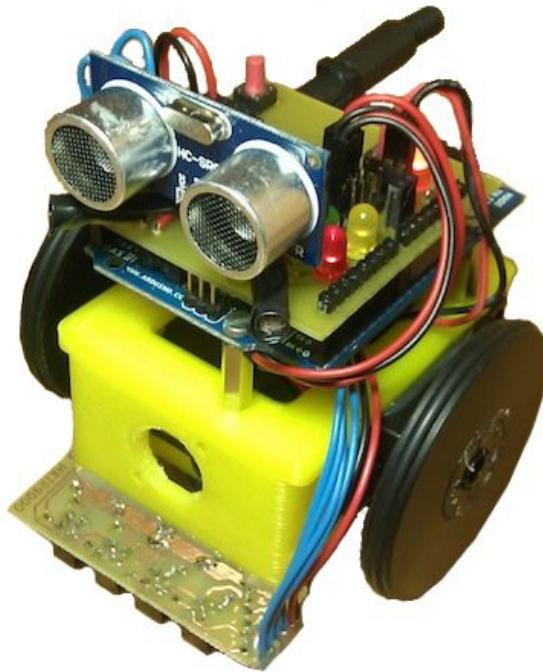


Trabajo de investigación

Robots educativos libres, imprimibles y de bajo coste



Realizado por:
Carlos García Saura

Tutora:
M^a Carmen San José



2012

Universidad Autónoma de Madrid

www.carlosgs.es/robots

Título:

Robots educativos libres, imprimibles y de bajo coste

Autor:

Carlos García Saura

*Presentado al Certamen Universitario
Arquímedes 2012
de Introducción a la Investigación Científica*

Índice

Introducción.....	1
Descripción.....	1
Metodología.....	2
Justificación.....	2
Objetivos.....	3
Parte I: Antecedentes.....	4
1. La electrónica al alcance de todos: El proyecto Arduino.....	4
1.1 El concepto de “Shield”.....	7
1.2 Software para el diseño electrónico.....	8
1.3 Sensores.....	9
1.4 Actuadores.....	10
2. Impresoras 3D: De bits a objetos.....	12
2.1 Materiales utilizados.....	13
2.2 Proyecto RepRap.....	14
2.3 Proyecto Clone Wars.....	14
2.4 OpenScad y la revolución del diseño 3D.....	18
3. Printbots: Robots libres e imprimibles.....	20
3.1 MiniSkybot.....	21
3.2 Evoluciones.....	21
3.3 Usando servos como motores de rotación continua.....	22
Parte II: El robot HKTR-9000.....	24
1. Resumen del taller.....	24
1.1 Opinión de los estudiantes.....	25
2. Especificaciones del robot.....	26
2.1 CRM-Shield.....	27
2.2 PCB con los sensores de línea.....	28
2.3 Sensor de distancia.....	29
2.4 Conexiones eléctricas.....	29
Parte III: Mejoras propuestas.....	31
1. Soporte para los sensores de luz.....	31
2. Integración del sensor de distancia.....	32
3. Uso de pulsadores en el shield.....	34
4. Integración de un altavoz.....	34
5. Uso de servomotores de rotación continua.....	35
6. Ruedas y bola de apoyo imprimibles.....	36
Parte IV: Diseño experimental.....	37
1. Printshield.....	37
1.1 Sensores LDR.....	38
1.2 Sensor de distancia.....	39
1.3 Servomotores de rotación continua.....	40
1.4 Pulsadores, diodos LED y altavoz.....	40
2. Diseño hardware.....	41
3. Coste del ArduSkyBot.....	44
4. Diseños y documentación realizados.....	45
Conclusiones.....	46
Agradecimientos.....	47
Fuentes, imágenes y licencia.....	48

Introducción

Este trabajo de investigación ha sido realizado por **Carlos García Saura**, estudiante de segundo curso del Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación, en la Universidad Autónoma de Madrid, orientado por M^a Carmen San José.

Descripción

La robótica educativa requiere plataformas que sean sencillas de utilizar, siendo a la vez muy completas y manteniendo un precio reducido; pero en muchas ocasiones es necesario dejar de lado al menos uno de estos aspectos.

En este trabajo de investigación se propone el uso de plataformas educativas de hardware libre.

Gracias a la tecnología de las impresoras 3D open-source que se han desarrollado en los últimos años, cualquiera puede disponer de una impresora 3D en su casa y fabricar sus propias piezas.

Los “printbots” son robots libres e imprimibles, es decir, existe una comunidad abierta que comparte todos los diseños y cualquiera puede fabricarlos utilizando una impresora 3D casera.

Se trata el uso de los printbots como elemento educativo, concretamente el robot HKTR-9000 y de su uso en un taller con estudiantes universitarios.

Después se proponen mejoras respecto a los printbots educativos, que culminan en el desarrollo del Printshield, una nueva plataforma educativa de hardware abierto y bajo coste basada en Arduino.

Por último, se demuestra el funcionamiento del ArduSkyBot, un printbot basado en el Printshield, y se exploran sus posibilidades educativas con vista a futuros talleres.

Toda la documentación puede encontrarse en <http://www.carlosgs.es/robots>

Educational robotics requires platforms that are simple to use, and at the same time full-featured and low-cost; but in many occasions some of these requirements must be left apart.

This research work proposes using open-hardware educational platforms.

Thanks to the open-source 3D printing technology that has been developed in recent years, anyone can have a 3D printer at home and manufacture his own parts.

“Printbots” are open-source robots that are printable, which means that they have an open community that shares every design, and anyone can build them using a 3D printer.

We also cover the usage of printbots as an educational element. Specifically, the HKTR-9000 robot and its usage on a workshop with college students.

Then we propose some improvements regarding educational printbots, that culminate with the development of the Printshield, a new educational platform which is Arduino-based, open-hardware and keeps a low cost.

Finally, we demonstrate the operation of a Printshield-powered printbot: the ArduSkyBot. We also research its educational possibilities regarding future workshops.

All the documentation can be found in <http://www.carlosgs.es/robots>

Metodología

- **Recopilación y estudio de la información:** Primero es imprescindible sintetizar el estado del arte respecto a todas las ramas de la ingeniería necesarias (robots educativos, electrónica, impresoras 3D... etc).
- **Proponer mejoras:** Una vez recopilada toda la información relevante, se deben idear nuevas posibilidades que investigar, y aportar alguna idea o mejora innovadora.
- **Experimentación:** Las primeras pruebas de concepto, en base a los datos adquiridos. Este paso es útil para poder llevar a cabo prototipos funcionales de las mejoras propuestas.
- **Diseño experimental:** Con la información recopilada y su experimentación se elabora el diseño y se construye un robot que demuestre la viabilidad de los objetivos de este trabajo de investigación.

Justificación

Como experiencia personal, conozco a multitud de estudiantes y profesionales aficionados a la robótica que manifiestan un gran interés por ella pero no le dedican tiempo.

En general, el argumento que mas utilizan es que les resulta una afición cara, o bien no reciben los estímulos necesarios en sus estudios, por falta de información... el caso es que **no le dedican tiempo a la robótica.**

También es muy común que cuando alguien decide construir su primer robot, lo haga con kits comerciales que normalmente tienen un precio elevado, o bien limitan mucho los resultados que puede conseguir con el robot. Además, **la mayoría de estos kits comerciales son plataformas privativas, haciendo muy difícil su extensión en cuanto al hardware, la mecánica, y en ocasiones incluso el software.**

Alguien que se ve forzado a seguir las especificaciones exactas dictadas por los fabricantes, ve limitada enormemente su creatividad... y **limitar la creatividad implica la pérdida de interés por la robótica.**

Esto fue así durante un tiempo, no hubo una alternativa barata y sencilla a la “robótica comercial”...
¡Pero esto ha cambiado!

Desde hace algunos años, se ha ido abaratando en gran medida la tecnología de las **impresoras 3D open-source**, lo que ha permitido que estas se introduzcan en diversas universidades, grupos de

investigación, laboratorios, institutos, e incluso en nuestros hogares.

Los **robots imprimibles** han supuesto un gran cambio: ahora es posible **desarrollar robots bajo demanda**, sin necesidad de recurrir a caros equipos comerciales.

Pero los robots libres e imprimibles suponen además una revolución en el paradigma de la robótica: comienzan a formarse proyectos de hardware abierto que evolucionan gracias a la comunidad. Conforme mas personas trabajan con estos diseños, van apareciendo mejoras en ocasiones muy ingeniosas, procedentes de cualquier parte del mundo.

En cuanto a aspectos didácticos, **fabricar un robot imprimible permite desarrollar competencias tan importantes como el diseño mecánico y electrónico, su fabricación (soldadura, impresión 3D, ensamblado...), programación, lectura y ajuste de sensores... etc.** En definitiva, se trata de un proyecto muy interesante para cualquier estudiante universitario o aficionado.

Estos son los motivos por los que considero necesario seguir investigando y documentando los robots educativos libres e imprimibles: para conseguir reducir su coste y acercarlos a todos los estudiantes y aficionados a la robótica.

Objetivos

- Sintetizar y exponer el “estado del arte” referente a los robots libres junto con las impresoras 3D, enfocados a la educación.
- Elaborar nuevos diseños, tanto mecánicos como electrónicos, manteniendo la simplicidad y reduciendo costes sin comprometer sus prestaciones.
- Construir los prototipos ideados y demostrar su viabilidad.
- Documentar y liberar los diseños, para ponerlos a disposición de todo aquel interesado en aprender a construir un robot.

Parte I: Antecedentes

Primero voy a aclarar los conceptos que usaremos a lo largo del proyecto, comenzando por:

1. La electrónica al alcance de todos: El proyecto Arduino

Todo robot necesita una electrónica que muchas veces está basada en un **microcontrolador programable**. Es posible diseñar por completo la electrónica de cada robot para adaptarla a sus características, lo que resulta en un diseño óptimo y elegante.

El problema de diseñar la electrónica a muy bajo nivel es que resulta complejo (tanto la parte hardware como software) para quien se inicia en la robótica.

Nosotros vamos a distinguir entre la placa controladora y la electrónica:

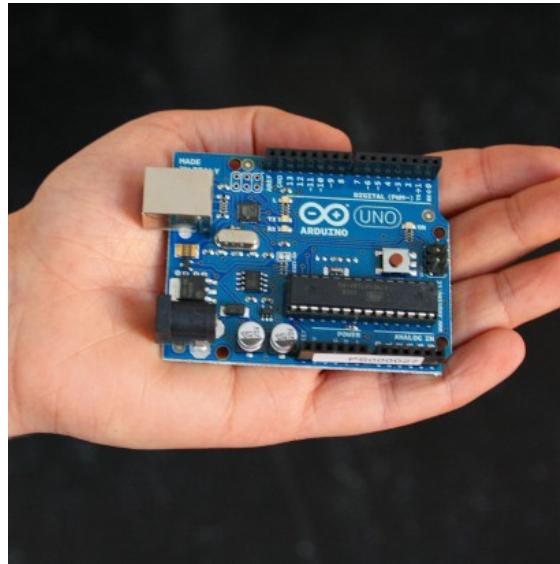
- La **placa controladora** es la que se puede conectar y programar con un ordenador. Genera señales eléctricas de control según el programa que tenga cargado y las señales recibidas por la placa (a través de la electrónica).
- La **electrónica** engloba del resto de componentes necesarios para el robot (interruptores, resistencias, diodos, motores, indicadores, conectores... etc). Recibe las señales de control de la placa controladora y actúa en consecuencia, pero sin procesar la información.

La electrónica sin una placa controladora no tiene demasiada utilidad, al igual que la placa controladora por si sola no es útil a efectos prácticos. Dependen completamente una de la otra.

Para entender esto, y como resumen: **la placa controladora como el “cerebro” y la electrónica como el “sistema sensorial y motor” de nuestro robot.**

Existe una gran variedad de placas controladoras tanto propietarias como libres. Normalmente cada placa dispone de un software desarrollado por la misma entidad que lo diseñó. El conjunto placa controladora y entorno de programación (IDE) constituyen una **plataforma** de desarrollo.

Nosotros vamos a usar la plataforma **Arduino** como placa controladora.

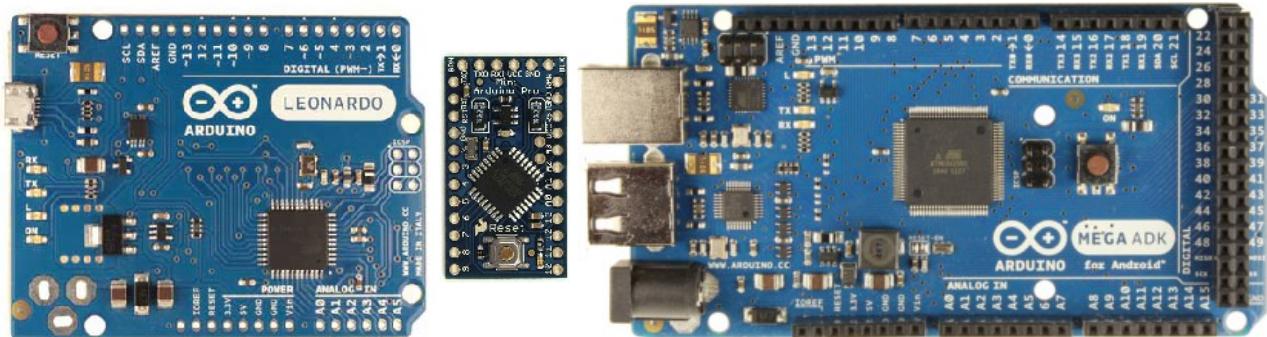


Placa Arduino Uno

Los motivos más destacados que nos hacen elegir esta plataforma son:

- Se trata de placas muy completas y sencillas de usar.
- Programable fácilmente por USB.
- Es hardware y software abierto (y multiplataforma).
- La documentación incluye multitud de ejemplos y funciones ya desarrolladas.
- Tiene una gran difusión en el ámbito educativo, y mucha gente tiene acceso a ella.
- Útil tanto como para principiantes como para expertos en robótica.

Algunas de las placas oficiales del proyecto son: Arduino UNO, Arduino Leonardo, Arduino ProMini, Arduino Mega ADK....

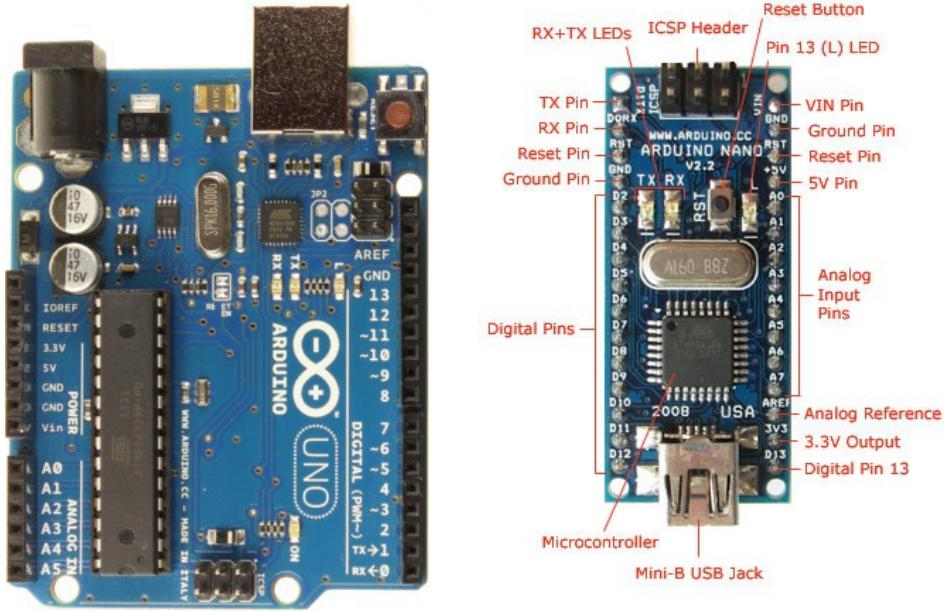


Varias placas Arduino

Al tratarse de hardware libre, cualquiera puede diseñar y fabricar su propio modelo compatible con

el software Arduino (y de hecho existen muchas variedades diferentes: Sanguino, SkyMega... etc).

