero que es básica si van a ejercer como guías a los alumnos (algorítmica, electrónica básica, etc.).

A pesar de todo ello, la docencia está recurriendo poco a poco al uso de la robótica en los últimos años, lo que lleva a pensar que esta evolución seguirá acrecentándose en el futuro.

En este capítulo se realizará una revisión del estado actual del empleo de plataformas robóticas en la docencia desde el ámbito regional hasta el nacional.

### 2.1.1. Cantabria

---

A pesar de que el uso de la robótica en el ámbito de la enseñanza en nuestra comunidad no es una herramienta recurrente, existen casos de su utilización que nos sirven para ejemplificar su importancia en el ámbito de la docencia.

Un ejemplo claro es el de los componentes de la Asociación de Profesores de Tecnología de Cantabria, quienes llevan varios años utilizando la robótica como herramienta didáctica en sus institutos, empleando diversas plataformas con el predominio claro de Lego, desde el más antiguo Lego Mindstorms RCX hasta el más actual NXT, y luchando por mantener una docencia práctica frente a la teoría pura.

En 2011, dentro del marco de la “OCEANS’11 IEEE Santander Conference [1]” se organizó un concurso de salvamento con minirobots en el que participaron los institutos IES Jose Hierro de San Vicente de la Barquera y el IES Valentin Turienzo de Colindres.



Ilustración 3 Alumnos y su tutor preparando los robots para la prueba

En 2013, por primera vez, la A.P.T. organizó CantabroBots, un concurso dirigido a estudiantes de Secundaria y Bachillerato en el que se compitió y se hicieron demostraciones en amplia variedad de categorías todas ellas relacionadas con la robótica educativa.

Contaron con la participación de más de 260 participantes de nueve centros de la región que habían desarrollado sus soluciones a los problemas propuestos en las clases de tecnología.



Ilustración 4 Robots diseñados por alumnos de secundaria Ilustración 5 Competición de sumo robot con Lego

En el ámbito universitario la robótica se ha empleado muy poco en el ámbito docente. En la práctica se ha empleado únicamente en una asignatura “Laboratorio de componentes electrónicos y fotónicos”, predecesora de la actual “Dispositivos electrónicos y fotónicos”, en la que se empleaban Home Boebots

En algunas asignaturas impartidas por el Departamento de Matemática Aplicada y Ciencias de la Computación se han empleado algunos robots (Lego NXT, Moways y últimamente Arduinos) pero como mera anécdota, aunque permitiendo en ocasiones la realización de trabajos voluntarios con estas plataformas.

Es destacable además que en los últimos años han tenido lugar diferentes cursos de robótica aplicada:

Las tres ediciones de las Jornadas de robótica y electrónica en la Universidad de Cantabria realizadas por AURE, en las que se pudo ver, trabajar y competir con los Aurebots y los Lego NXT. Recibió tanto a alumnos de la propia universidad como a profesores y alumnos de educación secundaria.



Ilustración 6 Alumnos de secundaria en las II Jornadas de Robótica de la ETSIIT

Los talleres impartidos en todas las ediciones de la Cantabria.Net en los que se pudo trabajar con los robots educativos Moway y con los Lego NXT Durante las jornadas se recibieron abiertamente tanto estudiantes de la propia universidad como a alumnos y profesores de educación secundaria, despertando nuevas vocaciones en estudiantes a punto de incorporarse a la universidad.

Otro ejemplo son los talleres impartidos en todas las ediciones de la Cantabria.Net en los que se pudo trabajar con los robots educativos Moway y con los Lego NXT y en los que participaron desde niños de ocho años hasta padres de familia compitiendo en diferentes retos como el sumo robot.

Asimismo, los talleres impartidos en la Universidad en el marco del Aula de Nuevas Tecnologías también con Lego y en los que participaron estudiantes de diferentes carreras universitarias, profesores, padres y estudiantes de primaria, ejemplifican el creciente interés en la robótica educativa.



Ilustración 7 Muestra de la variedad de público en uno de los talleres de robótica del Aula de nuevas tecnologías

### 2.1.2. España

---

La situación a nivel nacional es bastante similar a la local, a pesar de la incorporación de la robótica y de la programación de computadores específicamente en el itinerario curricular de la asignatura de Tecnología en los últimos cursos de la secundaria únicamente existen casos aislados de uso de la robótica en la educación.

Además esta asignatura, Tecnología, corre peligro de desaparecer debido a que la última reforma de la educación la dejaría como opción a elegir por las comunidades y los centros educativos.

En Cataluña se está trabajando mucho y activamente en este aspecto, introduciendo robots como medio de enseñar resolución de problemas reales ya durante el bachillerato y la educación secundaria.

No obstante, su uso práctico y más amplio sigue todavía limitado a la participación por parte de voluntarios, alumnos de colegios e institutos, en competiciones de robótica, algunas de las cuales resumimos a continuación:

#### 2.1.2.1. *Competiciones a nivel local*

---

##### 2.1.2.1.1. Concurso nacional de robots “Ciutat de Mataró”

Esta competición está organizada por la Escuela Universitaria Politécnica de Mataró, tiene competiciones de robots en las modalidades de sumo, mini-sumo y rastreadores.

##### 2.1.2.1.2. Robolid

Organizada por la Asociación de Microbótica de la Universidad de Valladolid, tiene competiciones de robots en las modalidades de sumo, mini-sumo, rastreadores y velocistas, es una competición a nivel regional pero a la que puede apuntarse quien desee y que es clasificatoria para “Hispabot”.

##### 2.1.2.1.3. Alcabot

Destinada a gente interesada en la robótica a nivel nacional, está organizada por el departamento de electrónica de la Universidad de Alcalá, es también una prueba clasificatoria para “Hispabot” y tienen competiciones de sumo, laberintos, “alcafutbol” además de “funygolf”.

#### 2.1.2.2. *Competiciones nacionales*

---

##### 2.1.2.2.1. Hispabot

Al igual que Alcabot está organizada por el departamento de electrónica de la Universidad de Alcalá, es una competición a nivel nacional en muchas modalidades, desde sumo hasta velocistas, pasando por laberintos o futbol.

#### 2.1.2.2.2. Ceabot

Es un concurso de robots humanoides organizado por el Comité Español de Automática para alumnos de grado y postgrado de los departamentos de Automática de las Universidades Españolas con las modalidades de carrera cronometrada y sumo.

#### 2.1.2.2.3. Cosmobot

Es una competición organizada por CosmoCaixa, el Museo de la ciencia de la Obra Social "la Caixa" y RoboticSpot, en las categorías de sumo y velocistas y con unos premios totales de cerca de 10.000 €.

Además existen multitud de seminarios y cursillos breves de robótica y electrónica aplicada. Los más conocidos por el número de ediciones y asistentes son los realizados en el contexto de las dos grandes "Partys" en España, la "Campus Party" de Valencia con su campus bot y la "Euskal Party" en Barakaldo, que en casi todas sus ediciones ha incorporado tanto charlas sobre plataformas como Arduino como talleres prácticos de robótica.

Igualmente durante los últimos tiempos se han venido realizando numerosos cursillos de robótica para niños, especialmente durante los periodos vacacionales. Algunos que se pueden mencionar son el "ClauticCamp" organizado por una empresa privada en y que ya va por su tercera edición, el "Camp Tecnológico" también ya por su tercera edición y organizado por la Asociación de Robótica Educativa de Extremadura y los "Camp Tecnológicos" que han tenido lugar durante el verano de 2013 en ocho ciudades españolas (Bakio, Madrid, Bilbao, Barcelona, Pamplona, Santander y Logroño) y organizados por diversas instituciones educativas del País Vasco como la Universidad de Deusto de Bilbao, la Ikastola Axular y Politécnico EASO en San Sebastián, el Colegio Sagrado Corazón de Vitoria o la Escola Universitaria Salesiana de Sarria de Barcelona.

## 2.2. Plataformas robóticas para docencia

En la actualidad existen multitud de plataformas robóticas que se pueden utilizar en la enseñanza, a continuación se detallan algunas de las más conocidas/utilizadas:

### 2.2.1. Lego Mindstorms

Es un juego de robótica fabricado por la empresa Lego y desarrollado en colaboración con el Massachusetts Institute of Technology (MIT).

La primera versión del Lego Mindstorm aparece en 1998 y consistía en un kit con más de setecientas piezas y un bloque RCX (que contenía el microcontrolador y las interfaces con los sensores y actuadores) tuvo unas ventas de más de 80.000 unidades en los primeros tres meses que estuvo a la venta. Costaba aproximadamente 200\$ (1998)

El cerebro del bloque RCX era un microcontrolador Hitachi H8/3292 funcionando a 16 Mhz., disponía de una memoria ROM de 16kb y de 32kb de memoria RAM, además disponía de un display LCD donde se mostraban datos como las entradas y la carga de la batería y disponía de un puerto de comunicación por infrarrojos lo que le permitía conexión inalámbrica con los dispositivos de la época (portátiles y PDAs). Se alimentaba mediante seis pilas LR6 de 1,5v.



Ilustración 8 Bloque RCX de Lego Mindstorms

La programación se realizaba mediante bloques desde un entorno proporcionado por Lego aunque pronto surgieron alternativas desarrolladas por la comunidad para poder programar los robots desde en C hasta Java.

En 2006 Lego pone a la venta una nueva versión del sistema, Lego Mindstorms NXT con múltiples mejoras, su microprocesador para a ser un ARM7 de 32 bits con 256kb de memoria flash y 64kb de RAM y el bloque añade conectividad USB, lo que mejora su conectividad con ordenadores personales, y bluetooth permitiendo la comunicación con otros robots, móviles, tablets, etc.

Los principales inconvenientes de esta plataforma para el objetivo de este proyecto son el precio, cada plataforma Lego NXT con su licencia ronda los 500-1000€ por equipo y su formato autocontenido, los sensores no se pueden ni modificar ni se puede añadir electrónica de usuario.

Las ventajas de esta plataforma son la familiaridad con el sistema Lego y la simplicidad de su programación, alguien sin conocimientos técnicos puede comenzar a desarrollar para Lego tras apenas una hora de contacto con la plataforma.

### 2.2.2. Moway

Construido al igual que el Aurebot alrededor de un microcontrolador PIC (un 18F86J50 en este caso) dispone de un grupo motor incorporado, varios diodos LED, sensores infrarrojos anticollisión, sensores de luminosidad y temperatura, micrófono y un altavoz. Se alimenta a través de una batería LiPo de una celda que se recarga mediante un puerto USB.



Ilustración 9 Robot Moway

La programación de la plataforma se realiza esencialmente mediante un entorno (MowayWorld) desarrollado por la propia empresa que los fabrica y es de tipo gráfico aunque se puede programar tanto en C como en ensamblador para PICS proporcionándose las librerías necesarias.

Los principales problemas de esta plataforma son el precio (330€ por el set educacional con dos robots y las guías educativas) y la limitación espacial para añadir electrónica de usuario. Las ventajas son su tamaño, documentación y presentación.

### 2.2.3. Skybot

El robot Skybot es un robot también con alma educacional diseñado para los talleres de introducción a la robótica impartidos durante más de 5 años en la Campus Party de Valencia. Tiene una estructura a medida fabricada en metacrilato cortado por laser y su “cerebro” lo compone una tarjeta “SkyPic” basada en un microcontrolador 16f876A de Microchip.

Dispone de cuatro sensores de infrarrojos de corta distancia, dos sensores de contacto y un sensor de luminosidad. La motorización es similar a la del Aurebot, estando compuesta de dos servos modificados para la rotación continua.

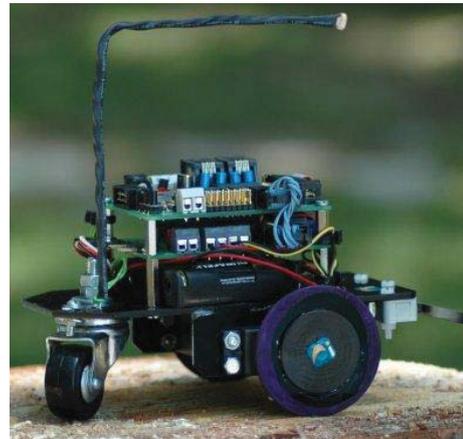


Ilustración 10 Robot Skybot (2)

### 2.2.4. Boebot

La plataforma robótica Boebot está construida alrededor de un microcontrolador Basic Stamp de Parallax, dispone de todo lo necesario para empezar a trabajar.

Los programas para la plataforma se desarrollan en PBasic, una vertiente de Basic para el robot.

Es muy similar al Aurebot, disponiendo de una motorización basada en servos, sensores de contacto, luminosidad e infrarrojos además de una pequeña placa de prototipos donde añadir electrónica de usuario.

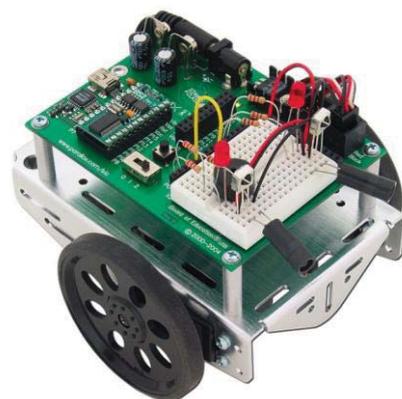


Ilustración 11 Boebot

Se alimenta mediante cuatro pilas LR6 de 1,5v. y tiene un coste por unidad de 160\$ (unos 121€)

Es la plataforma a la que sustituye el Aurebot en la impartición de clases prácticas de electrónica en la titulación de Ingeniería de Telecomunicación.

### 2.2.5. Arduino y Raspberry Pi

Estas dos plataformas no son plataformas robóticas por sí mismas, no obstante, se incluyen por la importancia de la comunidad que llevan detrás y por la simplicidad y facilidad en la construcción y desarrollo de dispositivos electrónicos con ellas.

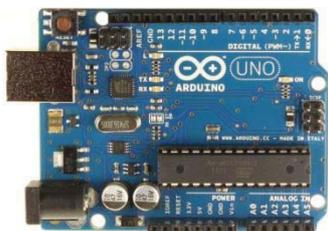


Ilustración 12 Placa Arduino Uno

En el caso de Arduino la propia placa contiene todo lo necesario para construir un robot únicamente añadiendo un par de motores (con su correspondiente driver de potencia) o servos de rotación continua (sin driver al poder usar los pines PWM para controlarlos) y todos los sensores que necesitemos ya sean digitales o analógicos.

En su versión más simple, la Arduino Uno, la placa dispone de un microprocesador ATmega328 funcionando a 16Mhz. con 31kB de memoria flash disponible para programas del usuario.

Dispone de seis pines analógicos (con lectura ADC) y catorce pines digitales, seis de los cuales son capaces de proporcionar una salida PWM para el control de servos o la generación de sonidos.

Además permite alimentar la placa y parte de los dispositivos a partir de su propia etapa de alimentación que permite una entrada entre 7 y 20 v. para producir una tensión de trabajo de 5v.

El entorno de desarrollo está basado en Processing, con un lenguaje muy similar a C y únicamente es necesario un cable USB para programarlo. El coste es de 22€ por placa y dispone de una gran comunidad de desarrolladores.

La Raspberry Pi tiene un planteamiento muy diferente siendo un “ordenador completo” de muy bajo coste (25-35€ dependiendo de la versión) que contiene todo lo necesario tanto para usarlo como ordenador de sobremesa, tarjeta gráfica con salida HDMI y RCA, USB para ratón y teclado, conexión Ethernet, como para el desarrollo de aplicaciones robótica con sus ocho pines de uso general y buses serie TTL, SPI, I2C.



Ilustración 13 Raspberry Pi Modelo B

Dispone de un procesador Broadcom BCM2835 que funciona a 700 Mhz. (“overclockeable” hasta 1Ghz.) y funciona con una gran cantidad de sistemas

operativos como Debian o Android. Permitiendo el desarrollo de aplicaciones en multitud de lenguajes, desde el bajo nivel con C hasta el alto con Python, pasando incluso con la programación gráfica con Scratch.

No es necesario hardware ni software adicional para su programación, un teclado y ratón USB y una televisión con entrada RCA bastan para desarrollar en esta plataforma.

### 3. El AureBot

---

Con el fin de obtener una herramienta educativa, y aprovechando la plataforma desarrollada durante las Jornadas de Robótica y Electrónica de la ETSIIT, se plantea desarrollar una plataforma robótica didáctica. Dicha plataforma se compone de un chasis mecánico con motorización, una electrónica de control con sus correspondientes sensores y un entorno de programación y test (AureBot GUI) que permite interconectar un ordenador con el AureBot.

#### 3.1. Funcionalidad

---

Aunque el objetivo principal de las plataformas 1.0 y 2.0 es el mismo: el aprendizaje práctico, las plataformas Aurebot 1.0 estaban diseñadas para un grupo reducido de aplicaciones, las suficientes para un seminario de unas cuatro o cinco horas.

Por una parte el robot “sigue líneas”, para lo que se incorporó un par de sensores reflexivos CNY70 (en la Ilustración 14 en recuadro azul) que eran capaces de proporcionar al robot la cantidad de luz reflejada y por lo tanto de distinguir entre un fondo claro y uno oscuro.

Y el robot “evita obstáculos” para lo que el robot disponía de dos interruptores de final de carrera (en la Ilustración 14 en recuadros verdes) que informaban al robot de un choque frontal y en qué lado se había producido.

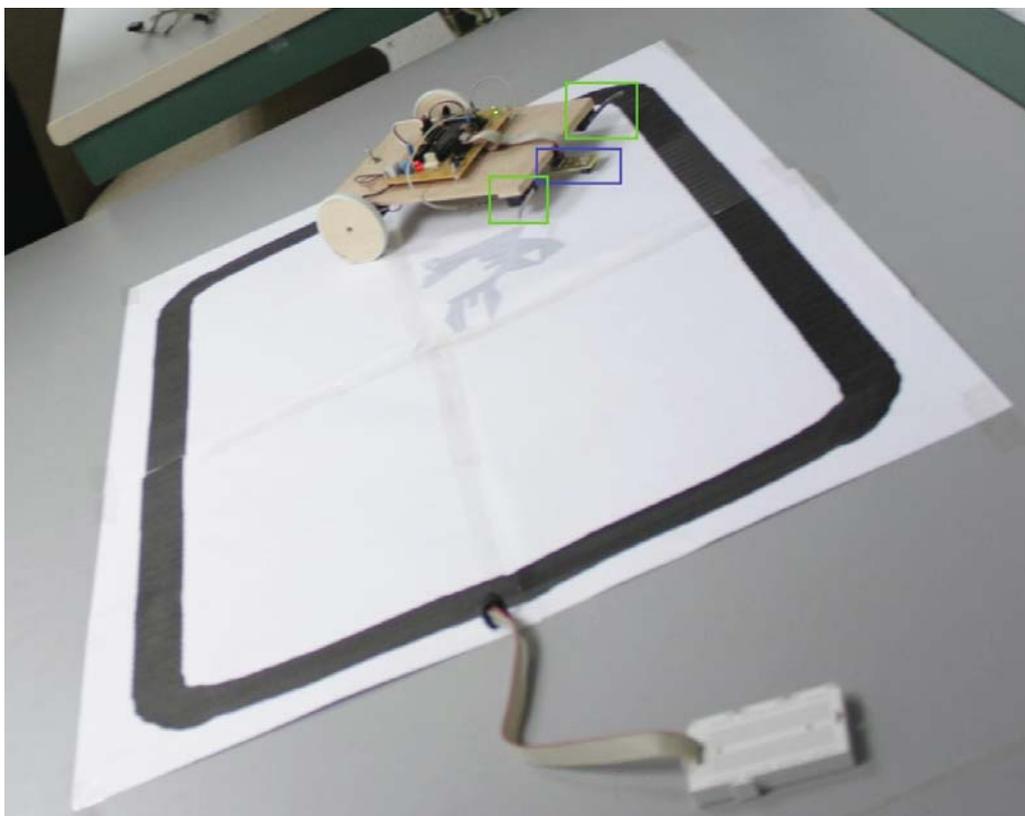


Ilustración 14 Aurebot 1.0 en un circuito sigue líneas

El objetivo del presente proyecto necesitaba de un robot más flexible que ofreciera mayor abanico de posibilidades y para ello se han realizado toda una serie de mejoras y modificaciones que se desarrollan a continuación.

## 3.2. Hardware

---

### 3.2.1. Aureboard, descripción general

---

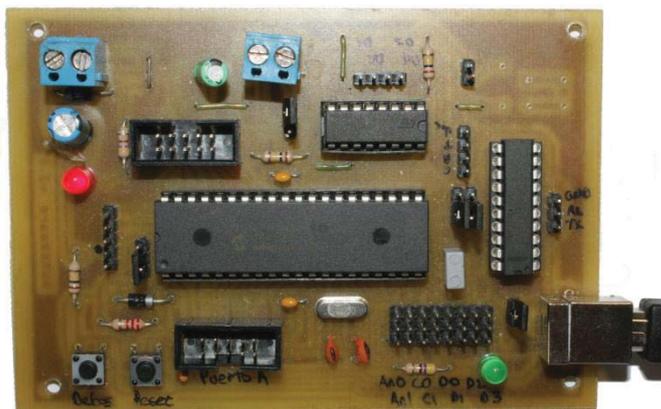


Ilustración 15 Aureboard 2.1

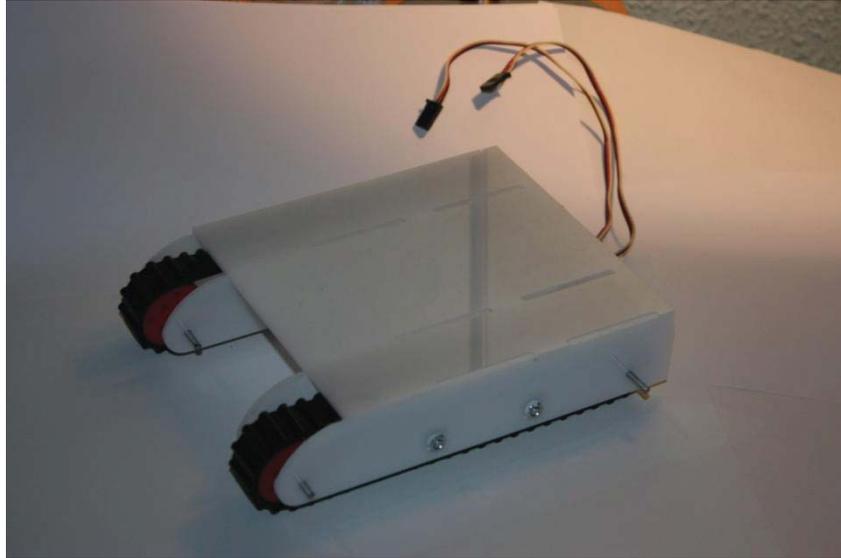
Con un tamaño de 115 mm. X 80 mm. La Aureboard v.2 dispone de un microcontrolador Microchip PIC 18F4550, un puerto analógico con 8 entradas que pueden usarse también como entradas/salidas digitales, un puerto de entrada/salida digital con 8 pines, un puerto de 8 pines accesible directamente en placa con 2 entradas analógicas y 6 entradas/salidas digitales, un driver para motores de corriente continua L293D, un relé, un conector de 4 pines al puerto serie, un adaptador ttl/rs232 MAX233, el puerto USB y el puerto ICSP. Además de 2 pulsadores, *debug* y *reset*, y 2 diodos LED: uno rojo indicador de la presencia de alimentación y uno verde controlable por software.

Respecto a la primera versión de la Aureboard se han introducido una gran cantidad de mejoras: una mejora en el bloque de alimentación y la posibilidad de seleccionar si se alimentará directamente o mediante el puerto USB, además de separar (configurable mediante jumper) la alimentación de la lógica de la alimentación de la etapa de potencia; una doble salida del puerto serie en lógica TTL y RS232, se ha añadido un puerto USB miniB en la placa y se ha diseñado un nuevo *layout* de la placa más limpio.

### 3.2.2. Chasis

---

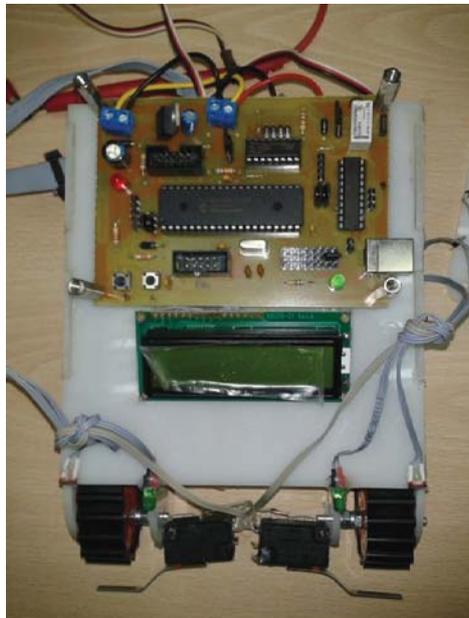
El chasis del Aurebot está diseñado en Autocad y cortado en una fresadora CNC. Se diseñó en base a la motorización con orugas y consiste en siete piezas de metacrilato blanco.



**Ilustración 16 El chasis de un Aurebot y su motorización**

El cuerpo del chasis hace las veces de soporte para la placa Aureboard, los sensores y los dispositivos básicos. Además, una bandeja superior contiene la placa de prototipos donde se incorpora la electrónica de usuario y protege de golpes y cortocircuitos a la placa principal.

Una segunda bandeja inferior hace las veces de contenedor de las baterías LiPo y cargador o de la caja de las pilas en el caso de utilizarse.



**Ilustración 17 Vista superior del chasis del Aurebot (sin bandeja de electrónica)**

### 3.2.3. Unidad de proceso

---

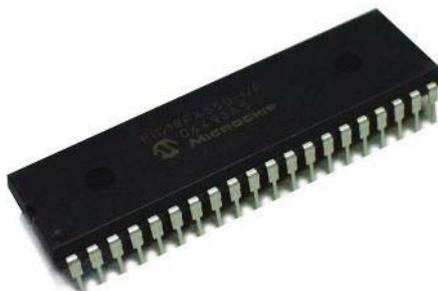


Ilustración 18 Microchip PIC 18F4550

El diseño de los Aurebots gira en torno a la unidad central de proceso: un PIC 18F4550 de Microchip [2] funcionando a 48 Mhz.

La elección de este microprocesador específico se debe a varios motivos: pertenece a la familia de los PICs, de los que existe una notable comunidad de personas desarrollando (especialmente con los 16F84 y sus sucesores 16F628 y 16F88) y que hasta hace muy poco se seguían estudiando en la mayor parte de las universidades; son bastante asequibles, con un precio de aproximadamente 5€ y los estudiantes pueden conseguirlos gratuitamente gracias al programa de “*samples*” de Microchip.