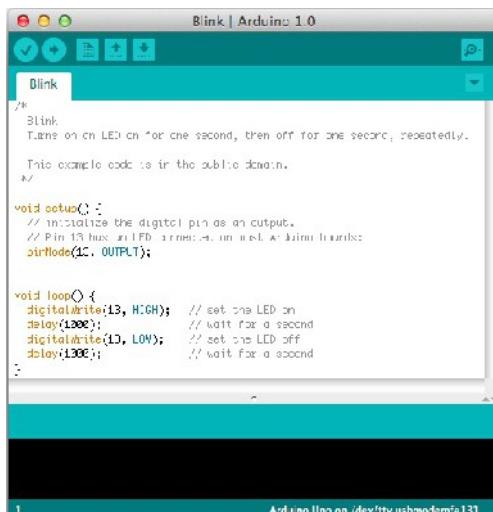
Cada modelo de placa tiene diferentes características. Esta gran variedad de placas Arduino la convierte en una plataforma ideal para cualquier tipo de proyecto electrónico.



Pines disponibles en la Arduino UNO y en la Arduino Nano

Todos los modelos de Arduino disponen de alimentación, pines tanto digitales (de entrada y salida, incluyendo PWM), como de lectura analógica, y puertos de comunicación como USB, serie, I²C, Wire... etc.

En cuanto a la programación, el IDE de Arduino se caracteriza por su **simplicidad**:



Arduino IDE con el ejemplo “Blink” cargado

Incluye una ventana con los ficheros de código del programa distribuidos en pestañas, y dispone de una “consola serie” para enviar y recibir datos de la placa.

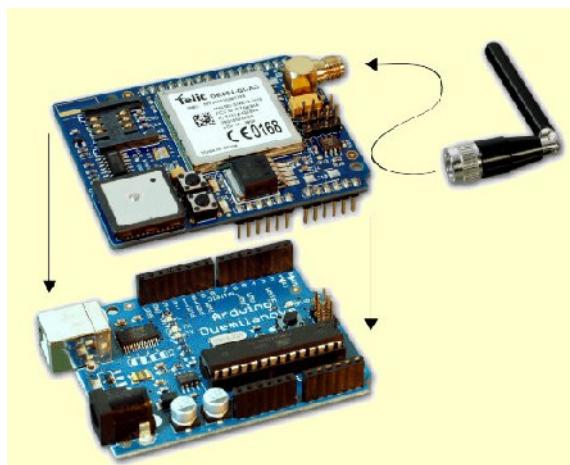
Debido a que está basado en Processing (<http://processing.org/>), se puede utilizar directamente el **lenguaje C**; perfectamente documentado y el mas recomendado para iniciarse en programación. De todos modos, Arduino se puede programar en una gran variedad de lenguajes (C++, Java... etc).

Además, cualquier lenguaje que permita acceso al puerto serie podrá comunicarse a través del USB con una placa Arduino (previa carga del programa correspondiente).

Hay mucha mas información sobre Arduino en su página web oficial: <http://www.arduino.cc/>

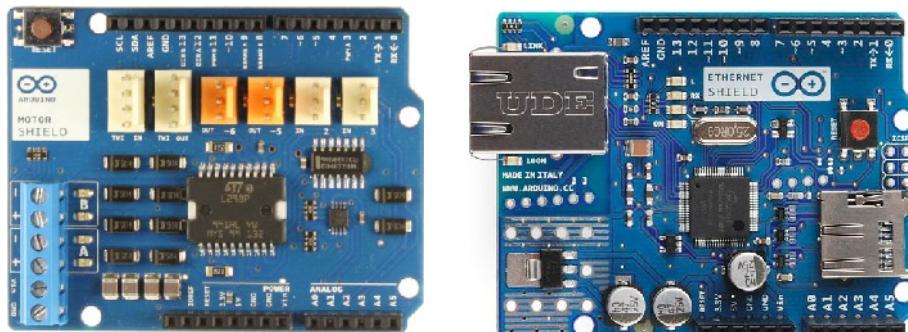
1.1 El concepto de “Shield”

Se denominan “Shields” a los **circuitos impresos que están diseñados para que se puedan conectar directamente a alguna de las placas Arduino**.



Esquema de conexión de un shield, en este caso se trata de un receptor GPS y red móvil

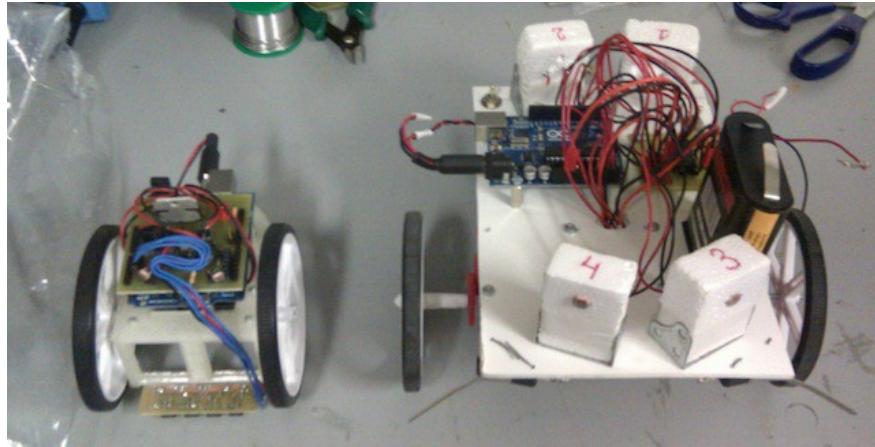
Hay multitud de diseños disponibles en internet, con funciones muy diversas.



Algunos ejemplos de shields: controladora de motores y tarjeta ethernet

La ventaja de tener la electrónica ensamblada en un “shield” es principalmente el **reducido tamaño**

que se consigue. Para las pruebas está bien usar una placa de prototipos, pero una vez probado el diseño electrónico, es mucho mas conveniente la fabricación de un “shield” que reduzca el tamaño y haga más elegante la electrónica final.

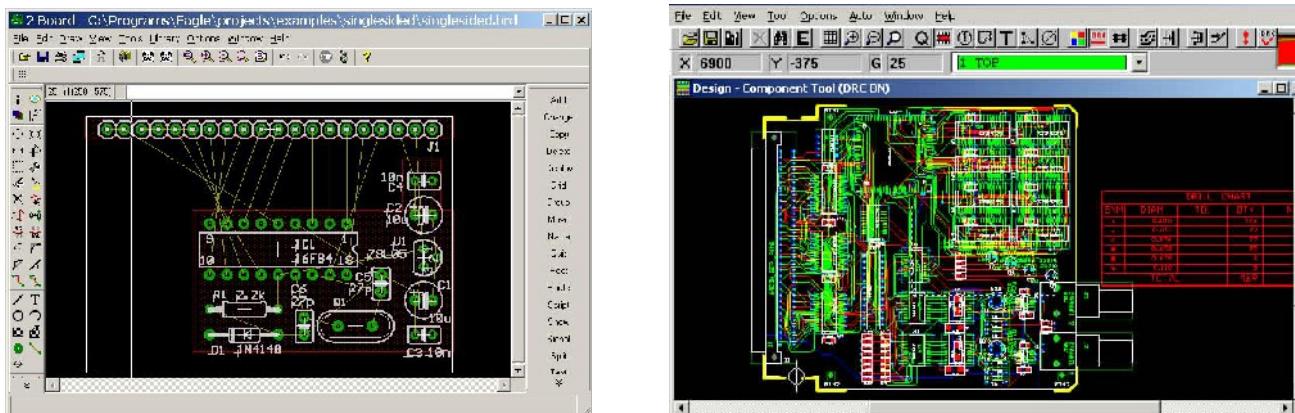


A la derecha se ve el diseño electrónico prototipado y a la izquierda el robot final usando el CRM-Shield

Además, otra de las ventajas es que una vez está ensamblado y funcionando el “shield” junto con la placa Arduino, al estar todo integrado y soldado, es mucho menos probable cometer un error en alguna conexión de la electrónica.

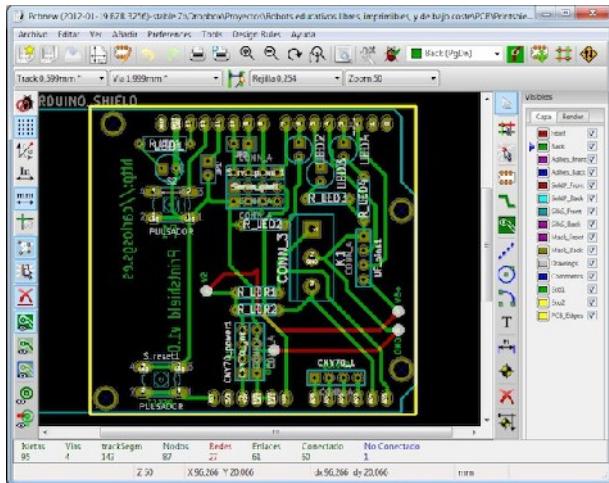
1.2 Software para el diseño electrónico

Existen diversos programas disponibles para el diseño de circuitos electrónicos. Los más conocidos son Eagle y OrCAD.



Diseño de circuitos impresos con Eagle y OrCad

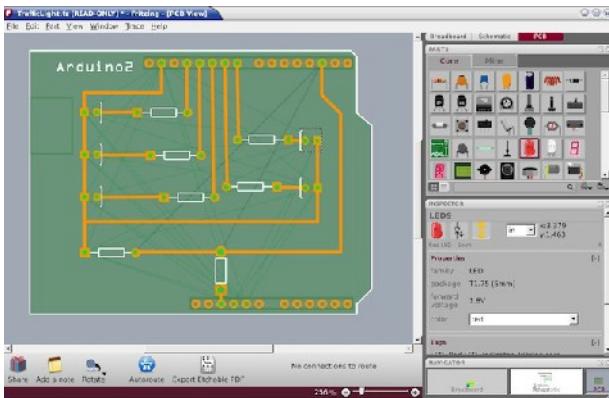
Estos programas son comerciales y privativos. La principal alternativa de código abierto a estos programas es **Kicad**.



Kicad mostrando el diseño de un shield para Arduino. En la página <http://www.carlosgs.es/robots> he enlazado un video-tutorial sobre Kicad realizado por Juan Camilo Fajardo.

Existen multitud de librerías de componentes disponibles para Kicad. Incluso las hay específicas para el diseño de los “shields” para Arduino (<http://www.thingiverse.com/thing:9630>).

Otro programa libre de diseño electrónico más reciente es Fritzing (<http://fritzing.org/>)



Se trata de un software muy sencillo de usar que Kicad, pero bastante completo enfocado a diseños con Arduino. En este trabajo he usado Kicad por falta de experiencia con este nuevo software.

1.3 Sensores

Denominamos “sensores” a los periféricos de los cuales podemos **leer información** desde nuestro programa. Traducen la información sobre diferentes aspectos físicos del mundo real a información eléctrica.

Hay una gran variedad de tipos de sensores: de luz (tanto visible como infrarroja), de distancia (tanto por infrarrojos como por ultrasonidos), de humedad, de presión atmosférica, pulsadores, sensores magnéticos, giroscopios... y un largo etcétera.



Sensores para medir humedad, sonido, distancia, contacto, aceleración y luminosidad

Para comunicarnos con estos sensores, podemos recibir señales de distintas maneras:

- **Señales de voltaje variable:** tanto discretas (p.e. Pulsadores) como continuas (p.e. Luz ambiente, distancia...)
- **Ancho de pulsos:** Hay sensores que devuelven el valor como un pulso eléctrico de duración proporcional al valor leído.
- **Digital:** Usando algún tipo de codificación, normalmente binaria, usando protocolos como Serie, I²C, One-wire...

La plataforma Arduino dispone de la mayoría de estas funciones de lectura ya implementadas (AnalogRead, PulseIn, Serial.Read...), lo cual simplifica mucho el trabajo de incorporar nuevos sensores a nuestros proyectos.

1.4 Actuadores

Se denominan “actuadores” a todos aquellos periféricos cuyo comportamiento controlamos desde nuestro programa. Traducen las señales eléctricas que generamos desde nuestro código a **acciones mecánicas y físicas** sobre el mundo real.

También hay una gran variedad de actuadores: motores rotativos o lineales, servomotores, emisores de luz, altavoces, generadores de campos magnéticos y eléctricos... etc



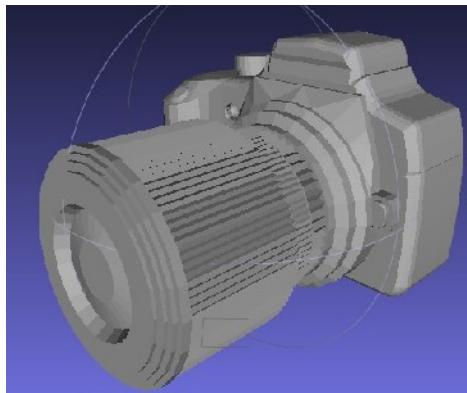
Actuadores: diodo LED, servomotor, motor con reductora, altavoz

La comunicación con los actuadores se lleva a cabo de diversas maneras:

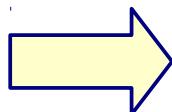
- **Señales discretas:** Encendido/Apagado, 5V/0V. El chip principal de una Arduino solo puede proporcionar unos pocos miliamperios en sus pines de entrada y salida. Por ello, en muchas ocasiones se necesita un “**driver de potencia**” que permita proporcionar una **mayor corriente** a los actuadores. Este driver puede ser desde un simple transistor hasta dispositivos integrados mas complejos (como el L293, L298...).
- **Ancho de pulsos:** Es importante hacer notar que las placas Arduino NO pueden generar señales de voltaje variable, sino que las “simulan” alternando muy rápidamente los estados encendido y apagado. Esto se denomina “modulación por ancho de pulsos”, y también puede usarse conjuntamente con un driver de potencia.
- **Digital:** Usando algún tipo de codificación, normalmente binaria, usando protocolos como Serie, I²C, One-wire...
- **Salidas combinadas:** Por ejemplo, si queremos controlar un motor a través de un driver, podemos establecer su sentido de giro de forma discreta (derecha, izquierda), y su velocidad usando ancho de pulsos (desde parado hasta el máximo posible).

2. Impresoras 3D: De bits a objetos

Las impresoras 3D son máquinas capaces de **convertir diseños realizados en el ordenador en objetos físicos**. Por ejemplo, un modelo 3D de una cámara de fotos puede ser impreso en un objeto real. En este caso fue necesario dividir el modelo en varias partes que luego fueron unidas.



Modelo 3D



Impresión 3D

Pieza propia impresa en 3D

Estas impresoras existen desde los años 80, y desde el principio se usaron para un **prototipado rápido e investigación**. Al contrario que las tecnologías sustractivas de material (tornos, taladros... etc), se trata de una **tecnología “aditiva”**. Esto quiere decir que se depositan sucesivamente **capas del material**, formando la pieza de forma progresiva.

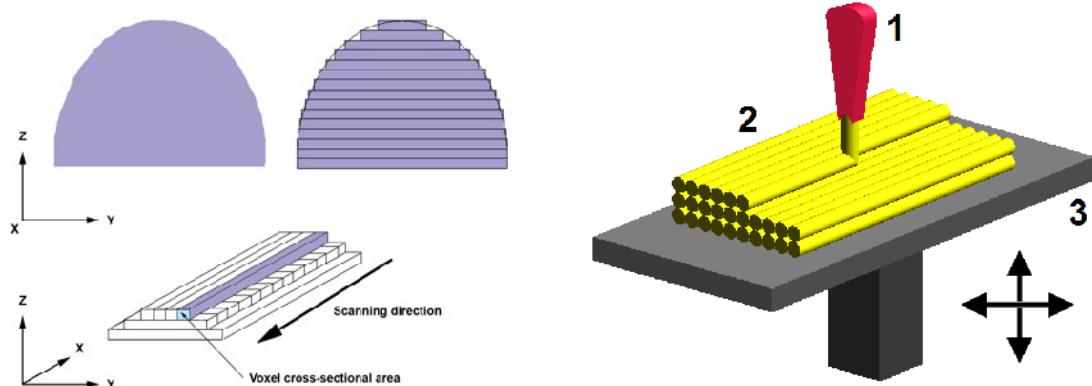


Modelos comerciales de impresoras 3D



Algunos de los métodos de fabricación son:

- Sinterizado selectivo por láser (SLS).
- Estereolitografía.
- Impresoras 3D de tinta.
- Modelado por deposición de material fundido (FDM).



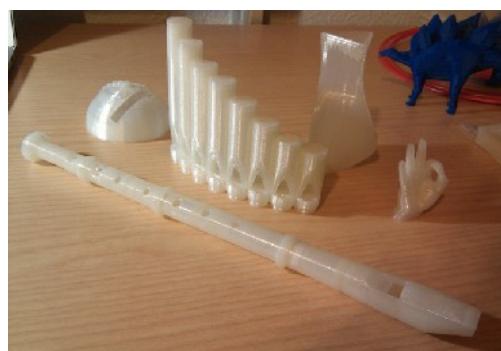
Diagramas que muestran el laminado de las piezas (izquierda) y el método FDM (derecha)

Tanto las impresoras 3D de tinta como el SLS utilizan polvo de material, que cohesionan capa por capa usando tinta o un láser. La estereolitografía utiliza un proyector de luz ultravioleta y un material líquido sensible, que solidifica por capas en las zonas iluminadas.

2.1 Materiales utilizados

Actualmente existen muchos proyectos de hardware abierto para la fabricación de impresoras 3D, y la tecnología que se impone en este ámbito es el **Modelado por deposición de material fundido (FDM)**, ya que ofrece la mejor relación resultado-precio. Los materiales más utilizados en esta técnica son:

- **Plástico ABS**, obtenido a partir del petróleo. Se usa fundido a 228°C y sobre una base calentada previamente a 110°C para mejorar la adhesión.
- **Plástico PLA**, que se sintetiza a partir de fuentes de azúcar naturales y renovables como el almidón de maíz. Es biodegradable y se utiliza fundido a 195°C. No requiere la impresión sobre base caliente, pero es recomendado utilizarla a 60°C.



Algunas de las piezas que he impreso tanto en ABS azul (izquierda) como en PLA natural (derecha)

- Experimentales: **Chocolate, caramelo, cera, silicona, policarbonato, nylon...**

Yo prefiero utilizar PLA debido a que no produce vapores tóxicos y su uso requiere menos energía,

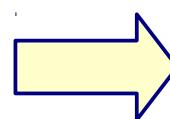
manteniendo unos resultados prácticamente igual de sólidos que con ABS. Además, el precio de ambos materiales es muy similar.

2.2 Proyecto RepRap

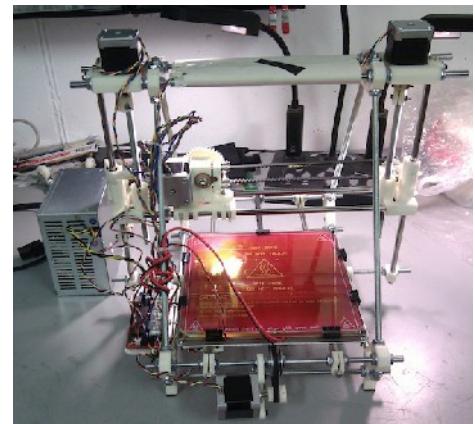
De todos los proyectos de impresoras 3D open-source, el pionero es RepRap. Su autor, Adrian Bowyer tenía en mente construir una **impresora 3D auto-replicante**. El resultado ha sido que en la actualidad existan múltiples diseños de impresoras capaces de imprimir sus propias piezas.



Piezas impresas



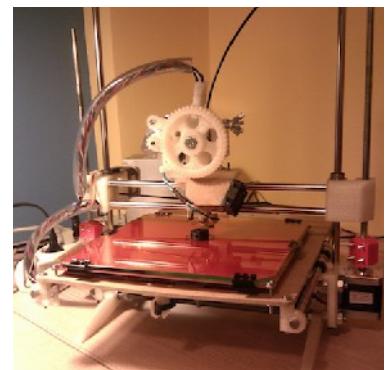
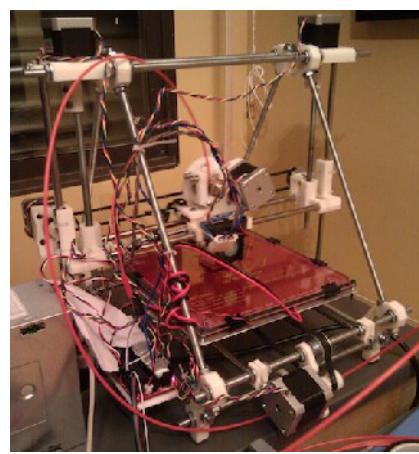
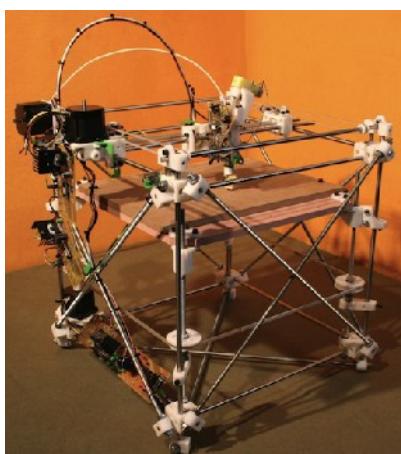
+ Piezas no
imprimibles
(motores,
electrónica...)



Impresora 3D propia

Se trata de impresoras de **hardware abierto**, en las que las piezas más específicas que las componen están fabricadas con otras impresoras 3D (o por ellas mismas), y por tanto son máquinas auto-replicantes.

Algunas de los diseños que engloba el proyecto RepRap:



El primero fue la RepRap Darwin, posteriormente surgieron otros diseños como la Prusa Mendel y finalmente la Printrbot. Estos dos últimos están representados por mis dos impresoras “Halcón Milenario” y “Lucy”.

2.3 Proyecto Clone Wars

A la hora de construir nuestra primera impresora 3D del proyecto RepRap, nos hacemos la

pregunta: **¿y de dónde saco las piezas impresas?** Para poder comenzar, se hace necesario disponer de acceso a una impresora 3D o conocer a alguien que lo tenga.



Logotipo del proyecto Clone Wars

En España, el proyecto **Clone Wars** (http://asrob.uc3m.es/index.php/Proyecto:_Clone_wars) fue fundado en la Asociación de Robótica de la UC3M en Abril de 2011 por Juan González Gómez y algunos estudiantes. Se trata de un punto de encuentro, una **comunidad de aficionados a las impresoras 3D open-source dispuestos a construirse una**.

Además, se ha creado un auténtico “**banco de piezas**” donde todo el mundo puede donar y recibir las piezas para construir sus propios equipos. Nos volvemos a preguntar **¿y de dónde saco las piezas impresas?** La respuesta: del banco de piezas de Clone Wars.

Inicialmente, se compró una impresora 3D para la UC3M, y se dio acceso a ella a los estudiantes, que comenzaron a fabricar sus propias impresoras, que se incluyeron sucesivamente en el banco de donación de piezas.

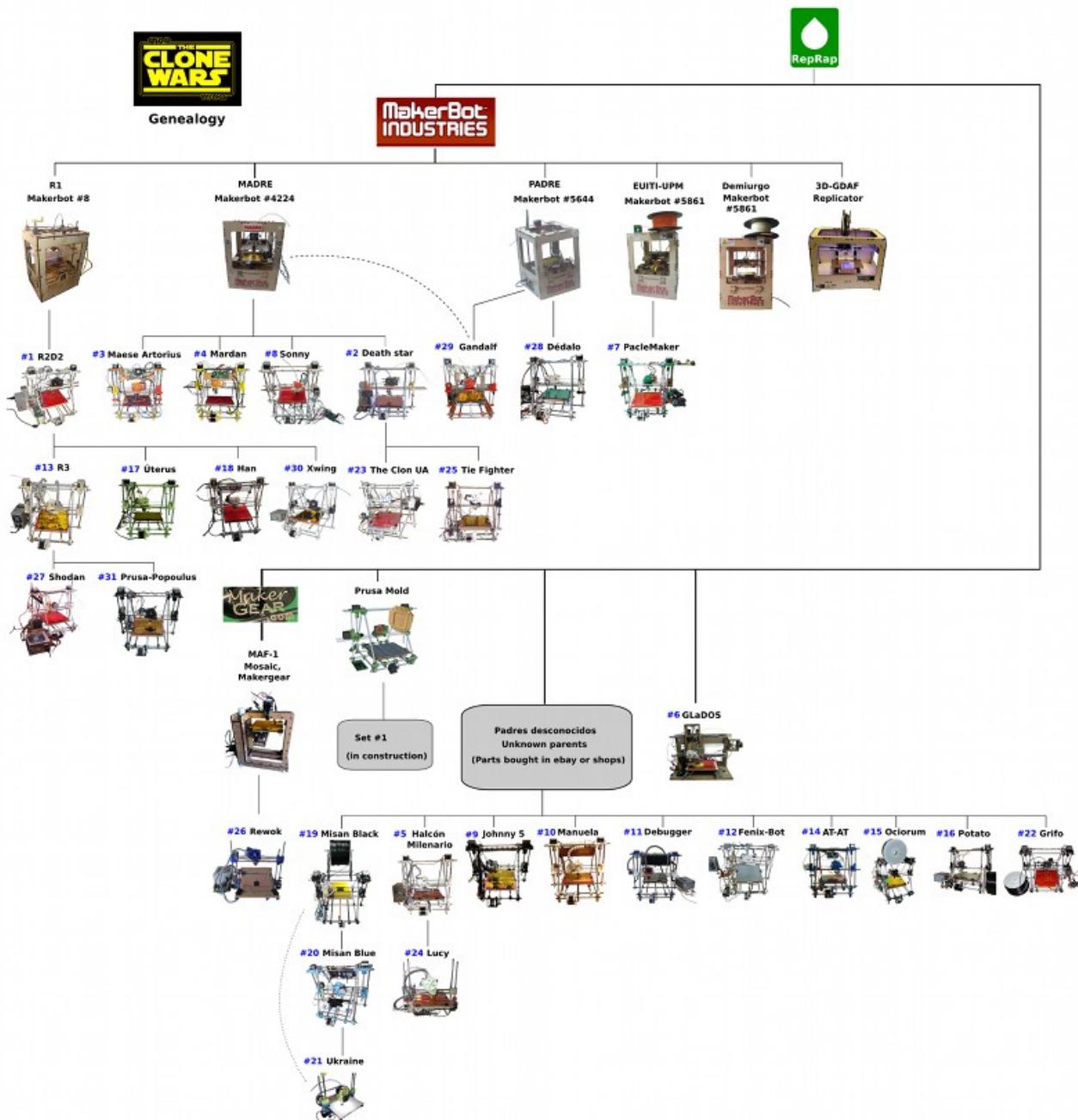


Primera reunión de Clone Wars (15/4/2011)

En el centro a la derecha está Juan González Gómez y a su lado el autor Carlos García Saura

Desde la primera reunión hasta el día de hoy, el proyecto **Clone Wars ha supuesto un increíble**

aumento de la cantidad de impresoras 3D open-source en España. El siguiente **árbol genealógico** muestra todas las impresoras del proyecto:

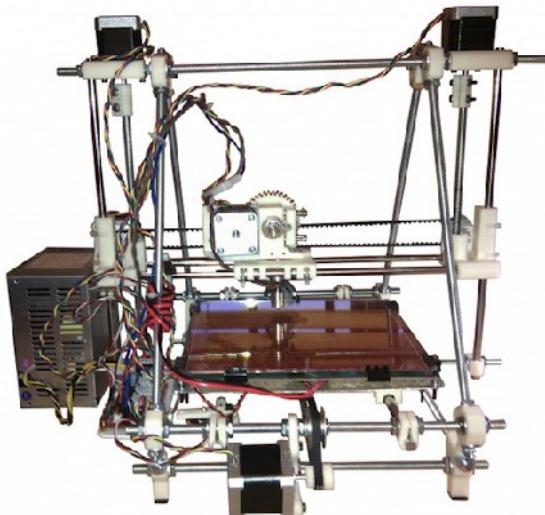


Impresoras del proyecto Clone Wars a fecha 5/7/2012

Desde que me apunté a Clone Wars desde sus orígenes, **he construido dos impresoras**, que he usado para realizar este trabajo de investigación:

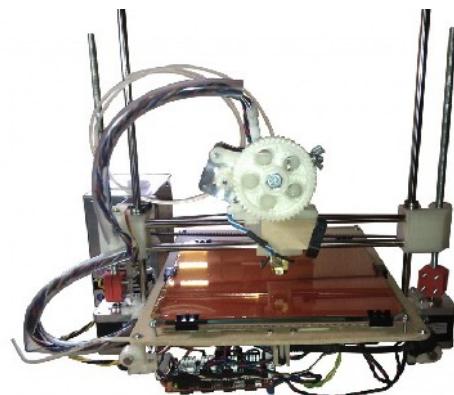
- **Halcón Milenario:** Familia “**Prusa Mendel**” (http://reprap.org/wiki/Prusa_Mendel), las piezas impresas las adquirí por internet. Ha sido construida en el **Club de Robótica y Mecatrónica (CRM) de la EPS-UAM** y entró en funcionamiento en Enero de 2012. Es la

primera impresora de estas características en la Universidad Autónoma de Madrid.



Impresora 3D “Halcón Milenario”

- **Lucy:** Familia “**Printrbot**” (<http://reprap.org/wiki/Printrbot>), **sus piezas fueron impresas por “Halcón Milenario”**. Se trata de un modelo de impresora simplificado (tamaño más reducido y mayor volumen de impresión), y la construí para tenerla en mi casa. Comenzó a funcionar en Junio de 2012. He realizado un tutorial basado en esta impresora (<http://carlosgs.es/node/Building-a-Printrbot-3D-Printer-Part-1-Overview>).

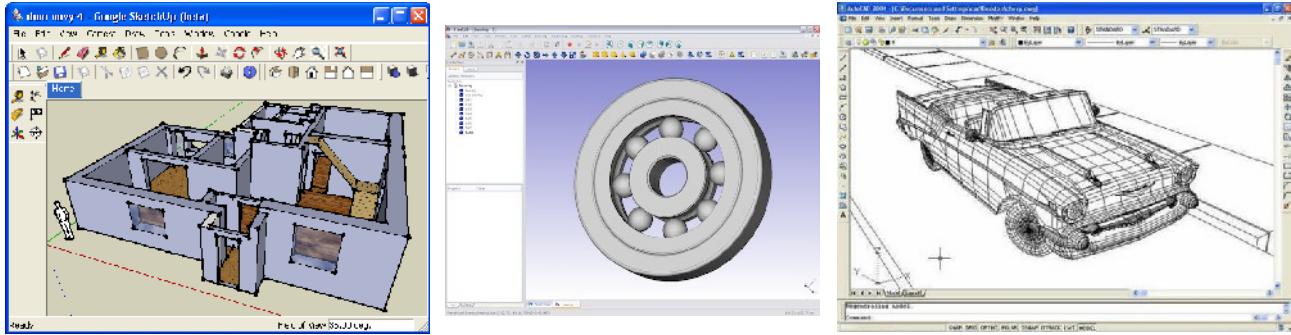


Impresora 3D “Lucy”

En resumen: **las impresoras 3D open-source (y concretamente las del proyecto RepRap) han conseguido reducir los costes en gran medida, permitiendo acercar esta tecnología a muchísimos aficionados y centros de investigación.**

2.4 OpenScad y la revolución del diseño 3D

Existen diversos programas de diseño 3D: Google Sketchup, FreeCAD, AutoCAD... etc Todos ellos ofrecen un entorno gráfico que puede utilizarse para el **modelado de piezas imprimibles**.

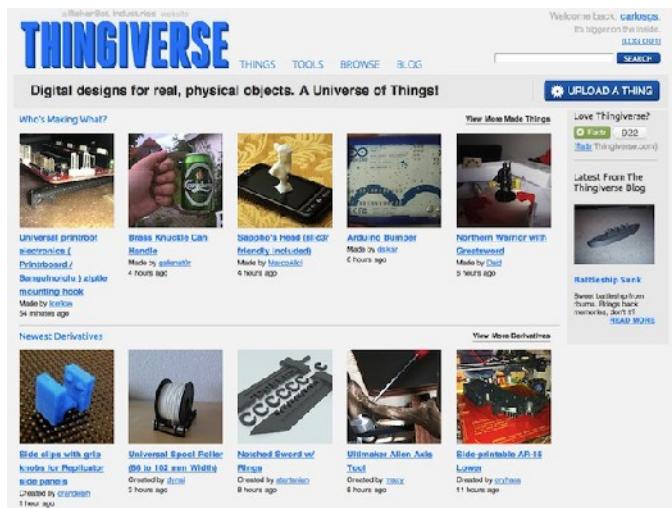


Google Sketchup, FreeCAD y AutoCAD

El formato estándar utilizado para pasar los diseños al software de la impresora es STL.