Además está disponible en formato PDIP lo que facilita su uso en placas de prototipos y tiene unos requerimientos mínimos de operación necesitando únicamente de un par de condensadores cerámicos, un cristal resonador y una resistencia funcionar.

La capacidad de ser dispositivo U.S.B. programable por software es el argumento final para su elección, esta capacidad nos permite programarlo como dispositivo serie para poder trabajar con la plataforma desde cualquier PC.

Los PICS son una familia de microcontroladores, inicialmente desarrollados por General Instruments y fabricados actualmente por Microchip.

Disponen de un conjunto de instrucciones de tipo RISC (Reduced Instruction Set Computer) de entre 35 instrucciones para los pics de gama baja, la familia 16F, y 70 en la gama alta, la familia 18F.

Tienen una CPU de 8 bits y su velocidad de reloj puede variar entre los 32 Khz. usando los osciladores internos de los que disponen algunos modelos, a los 48 Mhz. utilizando cristales externos y módulos PLL, aunque también existen PICs con CPUs de 16 bits, los dsPICS diseñados para el tratamiento de señal y de 32 bits, los PIC32.

Una nota importante es que ejecutan una instrucción cada 4 ciclos de reloj, esto quiere decir que si nuestro procesador tiene una frecuencia de reloj de 48 Mhz. se ejecutarán 12.000.000 de instrucciones por segundo.

Algunos microcontroladores muy conocidos son el 16F84, todavía utilizado para enseñar a programar en ensamblador y que podemos encontrar en algunos sistemas actuales (sistemas de alarma) su posterior actualización el 16F84A y sus revisiones el 16F628A y el 16F88, todos de 16 pines, los 16F876, 16F877 y sus actualizaciones

16F886 y 16F887, y los relativamente nuevos 18F2550 y 18F4550 con capacidades USB, estos últimos de 40 pines.

El 18F4550 pertenece a la gama alta de los microcontroladores PIC. Posee una arquitectura RISC (Reduced Instruction Set Computer) con unas instrucciones de 16 bits de longitud y 8 bits de datos.

Puede operar entre 31 Khz. Y 48 Mhz. a una tensión entre 2v. y 5v. dependiendo de la frecuencia de operación.

Tiene 32 kb. de memoria flash disponibles para la programación del usuario que permiten un mínimo de 100.000 ciclos de borrado/escritura, además dispone de 256 bytes de EEPROM para el almacenamiento de datos a largo plazo o entre programaciones y 2kb. de memoria RAM.

Entre las características destacables del microprocesador se encuentran:

- Capacidad para ser dispositivo USB (1.1 y 2.0) pudiendo actuar como puerto serie, ratón, teclado, etc. O un dispositivo personalizado.
- Tres fuentes de interrupción externa
- Dos módulos PWM
- Puerto serie (USART)
- Master Synchronous Serial Port (MSSP) para la interconexión con protocolos I2C o SPI en sus modos maestro y esclavo.
- Trece puertos analógicos con convertidores analógico/digitales de precisión variable de 8 a 10 bits.
- Tres temporizadores.

#### 3.2.4. Jumpers de configuración

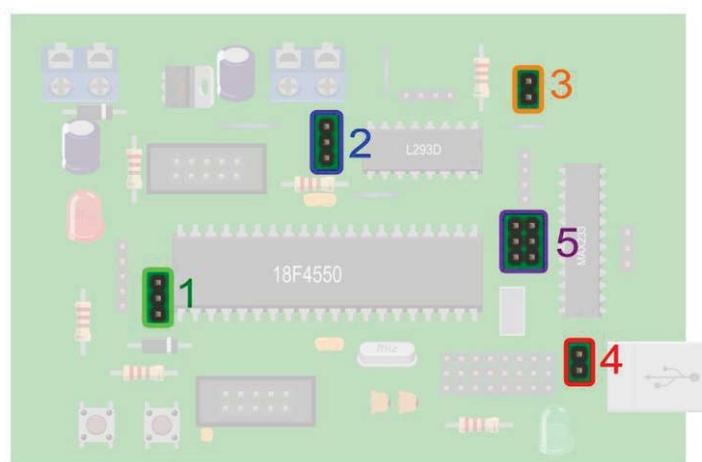


Ilustración 19 Jumpers de configuración

La placa dispone de seis jumpers que configuran diversas posibilidades de la placa:

**1.Origen de reset:** Define el origen del reset de la placa, puede elegirse entre el pulsador de reset (posición más externa) o el reset del conector ICSP (posición más interna).

**2.Alimentación del driver de continua:** Configura el origen de la alimentación del driver de continua, si se selecciona la posición más interna la alimentación del driver provendrá del conector de alimentación general y será compartida con el resto de la placa, si se selecciona la posición más externa la alimentación del driver provendrá del conector de alimentación auxiliar y será independiente de la alimentación de la placa.

**3.Alimentación via USB:** Si este jumper se encuentra cerrado se alimentará el circuito desde el conector USB.

**No han de usarse simultáneamente los conectores de alimentación USB y principal para evitar dañar los puertos USB del ordenador.**

**4.Relé:** Si se cierra este jumper se redirecciona una de las salidas del driver de motores para usar el relé integrado en la placa.

**5.Puerto serie:** Este jumper redirecciona los pines TX/RX del puerto serie del microcontrolador hacia la salida TTL de la placa, si se coloca en la posición más externa, o hacia la salida RS232, si se posiciona en la posición mas interna.

### 3.2.5. Interfaces con el exterior

La Aureboard dispone de varios métodos de comunicación con el exterior de diversa naturaleza, desde los más simples puertos digitales configurables como entrada o salida hasta el mucho más complicado puerto USB, pasando por conversores analógico-digitales y salidas de potencia.

#### 3.2.5.1. Diodos LED

Se dispone de dos diodos LED soldados en la placa: uno rojo y que indica la presencia de alimentación (no es controlable por *software*) y uno verde que al arrancar indica si se ha encendido en modo *bootloader* o ha tenido un arranque normal y que en arranque normal pasa a ser controlable por software.

Adicionalmente a los Aurebots se les ha montado un par de diodos LED frontales verdes que se pueden controlar por software.

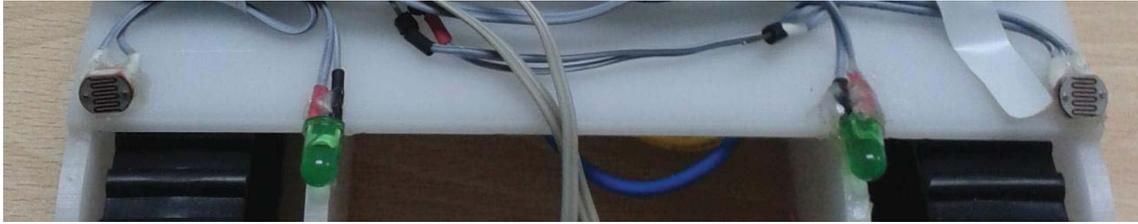


Ilustración 20 Detalle de los leds frontales del Aurebot

### 3.2.5.2. Puertos de conexión

Existen tres puertos de uso general en la placa:

El puerto A contiene todas las entradas analógicas de la placa, se han agrupado en un solo conector por ser los puertos más versátiles puesto que se pueden configurar también como entradas y salidas digitales, es un conector molex de 10 pines y por regla general estará conectado a la placa de prototipos situado en la parte superior del robot.



Ilustración 21 Pinout del puerto A

El puerto B es un puerto estrictamente digital, es un conector molex de 10 pines que se usa para el control del panel LCD (aunque existe don líneas libres, RB0 y RB1, que se pueden usar tanto como entradas como salidas digitales además de disponer de interrupciones por cambio y por flanco de subida y bajada)



Ilustración 22 Pinout del puerto B

Un puerto adicional con pines de uso general y sin conector específico, preparado para conectores estándar de 3 pines (señal, tierra, Vcc) esta disponible directamente en la placa, contiene seis pines adicionales que pueden configurarse como entradas/salidas digitales y los dos primeros canales analógicos que se usan principalmente para conectar las LDRs frontales del robot o los sensores ópticos reflexivos CNY70.

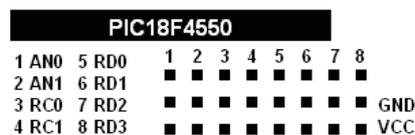


Ilustración 23 Pinout del puerto general

### 3.2.5.3. Puerto serie

La interfaz básica de un microcontrolador con el mundo exterior es el puerto serie, en el caso de la placa Aureboard dispone de un puerto serie TTL/5v. y de un adaptador de niveles TTL/RS232 Max233 [3].

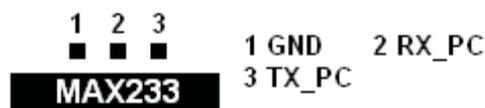


Ilustración 24 Pinout del puerto serie

Este puerto ha quedado en desuso debido a la incorporación de un puerto serie virtual a través de la capacidad USB del microcontrolador, pero sigue estando disponible desde la programación para su uso con otros dispositivos serie o como herramienta de depuración.

### 3.2.5.4. U.S.B.

Una de las principales diferencias entre la Aureboard 1.0 y la versión 2.0 es la conexión con el PC vía USB.

Puesto que los robots se usan en ordenadores actuales y el avance de la tecnología se está llevando por delante la incorporación de puertos serie en los ordenadores de sobremesa y portátiles, se incorporó un puerto serie USB de manera que mediante un pequeño driver de apenas unos kB. se permite la operación del robot en cualquier ordenador con sistema operativo Windows.

También se soporta la operación en sistemas operativos unix (Linux, Mac, etc.) siendo su configuración transparente y no necesitando de driver/módulo alguno.

Además el puerto USB provee energía que aunque limitada (5v. entre 100mA y 500mA) permite hacer pruebas básicas con el robot sin necesidad de una fuente de alimentación externa.

### 3.2.5.5. L.C.D.

Como método más natural de comunicación con el exterior y para permitir ver parámetros que se estén controlando sin la necesidad de un ordenador, el robot dispone de un display de cristal líquido iluminado de 2 líneas de 16 caracteres.

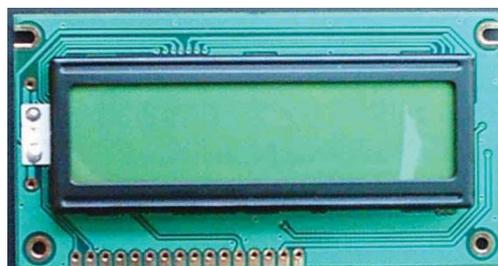


Ilustración 25 Panel LCD 16x2

Desde el código se puede escribir/borrar lo que se necesite en este, proporcionando una interfaz intuitiva para el usuario que permite la comunicación con el robot sin necesidad de un computador.

### 3.2.5.6. Placa de prototipos

---

Como medio para incorporar electrónica de usuario se ha añadido a los robots una placa de prototipos de pequeño tamaño conectada al puerto A de la placa, disponiendo en esta de alimentación (+5v y tierra) y de conexión con los pines AN0 a AN7 del PIC, que pueden configurarse tanto como salidas/entradas digitales como de entradas analógicas.

Este accesorio permite la rápida incorporación a la plataforma de nuevos sensores, componentes y electrónica a medida en general por el alumno, sin necesidad de tocar las partes internas del robot.



Ilustración 26 Placa de prototipos incluida en los Aurebots

### 3.2.6. Motorización

---

El Aurebot dispone de dos motores con reductora (servos adaptados para el giro continuo) que mueven el robot mediante unas orugas permitiendo a este su desplazamiento e interacción con el entorno.



Ilustración 27 Detalle de la motorización del Aurebot

Debido a los requerimientos de potencia los motores están conectados a través de un driver de potencia L293D [4] al microcontrolador, lo que permite controlar el movimiento y el sentido de giro de cada uno de los motores además de separar la alimentación lógica del sistema de la alimentación de potencia, evitando interferencias y posibles causas de inestabilidad.

Además se han programado unas librerías que permiten controlar también la velocidad de giro de los motores y por lo tanto la velocidad de avance del robot mediante PWM por software.

PWM, o modulación por ancho de pulsos, es una técnica por la cual se modifica el ciclo de trabajo de una señal periódica, controlando de esta manera la cantidad de energía que se envía a una carga, en este caso, a los motores y por lo tanto variando su velocidad.

### 3.2.7. Alimentación

La Aureboard se puede alimentar mediante 3 vías diferentes:

- La primera es a través del conector de alimentación, mediante este se pueden conectar tensiones de hasta 30v, los cuales mediante un regulador de tensión positiva LM7805 [5] se adaptarán hasta los 5v con los que trabajan todos los dispositivos de la placa, además se incluyen etapas de filtrado y un diodo de protección contra inversiones de polaridad.

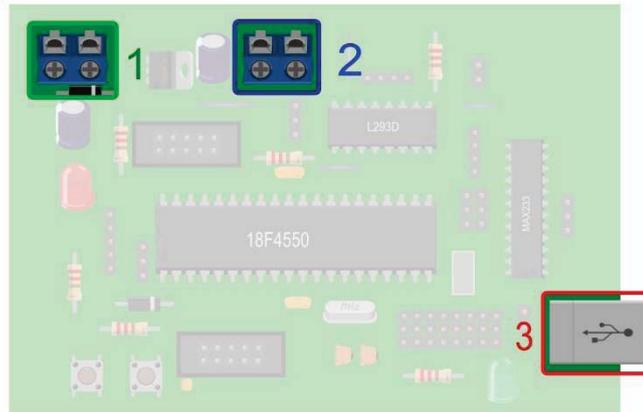


Ilustración 28 Conectores de alimentación de la Aureboard

- La segunda es el conector de alimentación del driver de potencia, este se puede alimentar desde la alimentación de la placa directamente o a través de este conector alimentarlo a una tensión superior, evitando caídas de tensión que afectarían a la estabilidad de la placa y un sobrecalentamiento del estabilizador de tensión si el consumo total supera 1A.