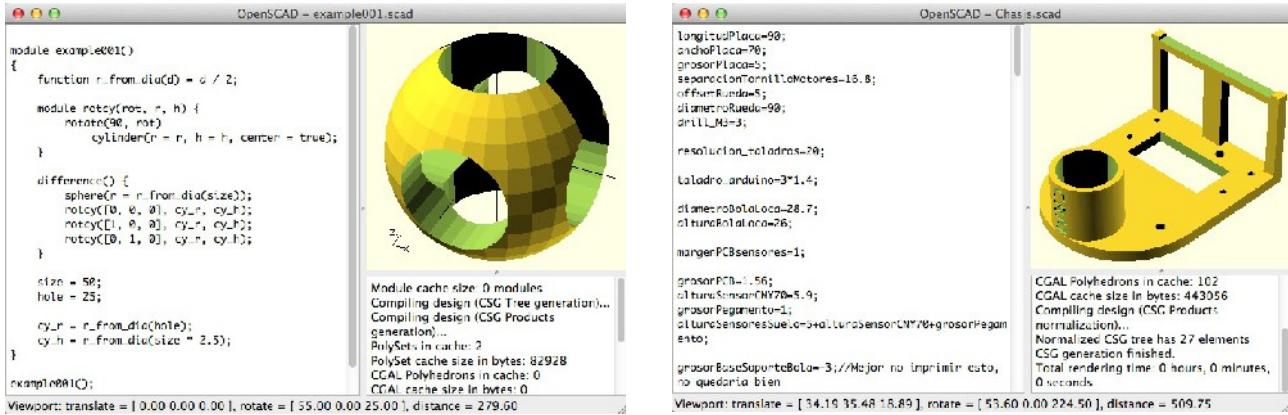
La página web **Thingiverse** (“El universo de las cosas”) es un portal donde gente de todo el mundo cuelga los diseños 3D que hace. Cualquiera puede subir fotos de las piezas que imprime, comentar las técnicas que emplea, sugerir alternativas...

En resumen, se trata de una especie de “Facebook” de las impresoras 3D.



Página web www.thingiverse.com

OpenScad es un programa libre de diseño 3D en el cual las piezas se diseñan mediante **código fuente**; el diseñador tiene a su disposición **figuras primitivas básicas** (prismas, esferas, cilindros...) y diversas funciones (**unión, diferencia...**), mediante las cuales puede componer objetos físicos mas complejos.



En la página <http://www.carlosgs.es/robots> se ha enlazado un

vídeo-tutorial sobre OpenScad realizado por Juan González Gómez.

La principal ventaja es que los diseños son **fácilmente parametrizables**, es decir, se pueden definir los parámetros básicos que describen el objeto (ancho, largo, altura, inclinación...), y cambiar su valor en cualquier momento. Al estar programado con código, es posible hacer que la pieza cambie su forma acorde a estos parámetros de forma automática.

OpenScad ha supuesto un gran avance, pero es un lenguaje limitado. Por ello es importante destacar la iniciativa de la “**Mecánica orientada a objetos**” (<http://iearobotics.com/oomlwiki/>) que busca utilizar la programación orientada a objetos como C++ en lugar de C puro conjuntamente con OpenScad. Se puede encontrar mas información sobre este concepto en estos dos artículos:

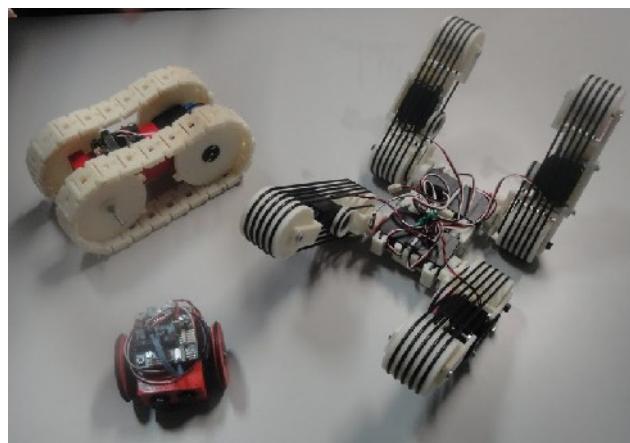
- Alberto Valero-Gomez, Mario Almagro, Nieves Cubo-Mateo, Juan Gonzalez-Gómez, “**A new Paradigm in Open Robotics with the C++ Object Oriented Mechanics Library**”, IEEE RoboCity 2012 Manuscript.
(http://www.iearobotics.com/downloads/papers/ooml_robocity12.pdf)
- Alberto Valero-Gomez, Juan Gonzalez-Gómez, Mario Almagro, “**Boosting Mechanical Design with the C++ Object Oriented Mechanics Library**”. Proc. of the 2012 IEEE int. conf. on Global Engineering Education Conference (EDUCON), 17-20 April, Morocco.
(http://www.iearobotics.com/downloads/papers/educon2012_oom.pdf)

3. Printbots: Robots libres e imprimibles

Desde que surgieron las primeras impresoras 3D y fueron llegando a los centros de investigación, se han usando para diversos objetivos, y entre ellos está muchas veces el **fabricar piezas para robots**. Según el tipo de impresión 3D utilizada, estas piezas pueden ser simples pruebas de concepto o bien partes definitivas listas para el ensamblado.

El problema con estos robots es que sus planos son, en general, privativos, ya que se encuentran restringidos al laboratorio, centro o empresa que ha desarrollado sus piezas.

En 2011, Juan González Gómez y Alberto Valero Gómez definieron el concepto de “**Printbot**” como “**robot libre e imprimible**”. Es decir, un Printbot es un **robot cuya estructura mecánica es imprimible, y el hardware es completamente libre** (esto incluye diseños 3D, esquemático de circuitos, código... etc).



Tres printbots: Orugator, MiniSkyBot y 4-Track

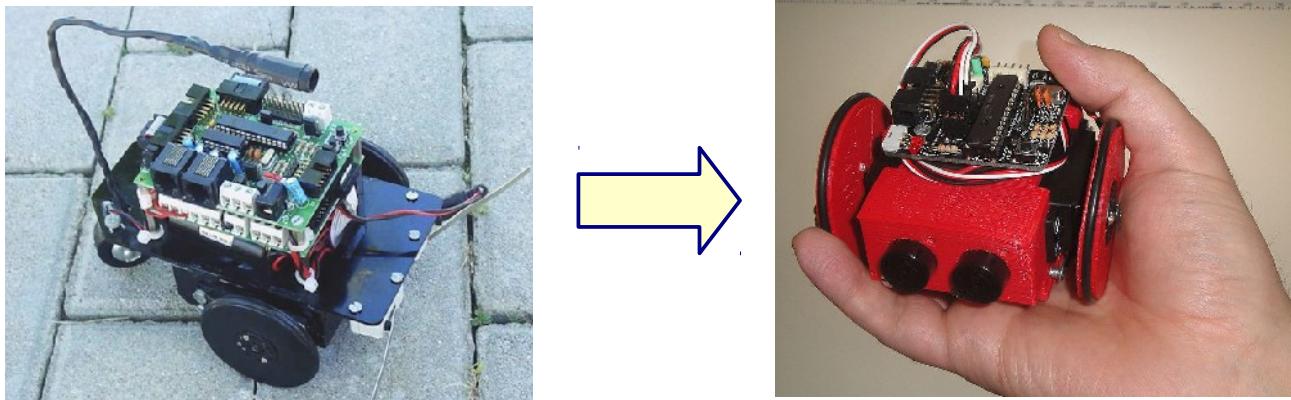
La idea es aplicar el paradigma del software libre al hardware, para conseguir formar comunidades de desarrolladores en torno a proyectos hardware. Es de lectura muy recomendada el artículo:

- Juan Gonzalez-Gomez, Alberto Valero-Gomez, Andres Prieto-Moreno, Mohamed Abderrahim (2011), "A New Open Source 3D-printable Mobile Robotic Platform for Education", *Proc. of the 6th International Symposium on Autonomous Minirobots for Research and Edutainment, May, 23-25. Bielefeld. Germany.*

(<http://www.ihearobotics.com/wiki/index.php?title=Paper:2011-Amire-Miniskybot>)

3.1 MiniSkybot

El primer Printbot fue el robot **MiniSkyBot 1.0**, terminado en Enero de 2011, que es descendiente del SkyBot original. El diseño mecánico del SkyBot fue realizado por Andrés Prieto Moreno. La electrónica del SkyBot, al igual que el MiniSkyBot completo, por Juan González Gómez.

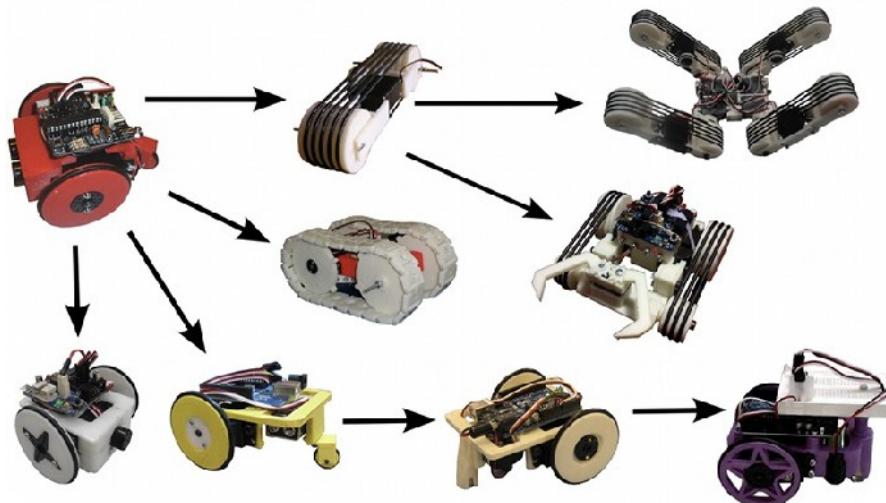


Robot SkyBot original (con la tarjeta SkyPic) y su evolución: el MiniSkyBot

El MiniSkybot tiene un chasis completamente imprimible, utiliza la electrónica de control SkyMega (que es compatible con el IDE Arduino), servos trucados para la rotación continua como motores, y dispone de sensores de distancia por ultrasonidos. Toda la documentación está disponible en su página web (<http://www.ihearobotics.com/wiki/index.php?title=Mini-Skybot>).

3.2 Evoluciones

Poco después, han ido surgiendo nuevos diseños; ¡los robots evolucionan gracias a la comunidad!



Algunos ejemplos de la descendencia y evoluciones del robot MiniSkyBot

Otro concepto importante es el de “hardware libre²”, esto es: **proyectos cuyos planos son libres, y además están realizados con herramientas de diseño libres**.

Al igual que ha sucedido con GNU/Linux frente al software privativo, el **hardware libre**² está suponiendo una gran revolución en el mundo de la robótica.

3.3 Usando servos como motores de rotación continua

Los servos son unos motores rotativos que mantienen el ángulo de giro que se les especifique utilizando ancho de pulsos (véase 1.4 Actuadores).



Servomotor y micro-servos

A pesar de que existen servomotores de rotación continua, la gran mayoría están limitados a 180°.

Internamente, consisten en un motor, la electrónica, un potenciómetro y una caja reductora.

Por ello, existen diversas técnicas de “trucado” para utilizarlos como motores “normales”:

- **Eliminar la electrónica interna del servo y la restricción de giro mecánica, manteniendo solo el conjunto motor-caja reductora.** Al retirar el driver de potencia que contiene originalmente, es necesario que utilicemos uno externo al servo. *Es el método utilizado en el robot SkyBot original.*
- **Conservar la electrónica, anulando el potenciómetro, y eliminando la restricción de giro mecánica.** De este modo, se reutiliza el driver de potencia del servo, lo que simplifica nuestro diseño electrónico. Además, proporciona un sistema muy sencillo para controlar la velocidad de giro del motor, ya que se utiliza **ancho de pulsos**. *Es el método utilizado en el robot MiniSkyBot y otros printbots.*

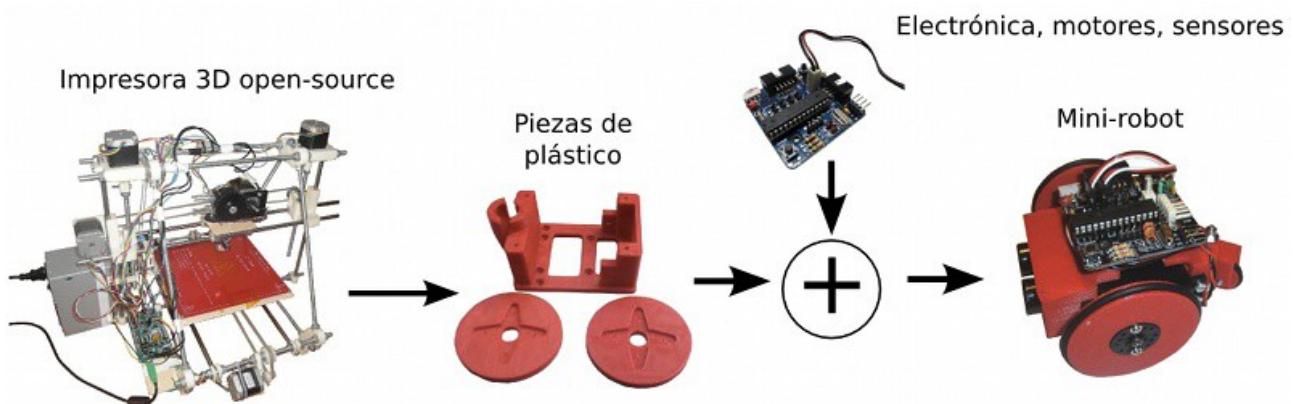
Nosotros utilizaremos el segundo método en nuestro diseño, debido al bajo coste de los servomotores y la simplicidad que conlleva no necesitar un driver de potencia.

En la página <http://www.carlosgs.es/robots> se han enlazado un par de tutoriales gráficos sobre el truaje de los servos.

La librería Robot v1.0, realizada por Santiago López Pina, simplifica el control de estos servos con Arduino (<https://github.com/darkomen/Arduino/tree/master/Robot>).

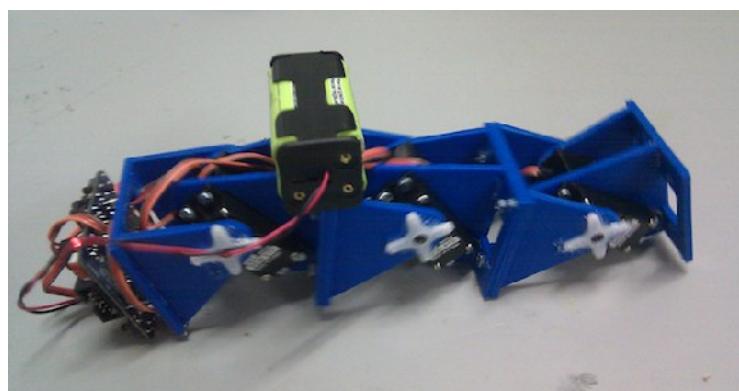
Como conclusión decir que **está claro que la tecnología de impresión 3D casera, gracias al hardware abierto y junto con los robots libres e imprimibles está suponiendo una revolución:**

- “**Patrimonio tecnológico de la humanidad**”
- **Tele-copia** en cualquier parte del mundo.
- Los diseños **evolucionan** gracias a la comunidad.
- **Coste muy reducido** en cuanto a materiales.
- Desarrolla la **creatividad**.



Esquema de construcción de un printbot

En cuanto a aspectos didácticos, **fabricar un Printbot permite desarrollar competencias tan importantes como el diseño mecánico y electrónico, su fabricación (soldadura, impresión 3D, ensamblado...), programación, lectura y ajuste de sensores... etc.** En definitiva, se trata de un proyecto muy interesante para cualquier estudiante universitario o aficionado.



Mi propia copia del printbot ArduSnake diseñado por Juan González Gómez

Los módulos son imprimibles y la electrónica es libre, por lo que cualquiera puede construir uno

Documentación del ArduSnake: <http://www.thingiverse.com/thing:21401>

Parte II: El robot HKTR-9000

En Mayo de 2012, Lucas Polo López, Miguel Gargallo Vázquez y Carlos García Saura organizamos desde el Club de Robótica y Mecatrónica un taller de iniciación a la robótica, para el cual necesitamos diseñar un printbot usando los componentes de los que ya disponíamos: el HKTR-9000

1. Resumen del taller

Los días 23, 24, 25, 28 y 29 de Mayo de 2012, tuvo lugar el "**Taller de Iniciación a la Robótica 2012**", organizado por el Club de Robótica y Mecatrónica de la EPS-UAM.

Objetivo y características del grupo

Fue dirigido a un grupo de unos **20 estudiantes de 18 a 23 años** matriculados en diversos grados y titulaciones la Escuela Politécnica Superior de la UAM (Informática, Telecomunicaciones...), y el objetivo era una iniciación a la **robótica básica**; construyendo y programando **su propio printbot**.

Duración del taller y dinámica de trabajo

El taller tuvo una **duración de 15 horas**, repartidas en 5 sesiones de 3 horas cada una. La dinámica estas sesiones consistió en una explicación inicial indicando los recursos didácticos a utilizar y posteriormente se realizaron varios **grupos de 4-5 estudiantes** que trabajaron conjuntamente para resolver los problemas propuestos. En cuanto a los **recursos** materiales empleados, fue necesario un laboratorio con ordenadores, un proyector para las presentaciones, y los kits que fabricamos previamente para construir los robots HKTR-9000 (descritos mas adelante).

Contenidos del taller

- En las dos primeras sesiones, se explicaron los **componentes básicos de un robot** y se introdujo la plataforma **Arduino**. Los alumnos aprendieron el funcionamiento de la **electrónica de sensores y motores** usando las placas de prototipos. Después programaron funciones básicas para su lectura y control.
- Las dos siguientes sesiones incluyeron la construcción del **robot HKTR-9000** (desarrollado expresamente por nosotros para el taller) y su programación. Los estudiantes aprendieron los **algoritmos de control PD** para finalmente programarlos y probarlos, haciendo que el robot siguiese la luz o una línea negra.
- La última sesión fue de trabajo libre, y se les proporcionaron **sensores de distancia y altavoces**. Con esto realizaron, por ejemplo, robots capaces de esquivar objetos y avisar de su posición mediante sonido, entre otras muchas funciones que se les ocurrieron.

Algunas fotos:



Las sesiones consistieron en una explicación previa y posterior trabajo en grupo de los estudiantes



En la última foto se aprecia el robot HKTR-9000 siguiendo una línea negra

El taller fue todo un éxito y los alumnos salieron en general bastante satisfechos.

1.1 Opinión de los estudiantes

Debido a la diversidad en el grupo, hubo opiniones muy distintas: algunos estudiantes tuvieron más dificultades (con la programación, entendiendo conceptos como el ancho de pulsos, control PID...), mientras que otros agotaban muy rápido los ejercicios propuestos y pedían más cosas que hacer.

Cuando nos dimos cuenta de este problema, decidimos re-organizar el resto de sesiones incluyendo al principio las explicaciones **reduciendo su carga teórica**, y después atendiendo las dudas de cada grupo de trabajo por separado. También incluimos **ejercicios adicionales mas avanzados** para aquellos estudiantes que los solicitasesen.

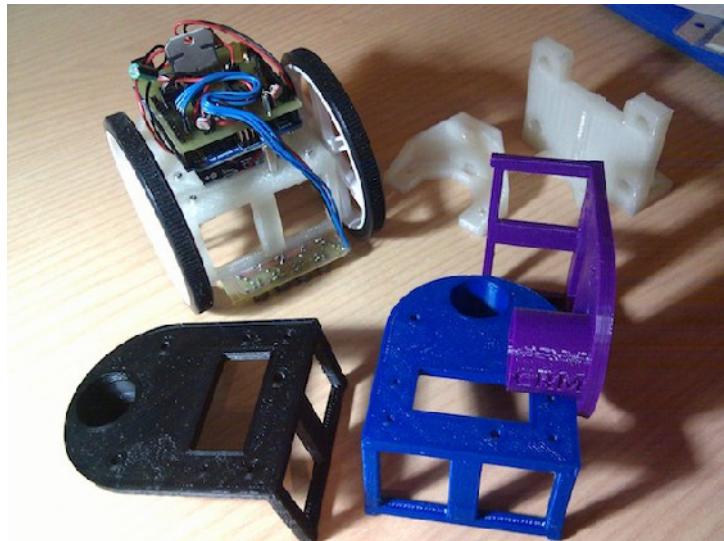
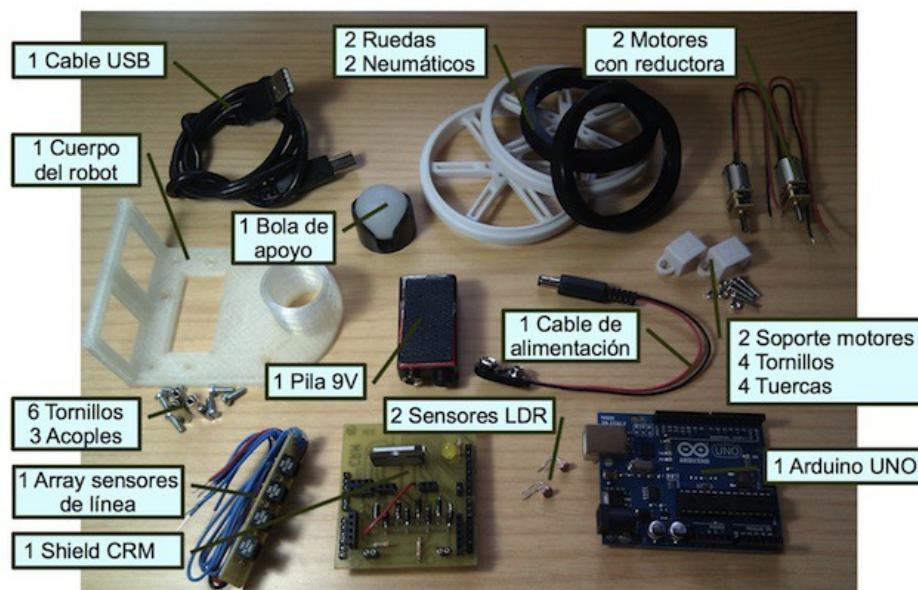
Otra solicitud común fue la de que realizásemos mas talleres tanto de diseño hardware e impresión 3D, como de diseño electrónico y fabricado de placas impresas.

Por ello, una vez finalizado, nuestra conclusión fue que **para futuros talleres sería necesario establecer “niveles” y ciertos requisitos previos**. Los talleres podrían ser o bien de robótica en general (como el que hicimos) o bien sobre aspectos concretos: electrónica y sensores con Arduino, diseño de placas impresas, diseño hardware e impresión 3D...

2. Especificaciones del robot

El HKTR-9000 fue diseñado para utilizar muchos componentes que ya teníamos en el CRM:

- Tornillos, tuercas, cables y conectores (incluyendo los de alimentación y pilas de 9V).
- Sensores LDR, CNY70 y de distancia por ultrasonidos.
- Placa Arduino UNO y cables USB.
- Algunas piezas de la marca Pololu: motores, ruedas y bola de apoyo.



Kit para construir el HKTR-9000 y el robot junto a varios chasis que fueron usados posteriormente en el taller

Al tratarse de un “printbot”, la estructura mecánica está diseñada para ser impresa en 3D, y todos los diseños son libres. Cuando lo publiqué el 23 de Mayo de 2012 en la página web Thingiverse (<http://www.thingiverse.com/thing:23593>) se convirtió en uno de los diseños más populares esa semana.

2.1 CRM-Shield

La electrónica se basa en una placa Arduino UNO con el CRM-Shield diseñado por nosotros, cuyas especificaciones son:

- Dos LDR como sensores de luz.
- Cuatro CNY70 como sensores de línea.
- Dos motores, controlados por el driver L298.
- Un diodo LED emisor de luz.
- Pines adicionales de expansión.

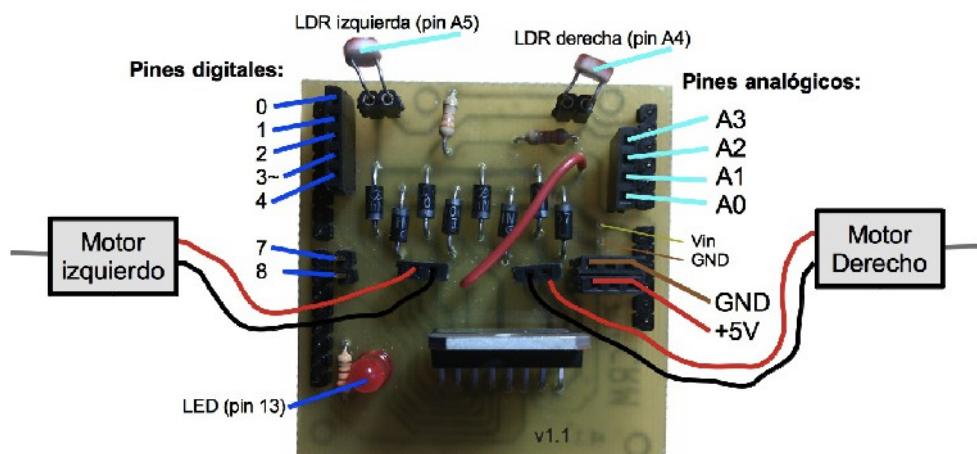


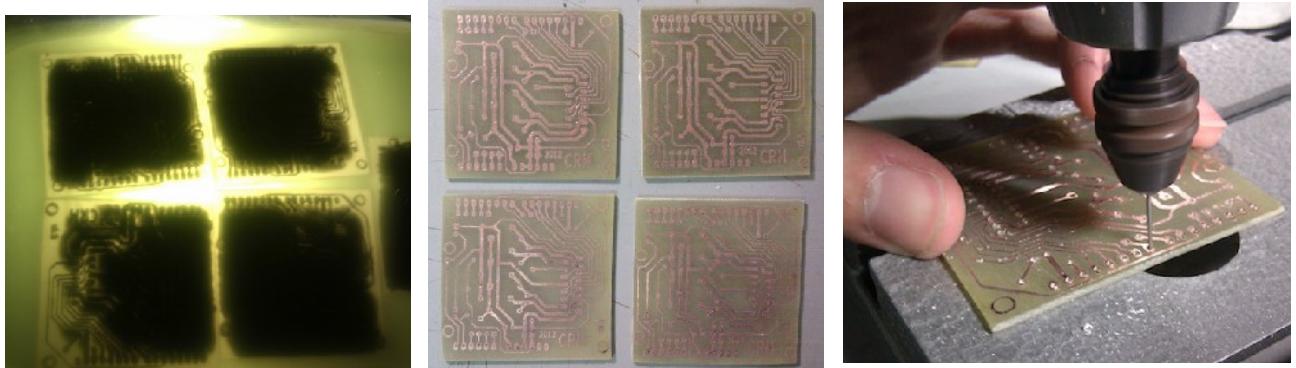
Diagrama de conexiones del CRM-Shield

La placa la diseñé con Kicad en base a las especificaciones que decidimos para el HKTR-9000. Hubo una primera versión de prueba, y después una definitiva que dio muy buenos resultados.



Diseño del PCB con Kicad y posterior fabricación para su uso en el taller

Para el taller necesitamos cinco de estas placas, que fabricamos la semana anterior al taller mediante la técnica de la **transferencia térmica de tóner** usando una impresora láser y una plancha, posteriormente eliminando el cobre con cloruro férrico.



*En la página <http://www.carlosgs.es/robots> se han enlazado
varios tutoriales para la fabricación casera de circuitos impresos.*

2.2 PCB con los sensores de línea

Diseñé una placa con cuatro sensores CNY70 capaces de detectar la posición de una línea negra bajo el robot, lo que le permite seguirla.

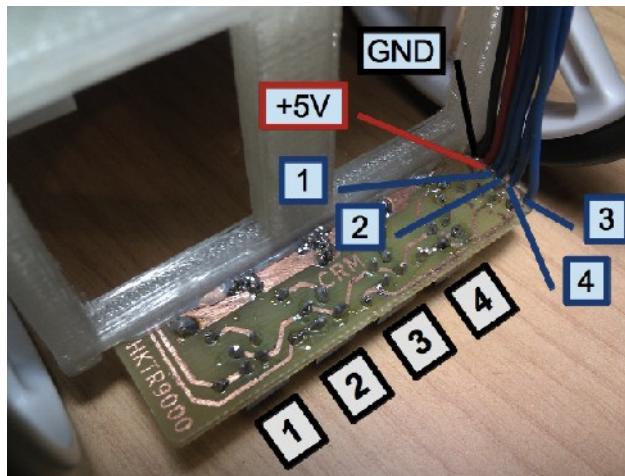


Diagrama de conexiones de la placa CNY70

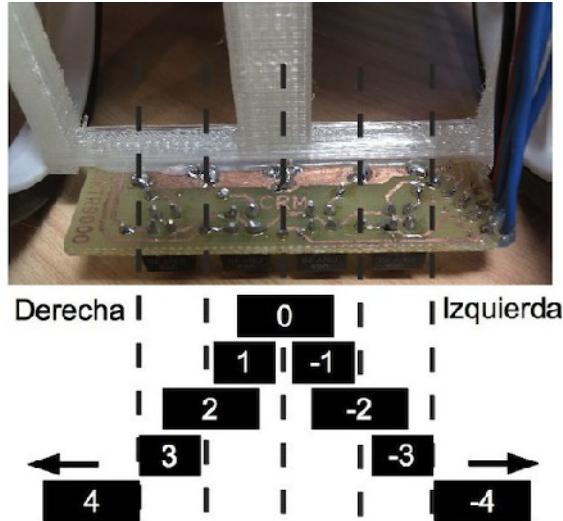
Cada sensor genera una salida analógica entre 0V y 5V, según detecte luminosidad reflejada (color blanco) o no (color negro).

Si se conectan los sensores a pines analógicos de la Arduino, se puede trabajar de dos maneras:

- **Leyendo el valor analógico directamente**, lo cual proporciona una **mayor resolución** de la posición de la línea en el suelo respecto al robot.
- **Tratando los pines como digitales**, que es lo que usamos en el taller: `pinMode(A0,INPUT)`

junto con digitalRead(A0). Esto es, **se leerá un 1 o un 0 según haya línea o no.**

La segunda opción proporciona una resolución suficiente para demostrar el funcionamiento de los algoritmos de control PD, representada en el siguiente diagrama:



En la imagen se representa el error de la posición de la línea respecto al robot

2.3 Sensor de distancia

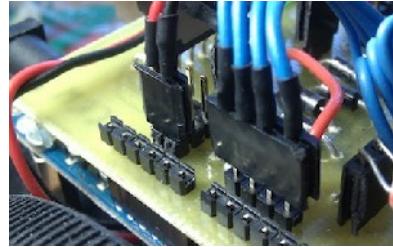
En las últimas sesiones del taller se proporcionó un sensor de distancia por ultrasonidos, concretamente el *LV-MaxSonar-EZ4*, que se conecta a los pines de expansión del CRM-Shield y va pegado en el frente del robot.

La posición del sensor fue decidida posteriormente al diseño del robot, lo que complicó las conexiones y la programación. Aun así varios grupos lograron conseguir que el HKTR-9000 esquivase obstáculos sin problema.

2.4 Conexiones eléctricas

Uno de los objetivos que nos exigimos para la realización del taller fue que no se requiriese soldar ningún elemento. Nuestra experiencia con otros talleres es que la soldadura es interesante para los estudiantes, pero quita mucho tiempo que preferimos dedicar a otros aspectos de la robótica.

Para evitar la realización de soldaduras, en la primera placa de prueba usamos conectores de pines macho en el CRM-Shield y conectores hembra para los cables.



Conexión mediante pines macho en placa y zócalo hembra en cable

Vimos que era demasiado tedioso tener que ensamblar tanto conector, por lo que finalmente nos decidimos por una solución mas elegante: usar **conectores hembra en la placa electrónica y conectar directamente el cable** (de tipo unifilar). Nos referimos a este método como “**conexión rápida mediante cable unifilar**”.



Conexión directa mediante zócalo hembra en placa y cable unifilar

Este método tan simple ha funcionado además para acercar mas a los estudiantes a la parte electrónica: es decir, en la mayoría de kits comerciales los conectores ya están ensamblados, pero **con este método el estudiante debe consultar los esquemas con mayor detalle y decidir dónde debe ir conectado cada cable individualmente, lo que le hace entender mejor la electrónica.**

Toda la documentación y diseños del robot HKTR-9000 junto con las placas impresas, se encuentran accesibles en la página web <http://www.carlosgs.es/robots>

La idea es ofrecer recursos educativos tanto para su uso por docentes en talleres como para cualquier persona que quiera aprender a construir su propio robot.