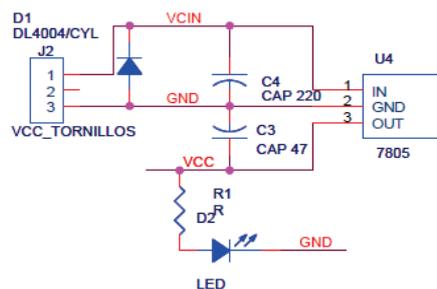


Ilustración 29 Etapa de alimentación de la Aureboard

- La tercera forma de alimentar la Aureboard es mediante el puerto USB, hay que usar esta forma de alimentación con las precauciones debidas: los puertos USB proporcionan entre 100mA y 250mA (en casos especiales hasta 500mA o incluso 1A) corriente suficiente para alimentar la placa pero no otros dispositivos adicionales como podrían ser relés o la motorización de la placa.

### 3.2.7.1. Alimentación por baterías

Otra de las mejoras introducidas en la segunda versión de la plataforma es la alimentación mediante baterías.

Se ha añadido a los Aurebots unas baterías LiPo (polímero de litio) de dos celdas (7,4 v.) y dos doscientos miliamperios/hora de capacidad además de un cargador para esta

incorporado al robot, (se incluye el esquema del cargador en el anexo “Esquema del cargador de baterías de dos celdas LiPo”).

Estas baterías proporcionan una mayor autonomía que las pilas y simplifican su carga, basta con conectar el robot a una fuente de laboratorio durante cerca de una hora para tenerlo de nuevo a plena carga.

Adicionalmente a las baterías de litio es posible, retirando la batería LiPo y el cargador, incorporar una caja de cuatro pilas LR6 en la parte interior del robot para aquellos casos en los que una fuente de laboratorio no esté disponible para la carga de la batería o simplemente se desee operar con pilas, ya sean normales o recargables.

La plataforma no requiere de ningún mantenimiento, no obstante, las baterías de LiPo sí que requieren ciertos cuidados mínimos: no se pueden descargar completamente (menos de 3,7v. por celda) puesto que se dañaría la batería y se requiere cargarlas a entre un 70% y un 100% de capacidad antes de su almacenaje (cuando no se vayan a utilizar durante un periodo largo de tiempo).

### 3.3. Software

---

El software del Aurebot está programado en C para el compilador *PIC MCU Compiler* de CCS Inc. Aunque alguna librerías incluyan partes de código en ensamblador para PICs por cuestiones de funcionalidad de las librerías básicas de CCS o por velocidad.

Dada que la capacidad de memoria del PIC es limitada se ha optado por la modularización de las librerías, desarrollando una librería central con las funciones esenciales para la operación de la plataforma y separando en librerías adicionales aquellas porciones de código necesarias para el control de los diversos dispositivos.

Además se ha desarrollado un cargador de programas para evitar la necesidad de una programadora hardware y varios programas de ejemplo y comprobación de la plataforma.

#### 3.3.1. Bootloader

---

Típicamente los microcontroladores PIC se programan mediante una interfaz denominada I.C.S.P. (In Circuit Serial Programming). El problema es que programarlos mediante esta interfaz requiere de una programadora, es decir, hardware adicional en la placa o fuera de esta y costes adicionales en componentes o programadoras comerciales.

Para evitar esto se opto por aprovechar la capacidad de estos microcontroladores de “autoprogramarse” y se diseñó un “bootloader”.

El “bootloader” es un fragmento de código que se incluye en todos los programas que se creen para la plataforma y que se ejecuta nada más arrancar la placa. Este comprueba el botón de la placa y si está pulsado enciende el diodo LED de la placa y permanece esperando a que se envíe por el puerto USB un nuevo programa para escribirlo en su memoria (“modo programación”).

Este fragmento de programa debe programarse por primera vez siempre mediante una programadora hardware (T-20, GTP USB, o equivalente).

Un programa para PIC se convierte (compila) desde el código C hasta código binario para el microcontrolador produciendo un fichero “.hex” que describe en varias líneas en hexadecimal la dirección y el contenido de la memoria de programa del 18F4550.

Un fragmento de un archivo “.hex”:

```
:1000100002D004EF0CF0000C25A003D01CEF00F080  
:1000200002D00CEf0CF0000C056ED8CF06F0E0CF3C  
:1000300007F00001E9CF0DF0EACF08F0E1CF09F0B9
```

La estructura de cada una de estas líneas es:

- **Start code** Un carácter, ":" para líneas con contenido, ";" para comentarios.
- **Byte count** Dos caracteres HEX que indican el número de datos en Data.
- **Address** Cuatro caracteres HEX para la dirección del primer dato en Data.
- **Record type** Dos caracteres HEX que indican el tipo de línea, de 00 a 05. (ver más abajo)
- **Data** Secuencia de  $2 * n$  caracteres HEX correspondientes a los Byte count datos definidos antes.
- **Checksum** Dos caracteres HEX de Checksum calculado según el contenido anterior de la línea en la forma: El byte menos significativo del complemento a dos de la suma de los valores anteriores expresados como enteros los caracteres hexadecimales (sin incluir ni el Start code ni al él mismo)

Una vez puesto el robot en “modo programación” el programa “Aurebot GUI” envía por el puerto serie cada una de estas líneas, el robot comprobará mediante la suma de comprobación que cada línea llega correctamente y la escribirá en la memoria del microprocesador, en caso de que haya habido algún error realizará una petición de repetición y el programa reenviará la línea de nuevo hasta que se escriba correctamente.

### 3.3.2. Interfaz serie

---

Desde la programación se permite el envío y recepción de cadenas de caracteres a través del puerto serie USB de la placa, el ordenador, ya sea desde la interfaz gráfica “Aurebot GUI” o desde cualquier programa de terminales serie puede a su vez también recibirlos y enviarlos lo que proporciona una manera de depurar programas o mostrar resultados.

```
printf("AUREBOT\r ;Hola mundo!"); // Envía texto por el puerto
serie TTL/RS232

printf(usb_cdc_putc, "AUREBOT\r ;Hola mundo!"); // Envía texto
por el puerto serie USB
```

### 3.3.3. Interfaz “gráfica” (L.C.D.)

---

Además de la interfaz serie el Aurebot dispone de un panel LCD en el cual se puede escribir, de una manera similar a como se escribe en el puerto serie, datos o frases.

Los displays LCD tradicionales requieren para su control unas siete líneas digitales, cinco de datos y dos de control, que en este caso tomamos del puerto digital o puerto B.

Mediante esta librería abstraemos al usuario de toda la lógica requerida para el envío de datos a la pantalla, su inicialización y configuración, la transformación de los datos para su envío y presentación teniéndose que preocupar este únicamente del formato adecuado puesto que se dispone únicamente de dos líneas de dieciséis caracteres.

Un ejemplo de su uso desde software podría ser:

```
lcd_init(); // Inicializa el panel LCD
lcd_setcursor_vb(0,0); // Posiciona el cursor en el origen
printf(lcd_putc, "AUREBOT\r ;Working!"); // Envía texto al panel
```

### 3.3.4. Librerías desarrolladas

---

Con el objeto de simplificar el desarrollo para la plataforma se han creado una serie de librerías que permiten el control de diversos dispositivos de manera más abstracta.

#### 3.3.4.1. Librería principal

---

En la librería principal, "aurebot.h", se encuentran las funciones principales para la configuración de la Aureboard, de sus dispositivos (como el puerto USB) y las definiciones de varios componentes como el led controlable ("LED") por software o el pulsador ("PULSADOR"), funciones de simplificación como la función "leer\_adc(puerto analógico)" que se ocupa de configurar, leer y devolver el valor de tensión presente en un puerto analógico de la placa o alias de funciones como "ENCENDER", "APAGAR" o "CONMUTAR".

#### 3.3.4.2. Librería "2 leds"

---

Esta librería únicamente contiene los identificadores de los dos diodos LED disponibles en el aurebot "LED\_I" para el led izquierdo y "LED\_D" para el led derecho.

#### 3.3.4.3. Librería "barra leds"

---

Esta librería proporciona algunas funciones para realizar efectos con una barra de diodos LED además de una función de autocomprobación.

La barra de leds es un dispositivo muy simple que puede conectarse tanto al puerto A como al puerto B y que dispone de ocho diodos LED correctamente polarizados a 5v. y que podemos controlar por software.

#### 3.3.4.4. Librería "bumpers"

Esta librería contiene las definiciones para el uso de los bumpers, o interruptores de final de carrera del robot, "BUMPER\_I" para el detector izquierdo y "BUMPER\_D" para el detector derecho.

#### 3.3.4.5. Librería "CNY70"

Esta librería contiene las definiciones necesarias para hacer uso de los sensores optoreflexivos CNY70, así como la función "cny70\_configurar()" de configuración de estos.

Los CNY70 son unos sensores de infrarrojos de corto alcance basados en un emisor de luz y un fototransistor que hace las veces de receptor, ambos apuntando en la misma dirección.

Ilustración 30 CNY70

Este componente, correctamente polarizado, nos permite detectar la cantidad de luz reflejada en pequeñas distancias.

Podemos usar esta información para distinguir entre fondos claros, aquellos que reflejan mucha luz, y fondos oscuros, aquellos que casi no reflejan luz. Y de esta manera conseguir que el robot realice tareas complejas como seguir un camino de cinta aislante negra en el suelo.

#### 3.3.4.6. Librería "flex\_lcd"

Esta librería proporciona las funciones necesarias para configurar ("lcd\_configurar()") y utilizar el panel LCD de 16 caracteres por líneas ("lcd\_gotoxy(x,y)", "lcd\_putc(c)").



#### 3.3.4.7. Librería "motores"

Esta librería proporciona las definiciones y funciones (“motores\_patras()”, “motores\_palante()”, “motores\_paderecha()”, “motores\_paizda()”, “motores\_parar()”) necesarias para la utilización de los motores de la plataforma.

Ademas también provee de las las funciones y configuración de interrupciones necesarias para el uso del control de velocidad de estos (“motores\_cv\_configurar()”, “motores\_cv\_off()” y las equivalentes a las anteriores con control de velocidad “motores\_cv\_palante(velocidad)”, etc.).

El control de velocidad de los motores se realiza mediante PWM por software utilizando uno de los temporizadores (el timer 2) del microprocesador 18F4550.

#### *3.3.4.8. Librería “servos”*

---

Proporciona las definiciones, funciones y configuración de interrupciones necesarias para el control de servos desde la plataforma (“servos\_configurar()”, “servos\_off()”, “servos\_mover\_a(grados, servo)”).

Al igual que el control de velocidad de los motores el control de servos se realiza mediante PWM por software utilizando el temporizador dos del PIC, por lo que solo puede utilizarse simultáneamente uno de los dos sistemas (control de velocidad o servos).

#### *3.3.4.9. Librerías adicionales*

---

Además de estas librerías de uso para docencia se han desarrollado algunas otras librerías de uso más específico con dispositivos o protocolos que bien podrían ser útiles:

##### *3.3.4.9.1. Librería protocolo I2C*

Librería con funciones adicionales necesarias para la operación con el protocolo serie I2C.

I2C es un bus de comunicación serie diseñado por Philips. Se utiliza para comunicar microcontroladores y dispositivos en muy cortas distancias (mismo circuito impreso) mediante el uso de únicamente tres líneas: datos, reloj y tierra.

##### *3.3.4.9.2. Librería protocolo 1-Wire*

Librería para comunicarse con dispositivos que usen el protocolo 1-Wire. 1-Wire es un protocolo de comunicaciones serie diseñado por Dallas Semiconductors y que se caracteriza por utilizar únicamente una línea para la transmisión y recepción de datos (además de la tierra común). Además esta única línea es también utilizada por que los dispositivos esclavos para alimentarse.

### 3.3.4.9.3. Librería dispositivo DS 18B20

Librería para leer la temperatura de termómetros 1-Wire Dallas 18B20.

El DS1820 es un termómetro con interfaz 1-wire que puede medir temperaturas entre  $-55^{\circ}\text{C}$  y  $+125^{\circ}\text{C}$  en incrementos de  $0,5^{\circ}\text{C}$ .

No necesita de ningún componente externo, ni siquiera alimentación, puesto que utiliza “alimentación parásita” cogiendo la energía que necesita del bus cuando este se encuentra a nivel alto (+5V.).

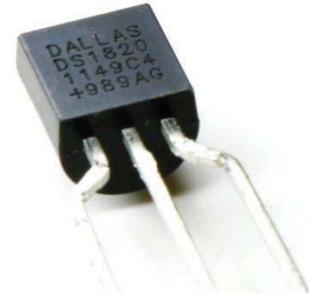


Ilustración 31 Termómetro DS1820

### 3.3.4.9.4. Librería dispositivo DS 1307

Librería para leer y escribir la fecha y hora de un reloj de tiempo real (Real Time Clock, RTC) Dallas 1307 a través del protocolo I2C.

El DS1307 es un reloj de tiempo real, mediante un cristal resonador y una pequeña pila nos permite mantener la hora y fecha en un sistema como pudiera ser la plataforma del robot y que no dispone de un reloj real.

Además dispone de una salida por alarma configurable y que nos permitiría despertar el Pic en horas o fechas preprogramadas.

### 3.3.4.9.5. Librería dispositivo DS 1624

Librería para configurar y leer la temperatura de un termómetro I2C Dallas 1624.

El DS1624 es un termómetro capaz de medir temperaturas de entre  $-55^{\circ}\text{C}$  y  $+125^{\circ}\text{C}$  con una precisión de  $0,03125^{\circ}\text{C}$  (13 bits) y que no requiere de ningún componente externo.

### 3.3.4.9.6. Librería WiiMotion Plus

Librería para configurar y leer los valores del acelerómetro del dispositivo Motion Plus de la Wii mediante un bus I2C.

El Wii Motion Plus es un accesorio de la consola Wii de Nintendo diseñado para mejorar la sensibilidad y precisión al movimiento de sus mandos. Dispone de 2 giroscopios, uno de dos ejes y otro de un único eje que nos proporcionan una lectura sobre la velocidad de giro en los tres ejes.

En los últimos tiempos se ha utilizado ampliamente en robótica por su bajo coste (20€) y su simplicidad de uso gracias al protocolo I2C.

### 3.3.5. Interfaz gráfica (Aurebot GUI)

Como método de control y programación desde el ordenador se ha desarrollado una interfaz gráfica en Visual Basic 6.0, más amigable que una consola de texto.

Dispone de una pantalla principal en la que se muestran mensajes de estado del programa y los mensajes enviados y recibidos a través del puerto serie. Además de los dos parámetros necesarios para conectar con el robot (puerto serie y velocidad) y botones para conectar con la plataforma, enviarla un nuevo programa para su escritura en la memoria del microcontrolador y para la limpieza de la pantalla de mensajes.

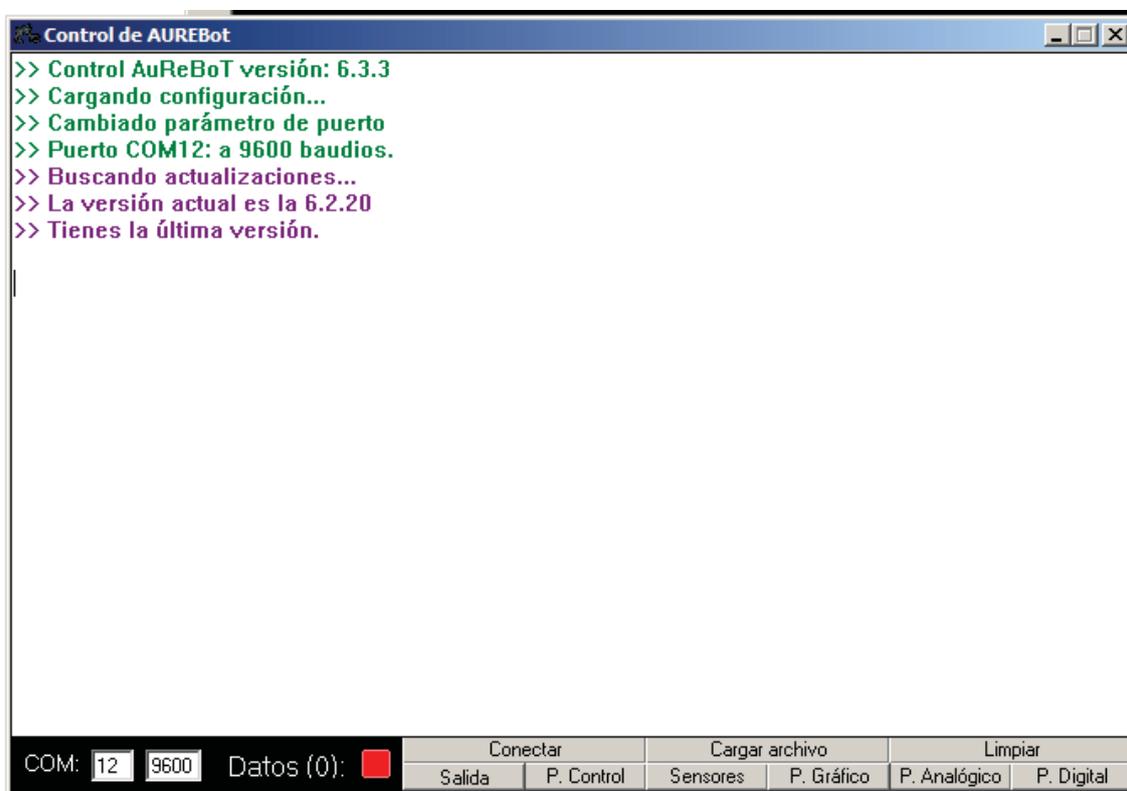


Ilustración 32 Pantalla principal del programa de control del Aurebot

Este programa dispone además de varios paneles adicionales:

El panel de control de motores y dispositivos, permite mover el robot y activar algunos dispositivos que se le pueden incorporar al robot como servos o una fila de diodos LED, así como ejecutar rutinas preprogramadas o reiniciar el robot.



Ilustración 33 Panel de control de movimiento del robot

El panel de sensores, que muestra el estado de varios de los sensores que se le pueden incorporar a la plataforma de manera gráfica como son las resistencias LDR, los sensores de contacto o bumpers y los sensores reflexivos CNY70, además permite el control de encendido y apagado de los leds frontales de la plataforma.

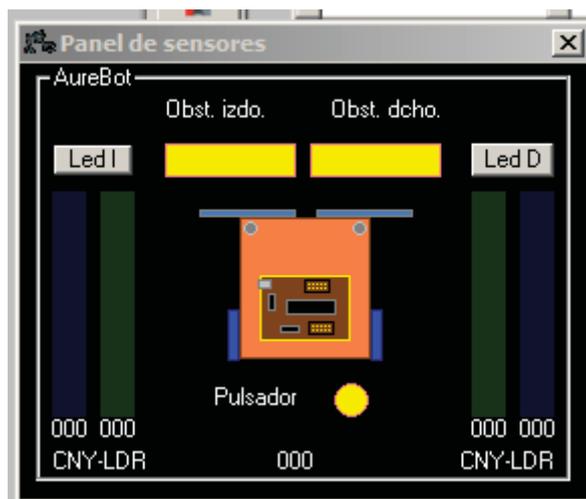


Ilustración 34 Panel de sensores

El panel gráfico, que dibuja en una gráfica tensión vs tiempo el valor presente en los canales analógicos del robot, pudiendo seleccionar los canales así como el tiempo de dibujo además de poder guardar en formato BMP las gráficas. Además permite de manera experimental realizar un análisis espectral de uno de los canales mediante el uso de la transformada rápida de Fourier o FFT.

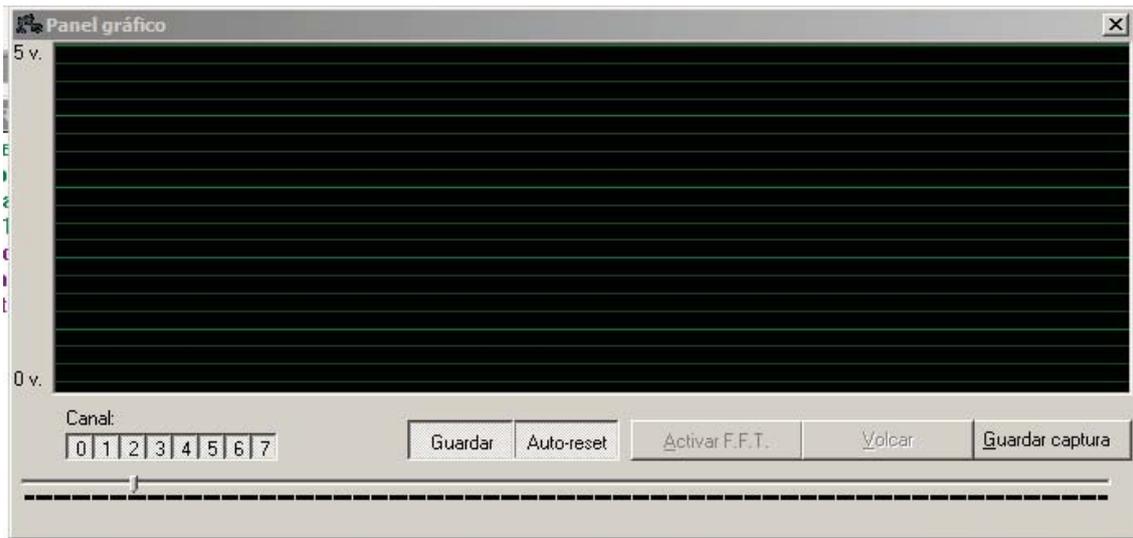


Ilustración 35 Panel gráfico

El último panel es el panel analógico que muestran en forma de barras de progreso verticales el valor analógico presente en los canales del robot.



Ilustración 36 Panel analógico

## 4. Resultados y valoración

---

La plataforma Aurebot 1.0 ya había sido probada con éxito durante dos ediciones de las Jornadas de Robótica de la ETSIIT, y la versión 2.0 durante una edición de estas también con éxito, pero las pruebas reales han sido la última convocatoria de la asignatura “Laboratorio de componentes electrónicos y fotónicos” de la Ingeniería de Telecomunicación y las tres convocatorias de la asignatura “Dispositivos electrónicos y fotónicos” del Grado de Ingeniería de tecnologías de Telecomunicaciones en las que los alumnos han tenido contacto directo con y sin supervisión con las plataformas.

En las cuatro convocatorias la plataforma utilizada ha sido la misma, con la modificación en la convocatoria 2012-13 del sistema de alimentación, dejando de lado las baterías de LiPo y usando pilas recargables LR6 por problemas de seguridad y simplicidad de mantenimiento.

Durante estas convocatorias los robots han pasado por las manos de mas de veinticuatro grupos que han realizado multitud de experimentos con ellos, sin mayores contratiempos que los cortocircuitos propios de las primeras experiencias con la electrónica.

Los resultados saltan a la vista recorriendo la escuela durante las últimas semanas del curso, pudiendo múltiples grupos de alumnos realizando pruebas con sus plataformas utilizando cualquier espacio libre que encontrarán.

Una vez llegan las presentaciones finales podemos comprobar que la experiencia ha dado sus frutos viendo la gran cantidad y diversidad de aplicaciones que han desarrollado:

- Asistentes de aparcamiento

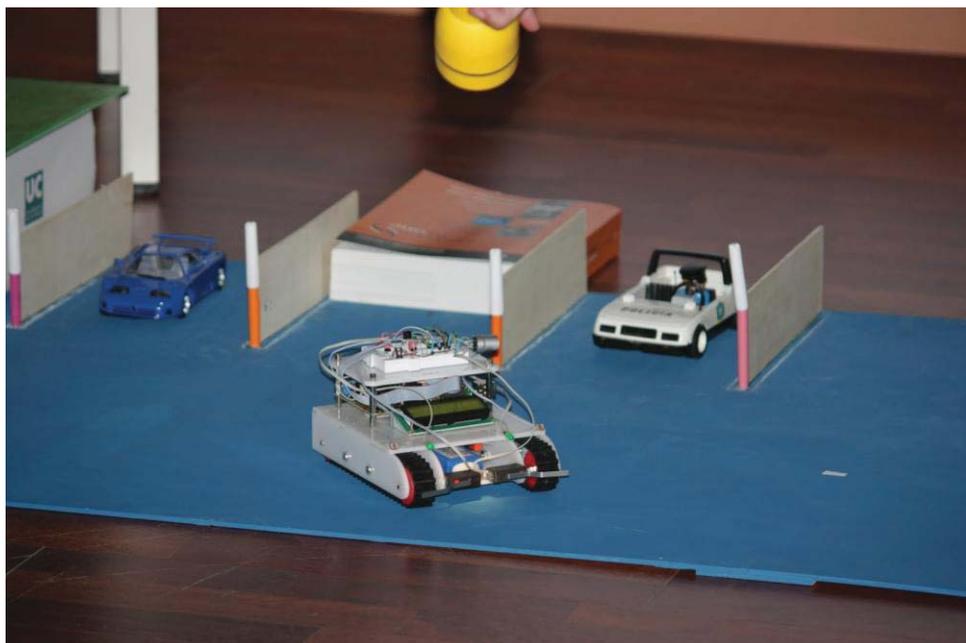


Ilustración 37 Robot aparcacoches

- Jugadores de fútbol controlados por mandos “remotos”



Ilustración 38 Robot futbolista

E incluso se han adoptado soluciones muy creativas para el problema de espacio en la placa de prototipos añadiendo estructuras.

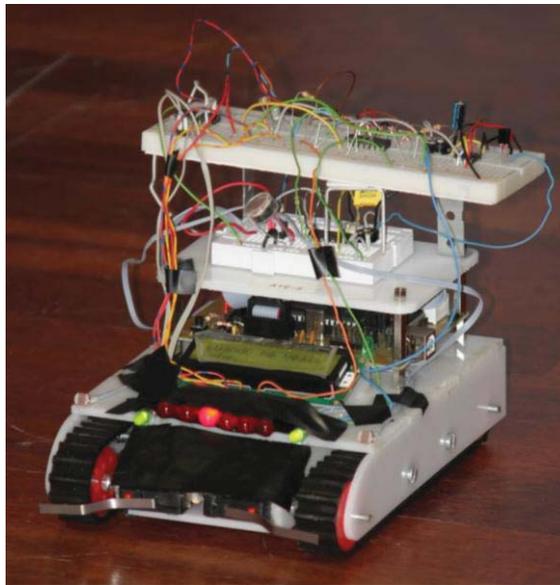


Ilustración 39 Estructuras añadidas

Otros proyectos dignos de mencionar han sido el robot policía Aurecop, robots con sintonías de videojuego como el SuperMarioBot y otros con nombre de androide de novela de Isaac Asimov como Robbie, capaz de apartar mediante un brazo obstáculos de su camino.