Además, hemos realizado un tutorial didáctico sobre el control proporcional derivativo aplicado a los robots seguidores de línea, con ejercicios propuestos para su implementación paso por paso.

Parte III: Mejoras propuestas

El taller realizado con el HKTR-9000 ha puesto a prueba los printbots como elemento educativo, con un resultado muy satisfactorio y conclusiones claras:

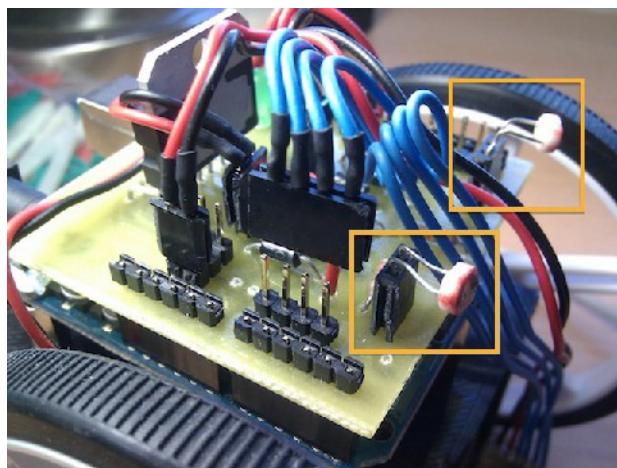
- Un printbot basado en Arduino con sensores de luz y de línea constituye una plataforma bastante completa para enseñar los conceptos básicos de la robótica y su programación.
- También es interesante el uso de otros elementos como los sensores de distancia y altavoces.

La realización del taller ha permitido identificar los puntos susceptibles de mejora (tanto técnica como para reducir el coste económico) de este tipo de plataformas.

Por ello **se va a partir de los diseños realizados para el printbot HKTR-9000** y proponer mejoras que puedan aplicarse a un Printbot que sea educativo y de bajo coste.

1. Soporte para los sensores de luz

Los sensores LDR en el HKTR-9000 están situados en el frente del robot.



Detalle de la posición de los sensores LDR en el HKTR-9000

El hecho de colocar los sensores mediante un zócalo y directamente en el CRM-Shield supuso un gran avance:

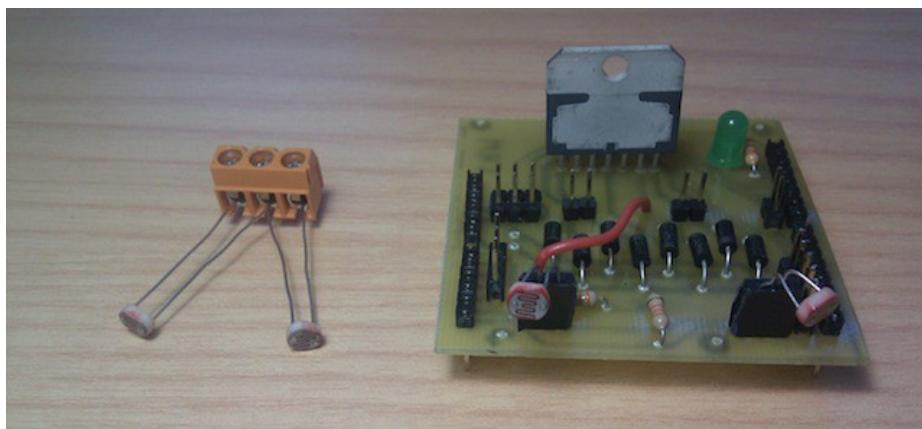
- **Gran simplicidad en las conexiones:** libre de soldaduras y conectores adicionales.
- Proporciona a los sensores una **posición privilegiada para la recepción de luz**.
- **Diseño mas compacto y limpio**, mayor integración de componentes.
- Facilidad a la hora de sustituir los sensores.

Aun así, durante el taller se identificaron varios puntos en contra:

- La **holgura en las conexiones**, que tiene como consecuencia algunas **lecturas erróneas** que complican el software.
- Los sensores quedan demasiado expuestos, **se dañan fácilmente** y es necesario un cuidado especial en el transporte del robot, para no desconectarlos.
- Al no tener ninguna carcasa que dirija la luz recibida por ambos sensores, es mas **difícil determinar la posición de la fuente luminosa**.

Para dar solución a estos problemas, se propone:

- El empleo de **terminales tipo bornier** que al usar tornillos, realizan una **conexión mucho mas fiable** y además fijan los sensores ante posibles tirones.
- Reducir el numero de pines en el conector: originalmente se usaban cuatro, pero **tres pines son suficientes** (dos de señal y uno común).
- Realizar algún tipo de **soporte que proteja los sensores** y ayude a que la **dirección de lectura de la luz** esté mejor determinada.

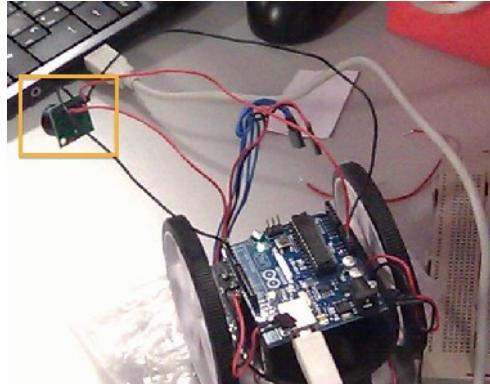


Sensores LDR en el bornier (izquierda) y en el CRM-Shield (derecha)

Con esto se pretende simplificar la instalación y uso de este tipo de sensores.

2. Integración del sensor de distancia

En los últimos días del taller, se les proporcionó a los estudiantes un sensor de distancia por ultrasonidos (modelo LV-MaxSonar-EZ4). Se utilizó **conexión rápida mediante cable unifilar**, y **pegamiento termofusible** para fijarlo en el frente del robot.



Sensor LV-MaxSonar-EZ4 conectado a la placa Arduino de un robot HKTR-9000

Los estudiantes mas avanzados lograron conseguir que el robot evitase obstáculos de forma bastante satisfactoria, pero en general hubo muchos problemas para hacer funcionar el sensor:

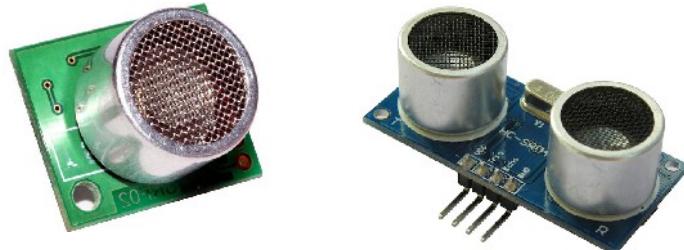
- Las conexiones mediante cable unifilar no son suficientemente sólidas y dan lugar a algunas **medidas erróneas**, que deben ser filtradas por el software.
- Es necesario aplicar **pegamento** al sensor.
- Muchas veces fue necesario volver a colocar el sensor de forma que apuntase mas hacia arriba y no interfiriesen las lecturas con su **proximidad al suelo**.

Estos hechos son la consecuencia de no haber planteado la integración del sensor cuando se empezó a diseñar el HKTR-9000.

En cuanto a aspectos económicos, el sensor utilizado resultó tener un **precio muy elevado**. No nos dimos cuenta de este hecho debido a que el Club de Robótica ya disponía de ellos desde antes de la realización del taller.

Para el diseño final, se sugiere:

- La **integración del sensor de distancia en el propio shield** con la electrónica, lo que además de simplificar las conexiones coloca el sensor en una **posición mas elevada**.
- El uso de un modelo de **sensor de coste mas reducido** (como el HC-SR04).



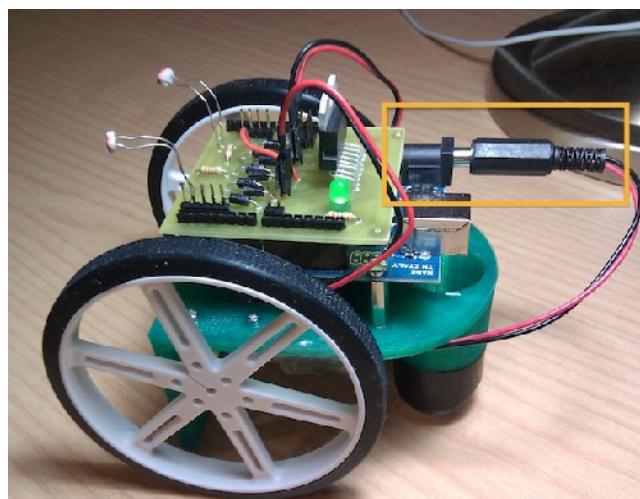
Sensores LV-MaxSonar-EZ4 y HC-SR04

Esto debería permitir que todos los estudiantes (y no solo los mas avanzados) puedan aprender fácilmente las interesantes funciones que tiene un sensor de distancia aplicado a su propio robot.

3. Uso de pulsadores en el shield

El hecho de que el CRM-Shield no disponga de algún tipo de interruptor supone tener que estar siempre pendiente de sujetar el robot al programarlo para que se ponga a avanzar encima de la mesa. Para solucionar esto, lo que hicimos en el taller fue incorporar un retardo de cinco segundos previo a la ejecución de los programas. De todas formas, este método es muy tedioso e ineffectivo (ya que un par de robots sufrieron las consecuencias de una caída por este motivo).

Cuando diseñamos el CRM-Shield, no pensamos demasiado en este problema, pero **resultó ser realmente incómodo no disponer siquiera de un pulsador de reset y depender completamente del cable de alimentación.**



Detalle del conector de alimentación en el HKTR-9000 v0.1

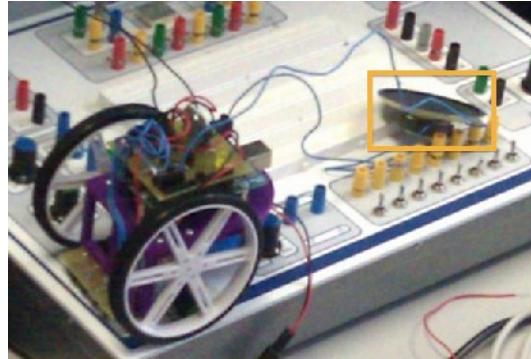
No es posible incorporar en el shield un interruptor de alimentación, ya que la recibe a través de la placa Arduino, que permanece encendida mientras están conectadas las baterías o el cable USB.

Una buena solución es el uso de pulsadores, controlados fácilmente por el software.

Por tanto, **considero imprescindible que se incluyan pulsadores en el nuevo diseño**, que permitan controlar el inicio de los programas.

4. Integración de un altavoz

En la última sesión del taller, se utilizaron altavoces para la generación de tonos con las funciones propias de Arduino. Resultó ser una funcionalidad muy interesante para los estudiantes, que lo usaron para diversos objetivos: sonar, theremín lumínico... pero no hubo forma fácil de fijar el altavoz al robot.



Altavoz de 8Ω conectado a un HKTR-9000 usando cable unifilar

Se sugiere incorporar un **zumbador piezoeléctrico**, capaz de generar tonos, que son una **buenas forma de notificación**. Además, es realmente sencillo fijar este tipo de altavoces al chasis de los printbots.



Zumbador piezoeléctrico y posible situación en el chasis del HKTR-9000

5. Uso de servomotores de rotación continua

El robot HKTR-9000 utiliza motores de la marca Pololu, controlados por el driver L298. Estos motores aumentan considerablemente el precio del robot. Además, la necesidad de usar un driver de potencia para controlarlos implica una electrónica más sofisticada y aparatoso, lo que quita espacio a otros componentes.



Ejemplo de sustitución de motores de corriente continua por micro-servos trucados en un HKTR-9000

Para ajustarse a los objetivos del trabajo (manteniendo la simplicidad y el bajo coste) se van a emplear **servomotores trucados para la rotación continua**, que únicamente necesitan alimentación y un pin para controlar dirección y velocidad.

6. Ruedas y bola de apoyo imprimibles

Cuando diseñamos el HKTR-9000 lo hicimos partiendo de los componentes que ya teníamos en el Club de Robótica. Por ello, utiliza dos ruedas y una bola de apoyo comerciales, siendo el chasis la única pieza impresa de todo el robot. Esto añade un coste innecesario, y teniendo en cuenta que se dispone de una impresora 3D ¿por qué no utilizarla?

Un printbot educativo debe disponer de la mayor cantidad posible de piezas impresas para mantener un coste reducido. Por eso, es imprescindible el **uso de ruedas y bola de apoyo imprimibles**, como es el caso del robot MiniSkyBot.



El MiniSkyBot usa ruedas y bola de apoyo imprimibles

Como resumen, nuestra propuesta de Printbot educativo de bajo coste tendrá:

- Placa controladora Arduino UNO.
- Electrónica compactada en un shield, que incluya sensores de luz y distancia por ultrasonidos, así como pulsadores.
- Altavoz piezoeléctrico.
- Sensor de línea (cuatro sensores CNY70) con conexión rápida mediante cable unifilar.
- Chasis y ruedas impresas, incluyendo la bola de apoyo.
- Como motores, servos trucados para la rotación continua.

Parte IV: Diseño experimental

Para poner a prueba las mejoras propuestas, **se ha diseñado y fabricado un printbot educativo libre, imprimible y de bajo coste.**

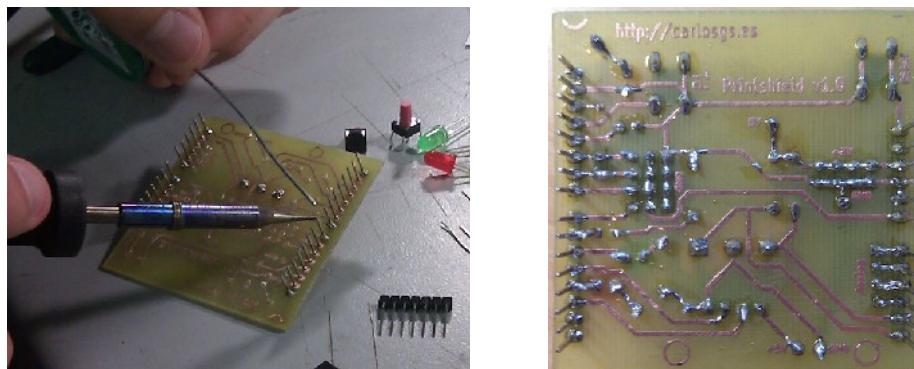
1. Printshield

La electrónica desarrollada la he denominado “**Printshield**” debido a que se trata de un **shield** específico para los printbots que funcionan con Arduino UNO.



Printshield sobre Arduino, alimentada por una pila de 9V y ejecutando un programa de prueba

Se trata de un circuito impreso diseñado con Kicad, y basado en el CRM-Shield. El diseño tiene una sola cara de cobre, y se requieren únicamente dos cables aéreos adicionales.



Soldadura de los componentes del circuito impreso

El Printshield integra prácticamente todos los elementos electrónicos del robot:

- Dos sensores frontales **LDR**.
- Un sensor frontal de **distancia** por ultrasonidos (HC-SR04).
- Cuatro **diodos LED** (tres de ellos conectados a salidas PWM, permitiendo variar su brillo).
- Conector para cuatro entradas analógicas (usadas para el **sensor de línea**).
- Conector para **altavoz** o zumbador piezoelectrónico.
- **Dos pulsadores**, uno de ellos de reset.
- Cuatro **conectores para servomotor** (u otras funciones).

Todas las mejoras que se propusieron en el tema anterior respecto a la electrónica han sido implementadas, siendo el diagrama de conexiones el siguiente:

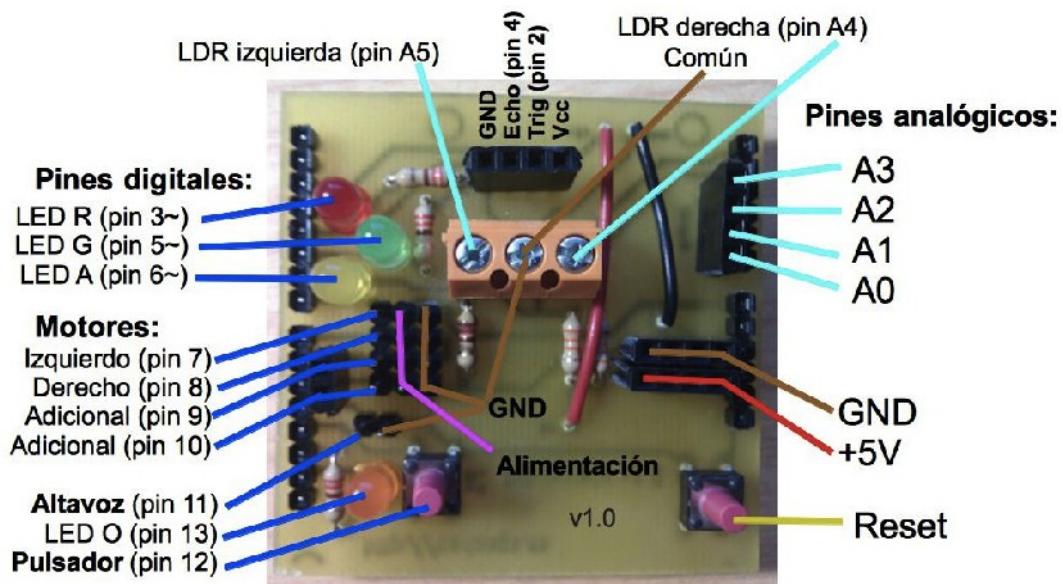
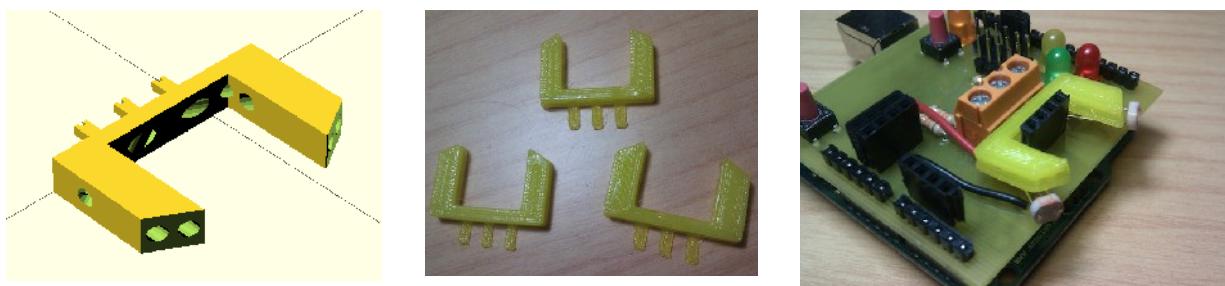


Diagrama de conexiones del Printshield

A continuación se van a proporcionar más detalles sobre esta nueva plataforma electrónica.

1.1 Sensores LDR

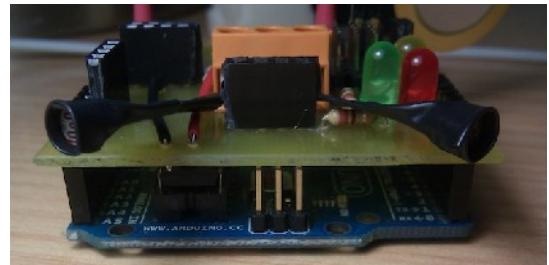
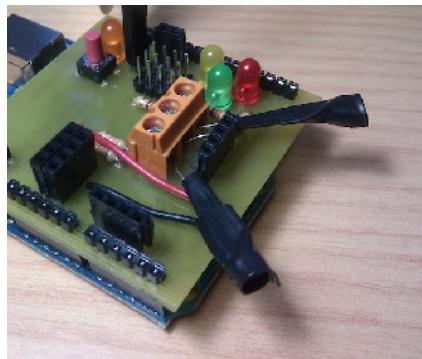
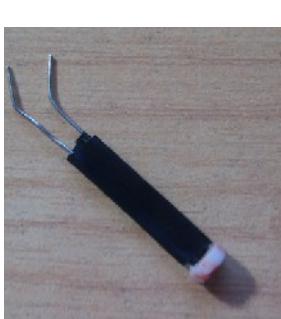
La idea inicial era realizar una **pieza impresa en 3D** en la que se insertasen los sensores de luz, y se fijasen en el Printshield usando el conector triple “bornier”.



Diseño 3D del soporta para los sensores LDR, varias pruebas impresas y prueba de concepto

El concepto no dio resultado, debido al reducido tamaño de los agujeros necesarios: la impresora 3D no dispone de tanta resolución, y quedan taponados. Aun así, no descarto la posibilidad de que un diseño mas adecuado llegue a funcionar.

A pesar de que la idea de imprimir un soporte para estos sensores no funcionó, **la conexión mediante bornier ha resultado funcionar muy bien**. Solamente ha sido necesario aislar los LDR con un poco de cinta aislante y fijarlos directamente con los tornillos.



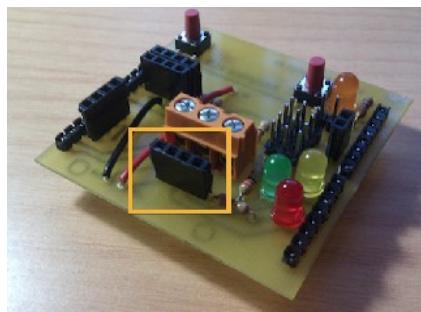
Sensor de luz con cinta aislante, vista lateral y vista frontal del montaje

La propia cinta aislante proporciona una “carcasa” que solo permite a los sensores recibir la luz dirigida al frente del robot, lo que **mejora la sensibilidad** y simplifica los algoritmos. El conector bornier cumple su función y **la conexión eléctrica es sólida y sin ruido**.

1.2 Sensor de distancia

Se buscó la manera de integrar en la misma placa varios sensores que debían apuntar al frente: los sensores de luz y el sensor de distancia. Por ello, desde el principio se tuvo en cuenta la distribución de los conectores para una **colocación óptima de los sensores**.

En el caso del sensor ultrasónico HC-SR04, se ha aprovechado que dispone de pines verticales que facilitan su montaje en la placa usando un zócalo. Este zócalo se ha situado justo en el frente del conector bornier, dejando un margen de distancia para los sensores LDR.



Conexión del sensor HC-SR04 en el Printshield

El sensor dispone de **cuatro pines**: dos de alimentación (Vcc, GND) y dos de control (Trig, Echo).

En la placa, se han conectado los pines de control a pines digitales de la Arduino.

Para la lectura de la distancia es necesario aplicar una señal al pin Trig; **la distancia leída por el sensor es proporcional a la duración del pulso recibido en el pin Echo**.

Las pruebas han sido muy satisfactorias, teniendo en cuenta que **las mediciones son mucho mas precisas** que con el sensor LV-MaxSonar-EZ4 y **el coste se ha reducido** de manera importante.

El mecanismo “**plug and play**” utilizado no puede simplificarse mas.

1.3 Servomotores de rotación continua

El Printshield dispone de **conectores para cuatro servomotores**.

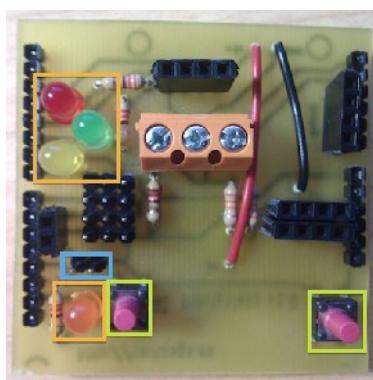


Conectores para servos (derecha) y pines adicionales (izquierda) en el Printshield

La idea es poder utilizar dos de ellos para conectar **servos de rotación continua que actúen como motores**, y dejar los otros dos de expansión para otras funciones. Por ello, dos de los conectores disponen además de zócalo hembra para admitir la conexión rápida mediante cable unifilar de **elementos adicionales como “bumpers” delanteros o sistemas de comunicación inalámbrica**.

1.4 Pulsadores, diodos LED y altavoz

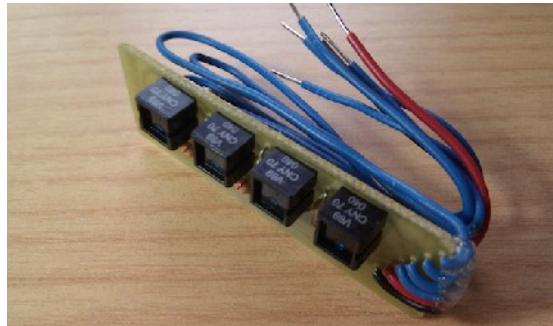
Se han incluido **dos pulsadores**: uno de ellos de **reset** y el otro conectado a un **pin digital** (para el cual es necesario activar la resistencia de pull-up interna de la Arduino). Esto facilita, por fin, la **ejecución de los programas en el momento deseado**, así como reiniciar la placa de manera cómoda sin necesidad de retirar la alimentación.



Diodos LED (naranja), conector para altavoz (azul) y pulsadores (verde) en el Printshield

Además, el Printshield incorpora un total de **cuatro diodos LED** de varios colores, que simplifican las notificaciones y el depurado del código. Tres de los diodos están conectados a salidas PWM de la Arduino, lo que permite **regular su brillo**. La posición de estos en la placa está pensada para sustituirlos por un **led RGB** en posteriores versiones del diseño.

Otra novedad es el conector para el altavoz o zumbador piezoeléctrico (aunque podría utilizarse para otras funciones fácilmente).



Placa con cuatro sensores CNY70

Respecto a la **placa de sensores CNY70** utilizada para seguir una línea, es la misma empleada en el robot HKTR-9000 ya que ha demostrado ser muy compacta, fiable y práctica.

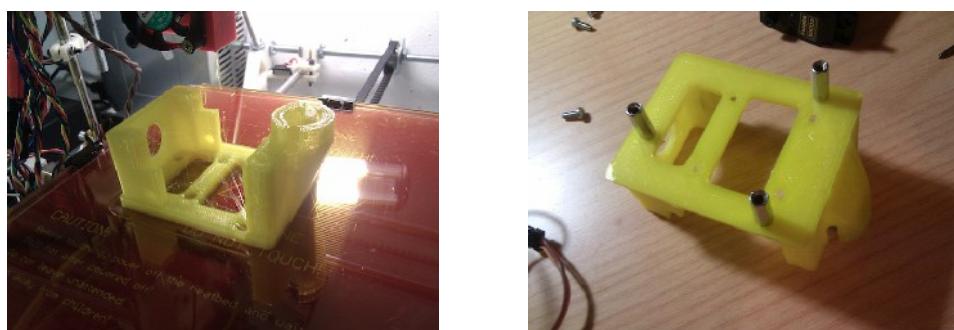
2. Diseño hardware

Se ha utilizado la **estructura de un MiniSkyBot** para demostrar las capacidades del **Printshield**.

Este diseño se ha elegido por los siguientes motivos:

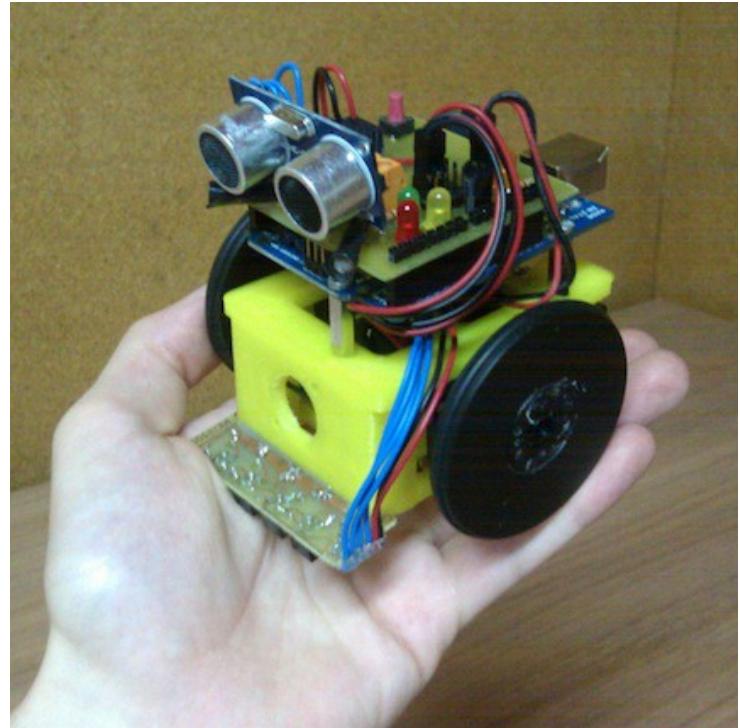
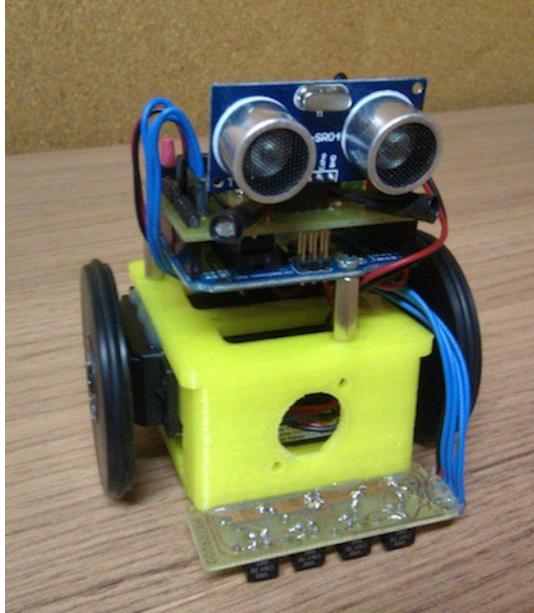
- Es un diseño mecánico muy compacto, y **completamente imprimible**; incluyendo las ruedas, la bola de apoyo e incluso el **sostén para las cuatro pilas**.
- La placa de sensores CNY70 se puede colocar en el frente de forma sencilla.
- Pensado para utilizar **servomotores trucados** para la rotación continua.

Para fabricar el chasis he usado mi impresora 3D “Halcón Milenario”, que tuvo listas las piezas en una hora y media. El material elegido fue PLA amarillo.



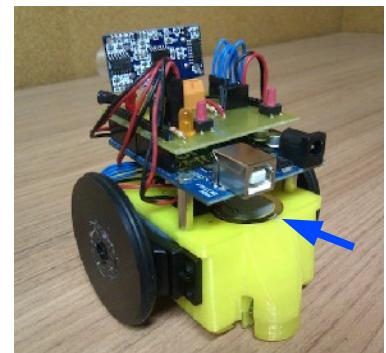
Impresión 3D del chasis del MiniSkyBot, y colocación de los tornillos de sujeción

He denominado este diseño “**ArduSkyBot**” debido a que es un MiniSkyBot basado en Arduino.



Vista frontal y tamaño del robot ArduSkyBot

En las imágenes se puede apreciar el **reducido tamaño** del diseño, así como los diversos sensores y componentes que este integra.



Vistas inferior, superior y posterior.

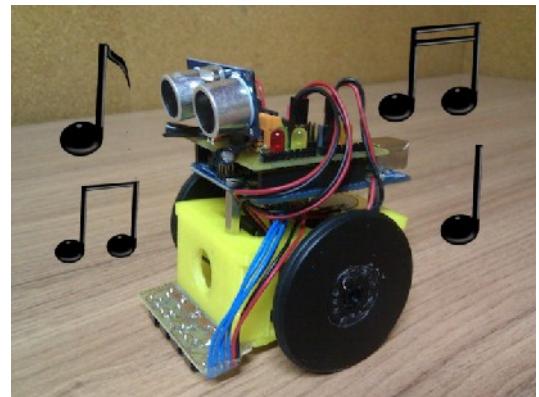
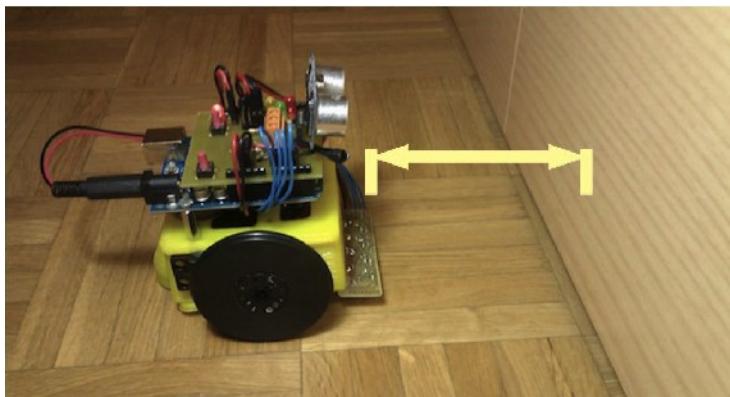
Se ha señalado la ubicación del zumbador piezoeléctrico justo debajo de la placa Arduino.