La valoración sobre la utilización de los robots en clase es muy satisfactoria, basta mencionar el hecho de que pese a la baja compensación en forma de nota de la práctica final y los grandes requerimientos de tiempo los alumnos han estado trabajando fuera de horas para tener todo listo para las presentaciones.

Una visión más crítica de los resultados nos la proporciona el estudio estadístico realizado por los profesores de la asignatura de grado durante la convocatoria 2011-12 (incluido en los anexos) y que nos dice que la percepción del alumnado sobre la plataforma es muy buena habiendo cumplido sobradamente los objetivos de la plataforma.

Según este estudio se les ha ayudado con la comprensión y retención de los conocimientos de la asignatura así como ha aumentado su motivación y mejorado sus capacidades de trabajo en grupo.

Además se ha reavivado la curiosidad por la electrónica e incluso se han divertido con las prácticas.

## 5. Conclusiones y líneas de futuro

---

Durante este proyecto se ha rediseñado la plataforma Aurebot para adaptarla a su uso continuado en docencia con el objeto de fomentar el trabajo en equipo y el aprendizaje experimental, construyendo ocho unidades completas y una adicional de repuesto.

Se le han añadido nuevos sensores y dispositivos y se han programado unas librerías para simplificar el desarrollo, además de una interfaz gráfica para Windows (funcional en Unix mediante Wine) para la programación y depuración de los programas.

El proyecto ha alcanzado satisfactoriamente todos los objetivos propuestos. No obstante, como todo sistema es susceptible de mejorar, detallamos algunos de los aspectos mejorables a continuación:

### 5.1. Baterías

---

Durante las clases se han observado algunos problemas con las baterías:

El principal, que afecta a la seguridad de los alumnos, es que se ha advertido una mala operación por parte de éstos en el uso del robot, accediendo a la parte de la electrónica que permanece tapada y provocando cortocircuitos en el conector de la batería y de la alimentación del driver de los motores. Esta tecnología de baterías no acepta cortocircuitos y se pueden llegar a producir quemaduras graves y explosiones debido a la gran capacidad de descarga de estas.

Asimismo, tampoco se han podido cumplir con las condiciones mínimas de uso y almacenamiento de las baterías (temperatura, descarga mínima, carga de almacenamiento) lo que ha provocado que varias de las baterías se encuentren en un estado no apto para su utilización y necesitando ser repuestas.

Estos problemas se han solucionado temporalmente sustituyendo las baterías por pilas recargables, aunque sería recomendable buscar algún otro tipo de tecnología (NiMH, NiCd, LiPo con menor capacidad de descarga) que permita seguir teniendo las claras ventajas que tiene el uso de baterías frente a las pilas.

### 5.2. Programación

---

Este sistema es utilizado por alumnos de primero de grado que en algunos casos han tenido un contacto mínimo con lenguajes de programación y algorítmica, lo que limita y dificulta el desarrollo de soluciones para las prácticas propuestas por parte de aquella parte del alumnado sin unos conocimientos medios de programación en C.

El desarrollo de una alternativa de programación gráfica, mediante bloques, solucionaría en gran medida este problema, se ha desarrollado una demo funcional de un programa online que permitiría pasar de programación de bloques a código C para PICS que posteriormente se puede incorporar a la programación del robot, no obstante, por falta de tiempo y por salirse de los objetivos del presente proyecto se deja como línea a desarrollar posteriormente.

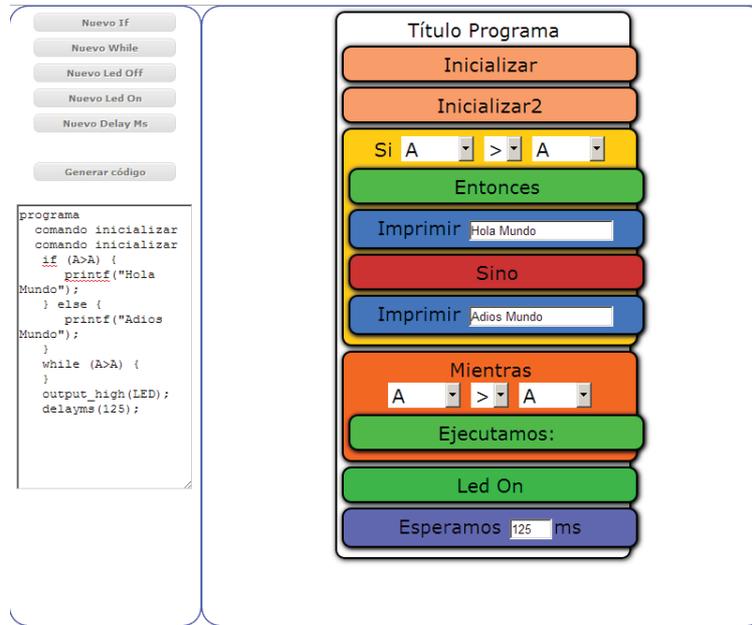


Ilustración 40 Captura de la demo de programación gráfica

### 5.3. Plataforma más actual

La aparición de nuevas plataformas embebidas como la Raspberry Pi dejan los microcontroladores como el 18F4550 en una posición bastante obsoleta, por velocidad de cálculo (12 Mhz reales frente a los 700Mhz), por posibilidades (USB Host con la infinidad de posibilidades que esto ofrece como conexiones wifi, sensores por usb, etc.; Ethernet, salidas y entradas de audio “real”) y coste (5€ del microprocesador frente a los apenas 27€ de la Raspberry comparando sus posibilidades)

Por esto, una de las posibles líneas de mejora sería el cambio de la placa hacia una plataforma embebida o la incorporación a la plataforma de un equipo embebido de bajo coste relegando la Aureboard al control de dispositivos y lectura de sensores.

## 6. Bibliografía

---

- «OCEANS'11 IEEE Santander Conference,» [En línea]. Available:  
1] <http://www.oceans11ieeesantander.org/>.
- Microchip, «Hoja de características PIC 18F4550,» [En línea].  
2]
- M. integrated, «Hoja de características MAX233,» [En línea]. Available:  
3] <http://www.maximintegrated.com/datasheet/index.mvp/id/1798>.
- S. Microelectronics, «Hoja de características L293D,» [En línea].  
4]
- F. Semiconductor, «Hoja de características del estabilizador LM7805,» [En  
5] línea]. Available: <http://www.fairchildsemi.com/ds/LM/LM7805.pdf>.
- Skybot. [En línea]. Available:  
6] <http://www.learobotics.com/proyectos/skybot/skybot.html>.

## 7. Anexos

### 7.1. Presupuesto construcción robots V1 – Jornadas 2008

#### *Componentes electrónicos*

Componente	Cantidad	Precio / Ud.	Total
Est. 7805	11	0,80 €	8,80 €
XTAL 4Mhz	11	0,84 €	9,24 €
condensadores 27p	22	0,20 €	4,40 €
condensadores 100n (plastico)	22	0,20 €	4,40 €
condensador 220u	11	0,20 €	2,20 €
condensador 47u	22	0,20 €	4,40 €
Diodo 1N4007	11	0,07 €	0,77 €
Led rojo 5mm	31	0,11 €	3,41 €
Led verde 5mm	31	0,11 €	3,41 €
Led amarillo 5mm	20	0,11 €	2,20 €
Resistencia 470 ohm	22	0,03 €	0,66 €
Resistencias 2k2 ohm (I2C)	22	0,03 €	0,66 €
Resistencias 150 ohm	60	0,03 €	1,80 €
Resistencias 10k ohm	99	0,03 €	2,97 €
Pulsador (PUL6X6VT)	22	0,24 €	5,28 €
Cont. macho para cable plano 10 hilos	44	0,27 €	11,88 €
Regletas clemas (REG2P)	11	0,24 €	2,64 €
Tira postes macho paso 2,54mm	88	0,46 €	40,48 €
Jumper (JMPN)	22	0,06 €	1,32 €
Tira de postes 3 pines paso 2,54mm (TIPOH3)	66	0,27 €	17,82 €
Zocalos 40 pines	11	0,21 €	2,31 €
Zocalos 20 pines	11	0,15 €	1,65 €
Zocalos 16 pines	11	0,12 €	1,32 €
Interruptor (INT1C)	11	0,89 €	9,79 €
Cont. hembra para cable plano 10 hilos	44	0,30 €	13,20 €
Portapilas (4)	11	1,19 €	13,09 €
Cable plano (1 metro)	11	0,46 €	5,06 €
CNY70	24	1,12 €	26,88 €
Resistencia 220	22	0,03 €	0,66 €
Bumpers	22	1,25 €	27,50 €
Resistencia 1K5	11	0,03 €	0,33 €
Condensador 100uF	11	0,13 €	1,43 €
Condensador 22uF	11	0,13 €	1,43 €
Diodo Zener 5v1	11	0,32 €	3,52 €
Diodo 1N4148	44	0,05 €	2,20 €
Diodo Zener 8v2	11	0,32 €	3,52 €
Transistor BC307	22	0,08 €	1,76 €
Resistencia 100k	11	0,03 €	0,33 €
Conector DB9 Hembra	11	1,02 €	11,22 €
Conector usb c.i.	15	0,74 €	11,10 €
REVREPRO (Revelador)	1	2,86 €	2,86 €
Placa perforada puntos	11	3,27 €	35,97 €
Placa cobre PFB200X300	6	14,82 €	88,92 €
portapilas (2)	11	0,44 €	4,84 €
Carrete estaño 1mm 60/40 c/flux de 250 grs.	3	5,96 €	17,88 €
CNT5PM	13	0,13 €	1,69 €
CNT5PH	13	0,16 €	2,08 €

Aurebot

Anexos

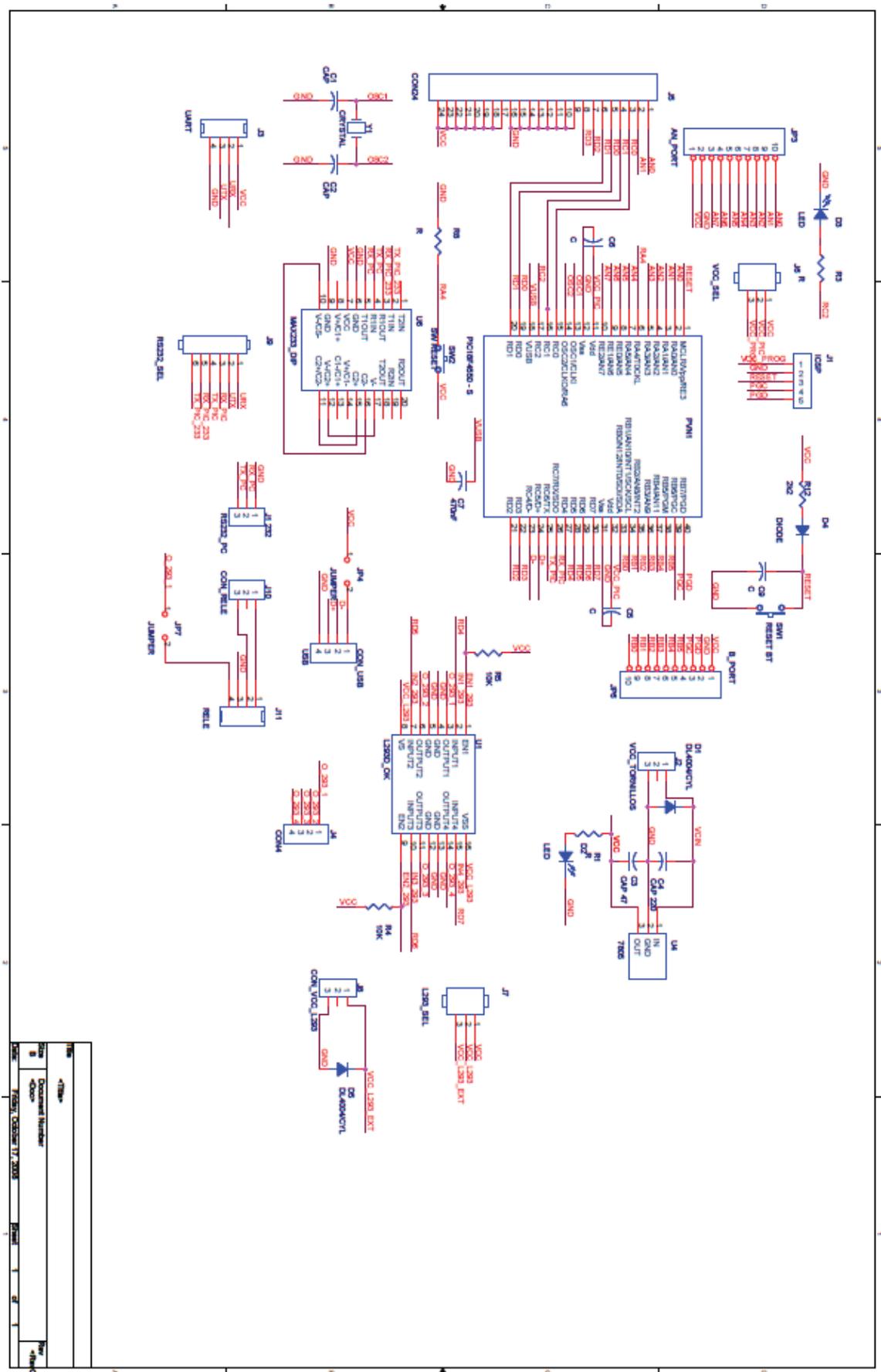
L293D	14	3,61 €	50,54 €
		Total	471,82 €
		Total + IVA	547,31 €

*Resto de materiales*

Componente	Cantidad	Precio	Total
BLISTER 4 PILAS ALCALINAS 1,5V TIPO AA (LR06)	17	1,85 €	31,45 €
SERVO ESTANDARD DY-S0210 6 KG	22	9,01 €	198,22 €
PAR RUEDA ROBOT 38 X 12	11	4,18 €	45,98 €
KIT 2 PANELES PVC 3 MM 30 X 40 CM AMARILLO	8	7,80 €	62,40 €
SEPARADOR HEXAGONAL M3 MACHO HEMBRA 10 MM 25 U	3	1,68 €	5,04 €
TORNILLO NEGRO DIN 7985 M3 X 6 MM 25 UNIDADES	3	0,30 €	0,90 €
		Total	343,99 €
		Total+I.V.A.	399,03 €

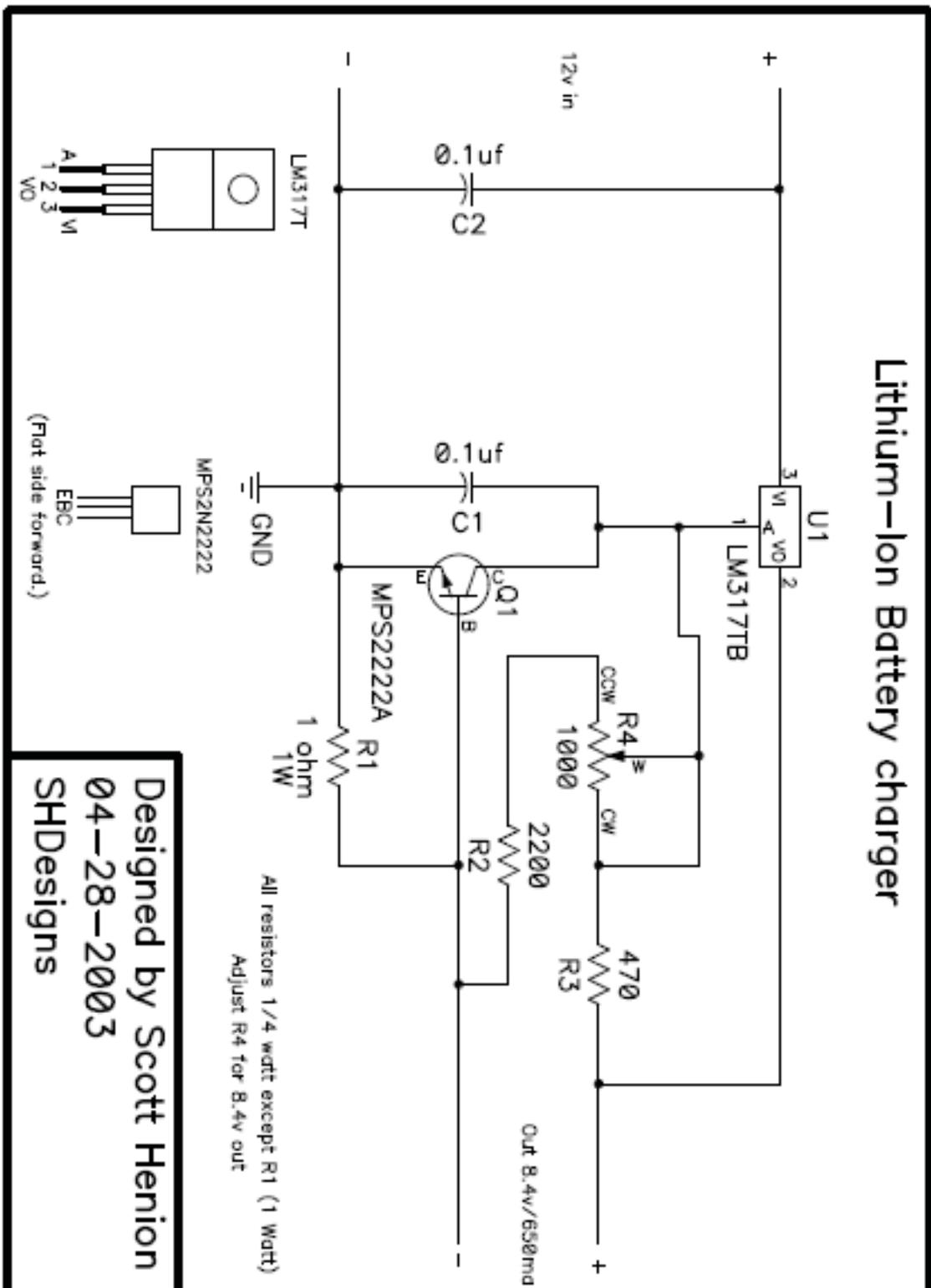
Total: 946,34€

## 7.2. Esquema de la Aureboard



Rev	<Rev>
Doc	Document Number
Doc	<Doc>
Doc	16C450-03/08/17/2008
Doc	16C450-03

### 7.3. Esquema del cargador de baterías de dos celdas LiPo



## 7.4. Resumen básico de C para PICs

---

### **Tipos de datos:**

int1 Entero de 1 bit -> 0 o 1  
 int8 Entero de 8 bits -> Sin signo: 0 a 255 Con signo: -127 a 127  
 int16 Entero de 16 bits -> 0 a 65.535  
 int32 Entero de 32 bits -> 0 a 4.294.967.295  
 char Caracter (8 bits) 'A'  
 float Coma flotante de 32 bits

Estos son nombres alternativos:

Short = int1  
 Int = int8

### **Declaración de variables:**

Signo Tipo\_variable nombre\_variable=valor\_inicial;

```
//Entero con signo con valor inicial cero.
Signed int balance=0;
```

```
//Entero sin signo con valor inicial 12.
Unsigned int dinero=12;
Int dinero=12; //Por defecto
```

### **Estructuras condicionales:**

#### **Condiciones:**

**A==B** Será verdadero si A es igual a B  
**A!=B** Será verdadero si A es distinto de B  
**A>B** Será verdadero si A es mayor que B ( A<B si es menor)  
**A>=B** Será verdadero si A es mayor o igual a B (A<=B si es menor o igual)  
**&&** AND, para que se cumplan 2 condiciones a la vez: (A>B && A==C) será verdadero si A es mayor que B y A es igual a C simultáneamente.  
**||** OR, para que se cumpla cualquiera de las condiciones (A>B || A==C) será verdadero si A es mayor que B ó A es igual a C.

#### **If-Else:**

```
If (condicion)
{
    //Si condicion es cierta se ejecuta esto.
} else {
    //Si condicion es falsa, se ejecuta esto.
}
```

#### **Switch:**

```
switch (a)
{
    case 0:
        //Si a=0 se ejecuta esto
        break;
    case 1:
        //Si a=1 se ejecuta esto
        break;
    case 'a':
        //También podemos usar letras, puesto que una letra esta
        representada por su código ASCII que es un número del 0 al 255. NO SE
        PUEDEN USAR PALABRAS
```

```

        break;
    default:
        //Si no se ha cumplido ninguna de las opciones, se ejecuta
esto
        break;
}

```

DEC	HEX	OCT	CHAR	DEC	HEX	OCT	CH	DEC	HEX	OCT	CH	DEC	HEX	OCT	CH
0	0	000	NUL	32	20	040		64	40	100	@	96	60	140	`
1	1	001	SOH	33	21	041	!	65	41	101	A	97	61	141	a
2	2	002	STX	34	22	042	"	66	42	102	B	98	62	142	b
3	3	003	ETX	35	23	043	#	67	43	103	C	99	63	143	c
4	4	004	EOT	36	24	044	\$	68	44	104	D	100	64	144	d
5	5	005	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
6	6	006	ACK	38	26	046	&	70	46	106	F	102	66	146	f
7	7	007	BEL	39	27	047	'	71	47	107	G	103	67	147	g
8	8	010	BS	40	28	050	(	72	48	110	H	104	68	150	h
9	9	011	TAB	41	29	051	)	73	49	111	I	105	69	151	i
10	A	012	LF	42	2A	052	*	74	4A	112	J	106	6A	152	j
11	B	013	VT	43	2B	053	+	75	4B	113	K	107	6B	153	k
12	C	014	FF	44	2C	054	,	76	4C	114	L	108	6C	154	l
13	D	015	CR	45	2D	055	-	77	4D	115	M	109	6D	155	m
14	E	016	SO	46	2E	056	.	78	4E	116	N	110	6E	156	n
15	F	017	SI	47	2F	057	/	79	4F	117	O	111	6F	157	o
16	10	020	DLE	48	30	060	0	80	50	120	80	112	70	160	p
17	11	021	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	DC2	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	DC3	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	DC4	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	NAK	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	SYN	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	ETB	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	CAN	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	EM)	57	39	071	9	89	59	131	Y	121	79	171	y
26	1A	032	SUB	58	3A	072	:	90	5A	132	Z	122	7A	172	z
27	1B	033	ESC	59	3B	073	;	91	5B	133	[	123	7B	173	{
28	1C	034	FS	60	3C	074	<	92	5C	134	\	124	7C	174	
29	1D	035	GS	61	3D	075	=	93	5D	135	]	125	7D	175	}
30	1E	036	RS	62	3E	076	>	94	5E	136	^	126	7E	176	~
31	1F	037	US	63	3F	077	?	95	5F	137	_	127	7F	177	DEL

**Bucles:****For:**

```

for (j=0; j<10; j++)
{
    //Esto se ejecutara desde j=0, mientras j sea menor de 10 y en
cada ciclo se suma 1 a j. 10 veces en total.
}

```

**while:**

```

while(condición)
{
    //Esto se ejecuta mientras condición sea cierta
}

do
{
    //Esto se ejecuta al menos 1 vez, y mientras se cumpla
condición.
} while (condición)

```

**Funciones:**

```

Tipo_que_devuelve nombre (tipo variables_que_recibe)

```

```

{
}

//Recibe enteros a y b y devuelve un entero con la suma
Int suma (int a, int b)
{
    int c=0;
    c=a+b;
    return c;
}

//No recibe nada ( () vacio) y tampoco devuelve nada (void)
Void main()
{
    adelante();
}

```

**Salida hacia el exterior:**

```
Output_high(pin); //Encender(pin)
```

Pone el pin a nivel lógico alto (5 v.).

```
Output_low(pin); //Apagar(pin)
```

Pone el pin a nivel lógico bajo (0 V.).

```
Output_toggle(pin); //Conmutar(pin)
```

Invierte el estado del pin.

```
Printf ("Hola mundo\r\n");
```

Envía por el puerto serie "Hola mundo", un retorno de carro "\r" y nueva línea "\n".

**Funciones específicas AUREboard:**

```
configurar_aureboard();
```

Se debe ejecutar al inicio y configura la tarjeta.

```
adelante(), atras(), izquierda(), derecha(), stop()
```

Hacen que el robot se mueva en la dirección deseada. (El robot se moverá en esta dirección hasta que llamemos a stop())

```
leer_adc(canal);
```

Nos devuelve el valor analógico del canal.

```
encender(pin), apagar(pin), conmutar(pin)
```

Realizan lo propio con los pines indicados.

**Retrasos, pausas**

```
delay_ms(100);
```

Hace una pausa de 100 mS.

```
delay_us(150);
```

Hace una pausa de 100 uS.

**Entrada desde el exterior:**

```
a=input(pin);
```

a Será verdadero si el pin tiene un estado lógico alto, o falso en caso contrario.

```
set_adc_channel (canal);
a=read_adc();
  Lee y nos devuelve el valor analógico del canal "canal".
  Con 0 V., a=0; con 5 V., a=255.
```

### **Esqueleto para aplicación**

```
// Incluimos las librerías, definiciones, configuraciones y
funciones del aurebot.
#include "../lib/aurebot.h"

//Incluimos otras librerías que vayamos a usar
#include "../lib/cny70.h" //Para configurar y leer los sensores
CNY70
#include "../lib/servos.h" //Para configurar y mover los servos
#include "../lib/sonar.h" //Para configurar y usar el sonar
#include "../lib/barra_leds.h"//Para configurar y usar la barra de
led
#include "../lib/sonido.h" //Para configurar y usar el altavoz

//Funcion principal
void main ()
{
  //Configuramos la aureboard
  configurar_aureboard();
  while (1)
  {
    //Ordenes
  }
}
```

### **Ejemplos simples**

```
//Encender el led
encender (LED);

//Encender el led si esta pulsado el pulsador
if (input (PULSADOR))
{
  encender (LED);
}
```

## 7.5. Valoración actividad AureBot 2011-12

**VALORACIÓN ACTIVIDAD AUREBOT 2011-12**  
**DISPOSITIVOS ELECTRÓNICOS Y FOTÓNICOS**  
**GRADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN**  
 UNIVERSIDAD DE CANTABRIA

Este informe resume los datos recabados de los alumnos. En esta edición, y a fecha de 16 de julio de 2012, han contestado 43 alumnos de los 96 que han realizado la actividad del Aurebot, lo que implica una respuesta del 45% del alumnado. La encuesta se realizó on-line, a criterio de los alumnos, en la que se preguntó sobre:

- Aspectos metodológicos sobre los videos preparatorios de la parte práctica de la asignatura, ya evaluados por Adolfo
- Valoración de la actividad de trabajo con la plataforma Aurebot.