Las únicas piezas del chasis que no pueden ser impresas son:

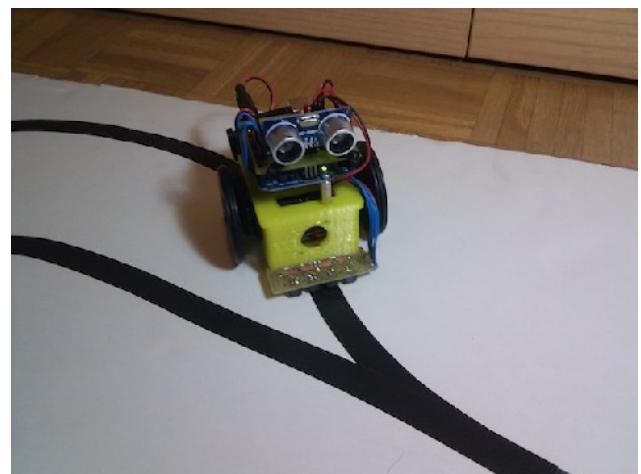
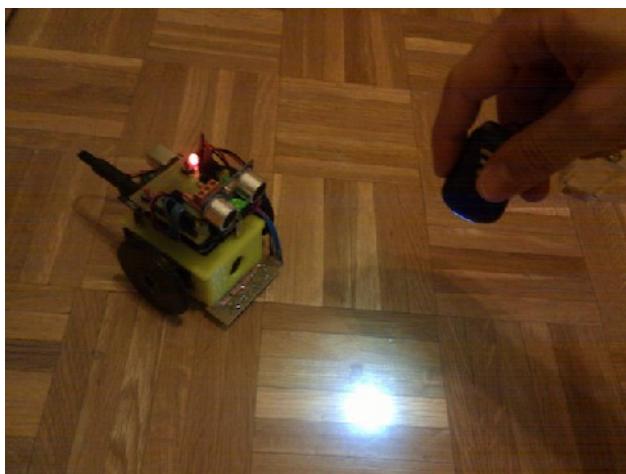
- Dos juntas de plástico a modo de neumático para las ruedas.
- Una canica de vidrio para el tercer apoyo.
- Tornillos y tuercas para fijar los servomotores y el compartimento de las cuatro pilas.

Debido a que dispone de multitud de sensores y actuadores, **el robot puede programarse para realizar diversas tareas.**

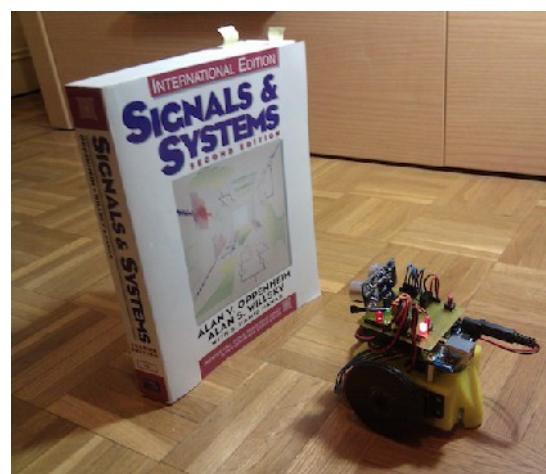
Algunas de las posibles funciones están representadas en las imágenes, son ejemplos claros de **ejercicios que se pueden proponer en los talleres**, en orden de dificultad.



Manteniendo la distancia con un objeto y reproduciendo tonos musicales



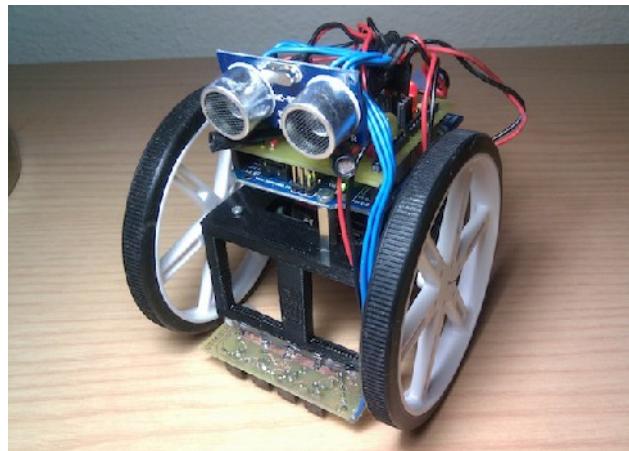
Siguiendo una fuente de luz y una línea. Se utilizan los LED integrados para notificar las acciones del robot.



Reconocimiento del terreno. El ArduSkyBot genera tonos y notifica los objetos que encuentra a su paso.

En el segundo caso se ha encontrado con un obstáculo infranqueable y no puede continuar.

Además, la placa Printshield es compatible y fácil de utilizar con otros printbots, lo que permite **dejar a cargo de los estudiantes las decisiones en cuanto a la estructura y diseño mecánico.**



Ejemplo de instalación del Printshield en un HKTR-9000 con servomotores trucados para la tracción.

3. Coste del ArduSkyBot

A continuación se detalla una **lista de los materiales necesarios para un posible kit de construcción del ArduSkyBot**, para su uso en un posible taller.

Se ha incluido el precio aproximado (Julio 2012) de cada elemento.

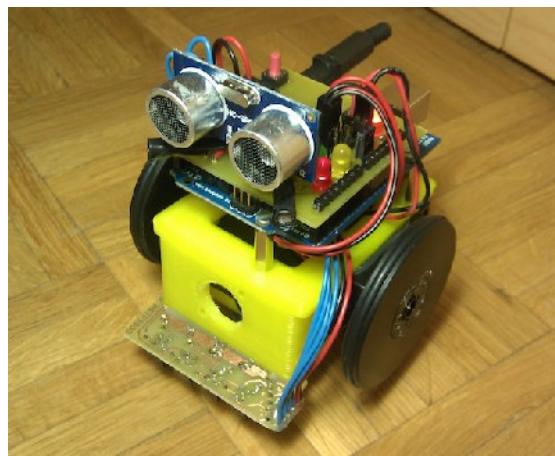
Elemento	Precio
Arduino UNO con cable USB	23€
Servos Futaba S3003	2 x 8€
Cables, conectores y componentes (LDR, pulsadores, LED, piezoelectrónico...)	5€
Sensor HC-SR04	2€
Printshield y placa de sensores CNY70 (circuitos impresos caseros)	2€
Cuatro pilas AAA	2€
Dos juntas de caucho (para las ruedas) y una canica de vidrio	1.5€
Tornillos, tuercas y soportes para placa	1€
Chasis, dos ruedas y compartimento para las pilas (impresos en 3D)	< 1€
Total:	< 53.5€

Si comparamos el coste del ArduSkyBot (53.5€) con el calculado al final del artículo "A New Open Source 3D-printable Mobile Robotic Platform for Education" para el MiniSkyBot (56.6€), vemos que es muy similar.

Por ello, se deben tener en cuenta las **prestaciones que se han incorporado al diseño:**

- Ahora esta **basado en la placa Arduino UNO**, por lo que se puede programar directamente por **USB**. Además, **muchas gente ya dispone de una de estas placas**, ya que es fácilmente reutilizable para cualquier proyecto, por lo que su coste podría descontarse.
- **Sensor de línea** formado por cuatro CNY70 y **sensor de luz** formado por dos LDR.
- El **sensor ultrasónico** de distancia está **mejor integrado** y tiene un coste menor.
- Dispone de **pulsadores, varios LED de notificación** y de un **zumbador piezoeléctrico**.

Por tanto, **se ha conseguido el objetivo de mantener un coste muy reducido sin comprometer las prestaciones del robot y potenciando sus capacidades educativas.**



Plataforma libre ArduSkyBot

4. Diseños y documentación realizados

En un esfuerzo por “Documentar y liberar los diseños, para ponerlos a disposición de todo aquel interesado en aprender a construir un robot”, se han subido a internet y documentado todos los diseños realizados:

- **Robot HKTR-9000 y CRM-Shield** <http://www.thingiverse.com/thing:23593>
- **Placa con 4 sensores CNY70** <http://www.thingiverse.com/thing:26819>
- **Printshield** <http://www.thingiverse.com/thing:26816>
- **ArduSkyBot** <http://www.thingiverse.com/thing:26818>

Adicionalmente, toda la documentación, diseños y código mas actualizados relacionados con este trabajo de investigación se encuentran disponibles en la web:

<http://www.carlosgs.es/robots>

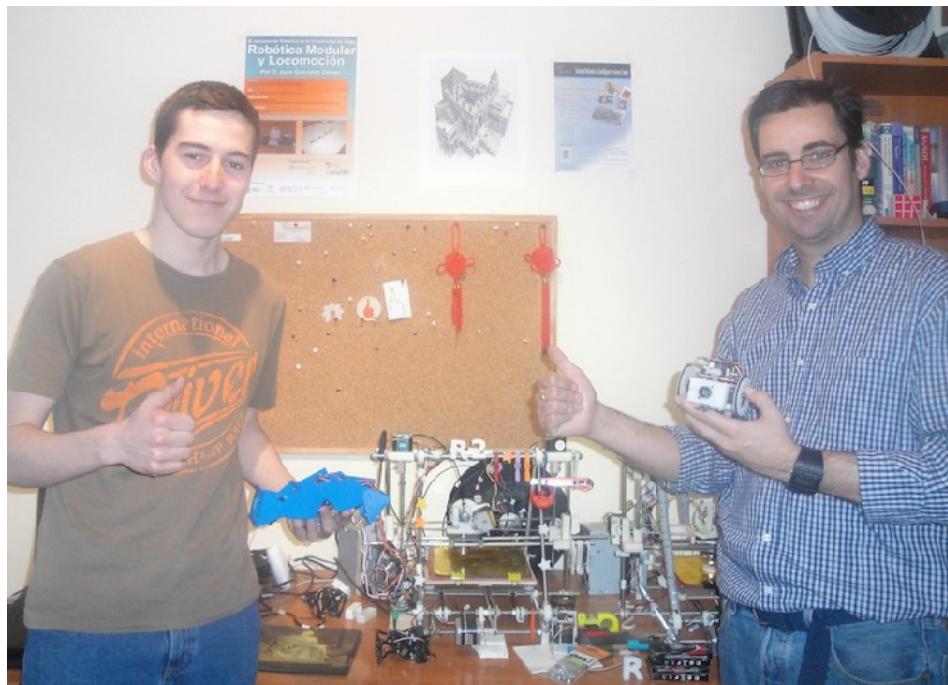
Conclusiones

- Los robots libres unidos a la tecnología de la impresión 3D casera (**printbots**) han supuesto una **revolución** debido a sus propiedades de **tele-copia y evolución gracias a la comunidad**.
- Es necesario **fomentar el uso de los printbots para la educación**.
- Se ha diseñado el **Printshield, una nueva plataforma educativa de bajo coste basada en Arduino**. Esta integra múltiples sensores y actuadores, lo que **simplifica la iniciación a la robótica**, y es **fácilmente expansible**.
- Se ha demostrado el funcionamiento del Printshield construyendo el **ArduSkyBot** y se han explorado sus posibilidades educativas con vista a futuros talleres.

Agradecimientos

Quiero mostrar mi agradecimiento a todas estas personas y entidades:

- **M^a Carmen San José:** Gracias a su experiencia y gran ayuda como tutora ha sido posible la realización de este trabajo. Este es el tercer trabajo de investigación que hago orientado por ella y le estoy muy agradecido por su incesante esfuerzo y dedicación.
- **Juan González Gómez:** Fue quien me introdujo a la robótica, y admiro toda su labor como investigador docente, promotor de la robótica imprimible y su gran trabajo documentando todos sus proyectos. Es el principal impulsor de las impresoras 3D open-source en España.
- **Club de Robótica y Mecatrónica** de la Escuela Politécnica Superior de la UAM: Como miembro del CRM en la Universidad Autónoma de Madrid, he tenido a mi disposición el material y herramientas necesarios para la construcción de los prototipos.
- **Software libre:** GNU/Linux, Kicad, OpenScad, Slic3r, Pronterface, Inkscape, LibreOffice...
- **Hardware libre:** Arduino, RepRap (impresoras 3D), diseños 3D open-source, printbots...



Carlos García Saura con un robot ArduSnake y Juan González Gómez con un MiniSkyBot

En el centro puede verse la impresora R2D2, la primera del proyecto Clone Wars, y R3 a su derecha.

Fuentes, imágenes y licencia

Bibliografía:

- Juan Gonzalez-Gomez, Alberto Valero-Gomez, Andres Prieto-Moreno, Mohamed Abderrahim (2011), "A New Open Source 3D-printable Mobile Robotic Platform for Education", *Proc. of the 6th International Symposium on Autonomous Minirobots for Research and Edutainment, May, 23-25. Bielefeld. Germany.*
(<http://www.iearobotics.com/wiki/index.php?title=Paper:2011-Amire-Miniskybot>)
- Alberto Valero-Gomez, Mario Almagro, Nieves Cubo-Mateo, Juan Gonzalez-Gómez, "A new Paradigm in Open Robotics with the C++ Object Oriented Mechanics Library", IEEE RoboCity 2012 Manuscript.
(http://www.iearobotics.com/downloads/papers/ooml_robocity12.pdf)
- Alberto Valero-Gomez, Juan Gonzalez-Gomez, Mario Almagro, "Boosting Mechanical Design with the C++ Object Oriented Mechanics Library". Proc. of the 2012 IEEE int. conf. on Global Engineering Education Conference (EDUCON), 17-20 April, Morocco.
(http://www.iearobotics.com/downloads/papers/educon2012_oom.pdf)

Recursos online:

- Proyecto Arduino (plataforma electrónica programable libre): <http://www.arduino.cc/>
- Software Kicad (editor de esquemáticos y PCB): <http://iut-tice.ujf-grenoble.fr/kicad/>
- Proyecto RepRap (impresoras 3D open-source auto-replicantes): <http://reprap.org/>
- Proyecto Clone Wars: http://asrob.uc3m.es/index.php/Proyecto:_Clone_wars
- Software OpenScad (diseño 3D mediante código): <http://www.openscad.org/>
- MiniSkyBot: <http://www.iearobotics.com/wiki/index.php?title=Mini-Skybot>
- Robot HKTR-9000: <http://www.thingiverse.com/thing:23593>

Imágenes:

- "Placa Arduino Uno", "Varias placas Arduino", "Pines disponibles...", y "Algunos ejemplos de shields" por Arduino Team (<http://www.arduino.cc/>)
- "Esquema de conexión de un shield..." por Harald Naumann (<http://www.gsm-modem.de/>)
- "Modelos comerciales de impresoras 3D" <http://www.printcountry.com/why-3d-printers-practical-choice.html> (izquierda) y <http://replicatorinc.com/blog/2009/03/why-3d-printers-won't-go-mainstream/> (derecha)
- "Diagramas que muestran el laminado de piezas..." by Materialgeeza [CC-BY-SA-3.0 (<http://creativecommons.org/licenses/by-sa/3.0>) or GFDL (<http://www.gnu.org/copyleft/fdl.html>)], via Wikimedia Commons

- “...y el método FDM” by Zureks [GFDL (<http://www.gnu.org/copyleft/fdl.html>) or CC-BY-SA-3.0-2.5-2.0-1.0 (<http://creativecommons.org/licenses/by-sa/3.0>)], via Wikimedia Commons
- “Logotipo del proyecto Clone Wars”, “Primera reunión de Clone Wars” e “Impresoras del proyecto Clone Wars” (modificada para acomodar el formato)
http://asrob.uc3m.es/index.php/Proyecto:_Clone_wars
- “Página web www.thingiverse.com” <http://www.thingiverse.com>
- “Tres printbots: Orugator, MiniSkyBot...”, “Robot SkyBot original (con la tarjeta SkyPic)...”, “Algunos ejemplo de la descendencia” y “Esquema de construcción de un printbot” por Juan González Gómez (<http://www.inearobotics.com>)
- “El MiniSkyBot usa ruedas y bola de apoyo imprimibles” por Miguel Ángel de Futos (<http://aerobotclubderoboticadeaeronuticos.blogspot.com.es/2012/03/robotica-imprimible.html>)

Las imágenes y contenido no citados que han sido realizados expresamente para este proyecto están publicados bajo la licencia:

Attribution - Share Alike Creative Commons license.



<http://creativecommons.org/licenses/by-sa/3.0/>

Autor:
Carlos García Saura