Este informe recoge sólo los datos de la última parte donde tenían que valorar los siguientes ítems en una escala 1 (muy en desacuerdo)-2-3-4-5(muy de acuerdo):

A continuación pongo la nota media de cada item respecto a la escala 1-5 anterior:

A) Respecto al **AUREBOT**:

- Me ha ayudado a comprender los contenidos de la asignatura: 3,3
- Es preferible a las clases expositivas en el aula: 3,6
- Ha reavivado mi curiosidad hacia la electrónica: 3,6
- Ha permitido darme cuenta de mi necesidad de aprender: 3,5
- Ha hecho el aprendizaje más interesante y divertido: 3,8
- Me ha permitido retener e integrar mejor la información: 3,5
- Mi aprendizaje ha sido más profundo que en las clases expositivas de aula : 3,5
- Me he sentido más responsable de mi propio aprendizaje: 3,6
- Me he sentido mucho más implicado y motivado que en otras clases: 3,7
- Ha permitido plantearme cuestiones totalmente nuevas: 3,3
- He mostrado una actitud más activa y participativa que en otras clases: 3,7
- Mi rendimiento académico se ha visto perjudicado: 3,4
- Me hubiera gustado haberle dedicado más tiempo: 3,9

B) Respecto al **TRABAJO EN EQUIPO**:

- En grupo he aprendido más que realizando la actividad en solitario: 3,4
- He tenido que abordar tareas de equipo y distribución de trabajo: 3,6
- He sido más tolerante con las opiniones de los demás: 3,7
- He sido más responsable tanto individual como respecto al grupo: 3,5
- Hemos podido superar muchas dificultades y, aunque a veces sea frustrante, esto me ha motivado: 3,5
- He trabajado más que mis compañeros: 3,0

C) Respecto a la **MOTIVACION** para el trabajo:

- Lo que más me ha motivado para trabajar es la nota a conseguir: 3,4
- Lo hubiera hecho de forma optativa aunque la nota no importara: 2,7

Ordenando de mayor a menor el grado de acuerdo con las afirmaciones tenemos:

### 5: Muy de acuerdo

3,9 Me hubiera gustado haberle dedicado más tiempo

3,8 Ha hecho el aprendizaje más interesante y divertido

3,7 He sido más tolerante con las opiniones de los demás

3,7 He mostrado una actitud más activa y participativa que en otras clases

3,7 Me he sentido mucho más implicado y motivado que en otras clases

3,6 Ha reavivado mi curiosidad hacia la electrónica

3,6 He tenido que abordar tareas de equipo y distribución de trabajo

3,6 Es preferible a las clases expositivas en el aula

3,6 Me he sentido más responsable de mi propio aprendizaje

3,5 Hemos podido superar muchas dificultades y, aunque a veces sea frustrante, esto me

ha motivado

3,5 He sido más responsable tanto individual como respecto al grupo

3,5 Ha permitido darme cuenta de mi necesidad de aprender

3,5 Me ha permitido retener e integrar mejor la información

3,5 Mi aprendizaje ha sido más profundo que en las clases expositivas de aula

3,4 En grupo he aprendido más que realizando la actividad en solitario

3,4 Mi rendimiento académico se ha visto perjudicado

3,4 Lo que más me ha motivado para trabajar es la nota a conseguir

3,3 Me ha ayudado a comprender los contenidos de la asignatura

3,3 Ha permitido plantearme cuestiones totalmente nuevas

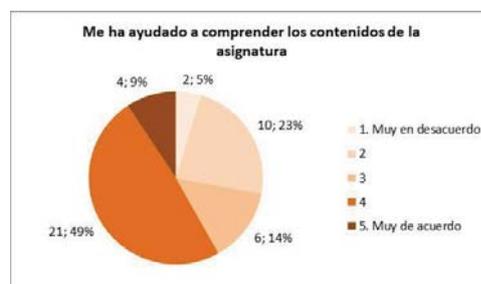
3,0 He trabajado más que mis compañeros

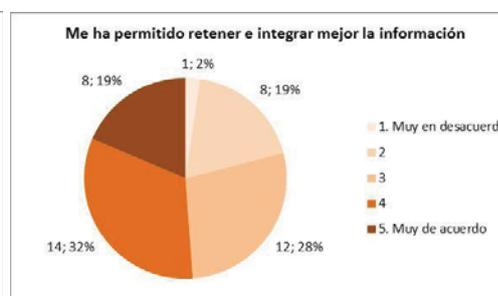
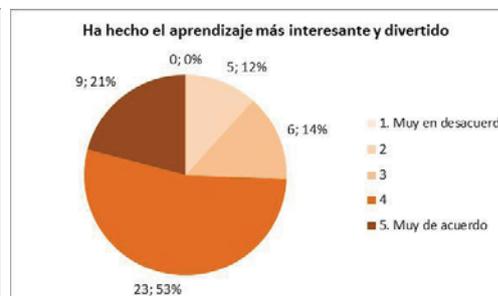
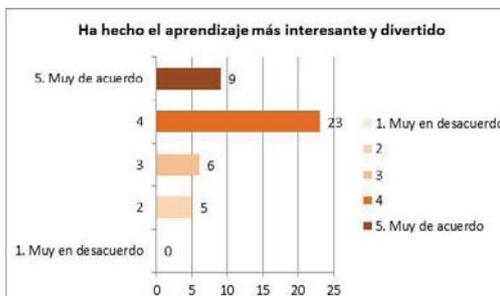
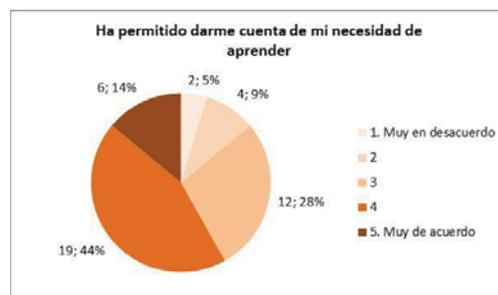
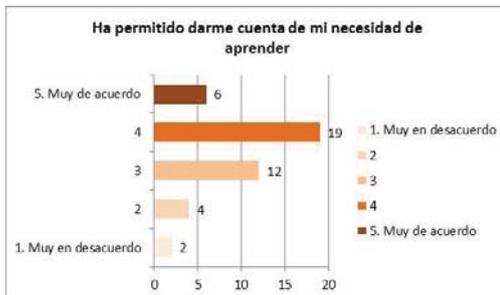
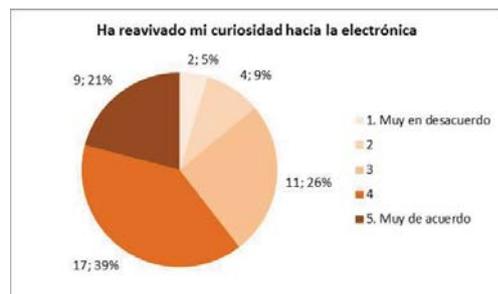
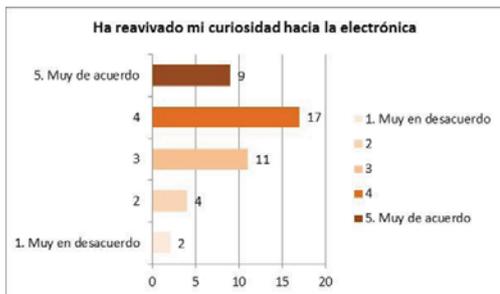
2,7 Lo hubiera hecho de forma optativa aunque la nota no importara

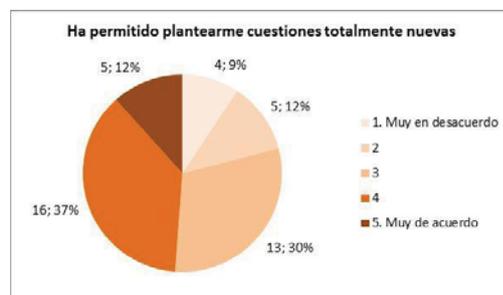
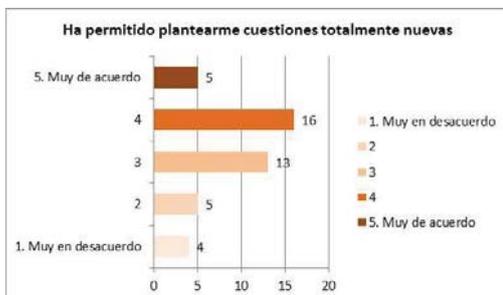
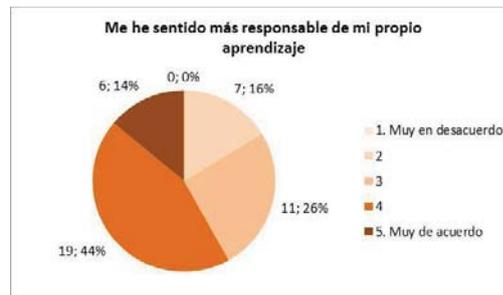
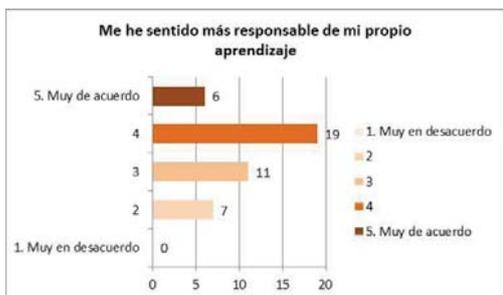
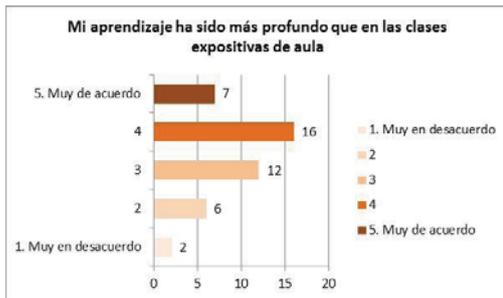
### 1: Muy en desacuerdo

A continuación se mostrarán las gráficas de valoración de cada uno de los ítems para tener una visión más completa del análisis.

#### A) Respecto al AUREBOT:

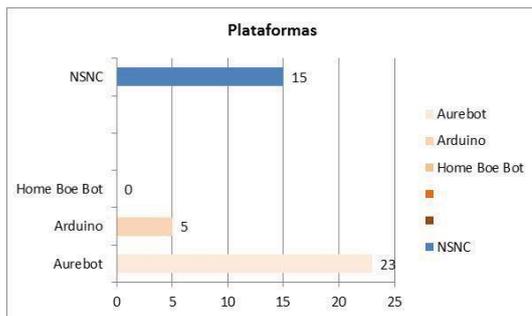






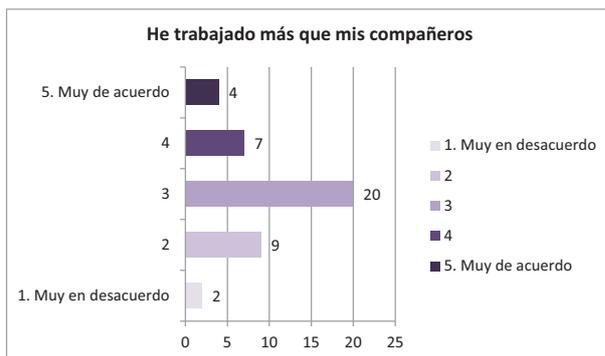
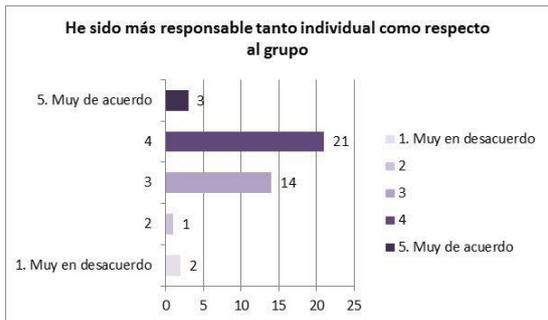
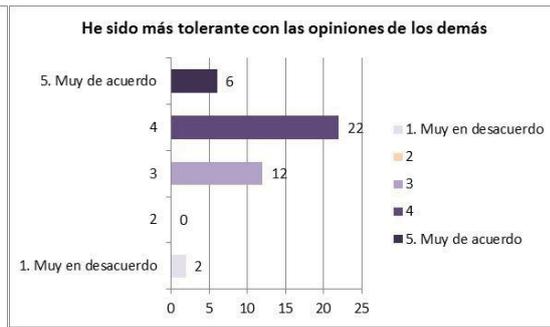
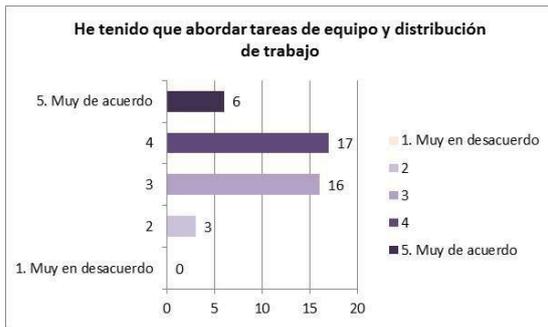


**B) Respecto a las PLATAFORMAS:**



**C) Respecto al TRABAJO EN EQUIPO:**





D) Respecto a la MOTIVACION:

