

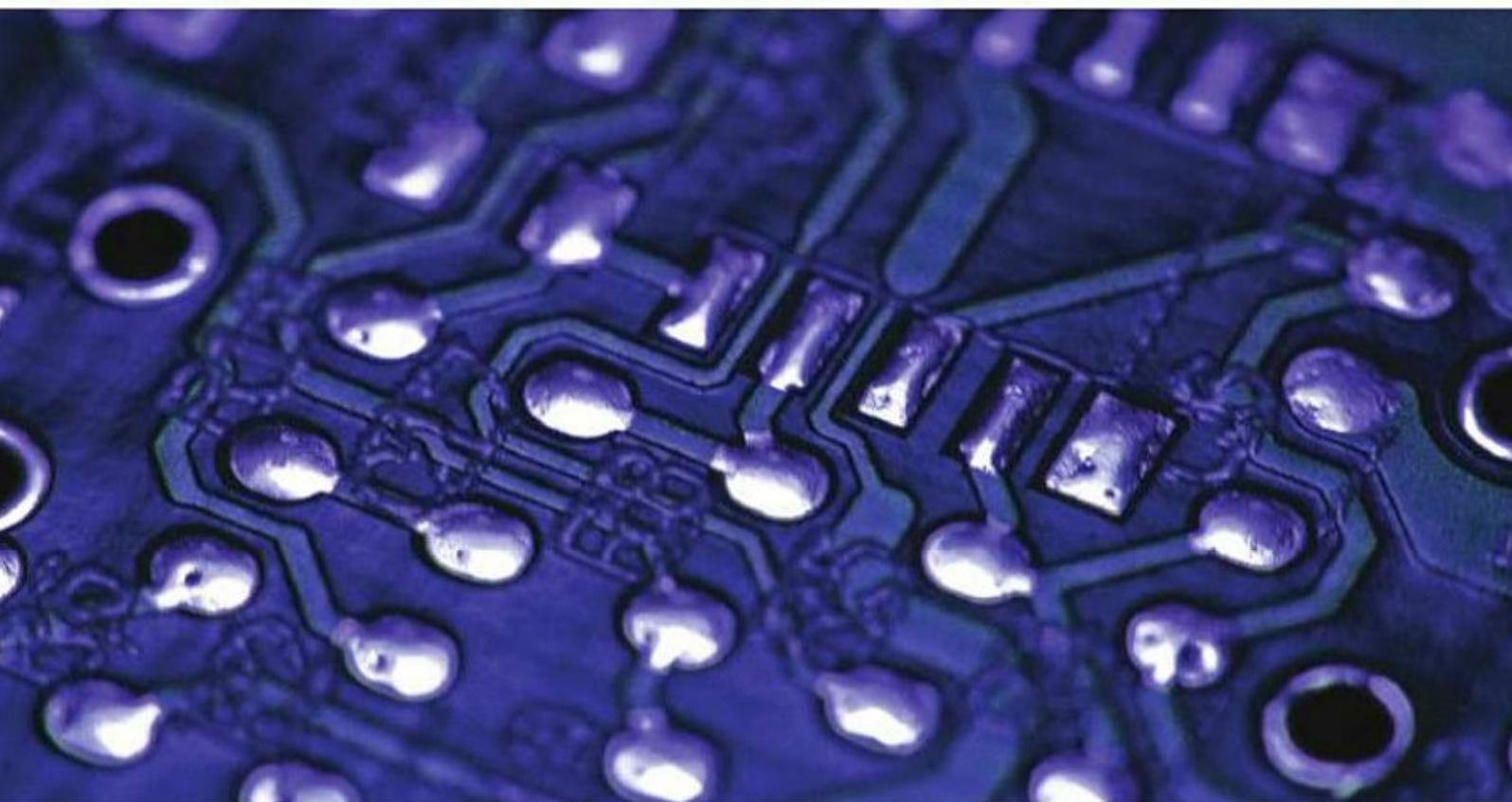
USERS

INCLUYE
VERSIÓN DIGITAL
GRATIS

ELECTRÓNICA

PLATAFORMAS ARDUINO Y RASPBERRY PI

HARDWARE ABIERTO + CONECTIVIDAD POR CABLE E INALÁMBRICA + PROYECTO DE SISTEMA DE TELEMETRÍA CON ARDUINO + INTRODUCCIÓN AL USO DE RASPBERRY PI



CONOZCA LOS SECRETOS DEL MUNDO DE LA ELECTRÓNICA



ELECTRÓNICA: PLATAFORMAS ARDUINO Y RASPBERRY PI



TÍTULO: Electrónica: plataformas Arduino y Raspberry Pi
AUTOR: Diego Aranda ... [et.al.]
COLECCIÓN: Manuales USERS
FORMATO: 17 x 24 cm
PÁGINAS: 320

Copyright © MMXIV. Es una publicación de Fox Andina en coedición con DÁLAGA S.A. Hecho el depósito que marca la ley 11723. Todos los derechos reservados. Esta publicación no puede ser reproducida ni en todo ni en parte, por ningún medio actual o futuro sin el permiso previo y por escrito de Fox Andina S.A. Su infracción está penada por las leyes 11723 y 25446. La editorial no asume responsabilidad alguna por cualquier consecuencia derivada de la fabricación, funcionamiento y/o utilización de los servicios y productos que se describen y/o analizan. Todas las marcas mencionadas en este libro son propiedad exclusiva de sus respectivos dueños. Impreso en Argentina. Libro de edición argentina. Primera impresión realizada en Sevagraf, Costa Rica 5226, Grand Bourg, Malvinas Argentinas, Pcia. de Buenos Aires en VI, MMXIV.

ISBN 978-987-1949-56-4

Electrónica: plataformas Arduino y Raspberry Pi /

Diego Aranda ... [et.al.]. - 1a ed. - Ciudad Autónoma de Buenos Aires: Fox Andina; Buenos Aires: Dalaga, 2014.

320 p.; 24x17 cm. - (Manual users; 265)

ISBN 978-987-1949-56-4

1. Informática. I. Aranda, Diego

CDD 005.3

Prólogo



Cuando en los años 80 irrumpió el movimiento de software libre, de la mano de Richard Stallman y su *Free Software Foundation*, muchos se preguntaron si realmente era una buena idea compartir el conocimiento de forma abierta, es decir, crear proyectos para que otros puedan reproducirlos, aprovechando el conocimiento aplicado por su creador, e incluso con la posibilidad de mejorarlo, modificarlo y realizar nuevas creaciones en base a él.

Durante los últimos treinta años, el movimiento de código abierto y de software libre ha mantenido una gran vigencia y, en especial, en la última década, se ha comenzado a popularizar la idea de hacer una analogía entre los circuitos y el código fuente, para que puedan quedar a disposición de cualquier persona o se puedan diseñar otras cosas a partir de lo que fue diseñado originalmente.

En este caso, la problemática está enfocada en la dificultad de copiar elementos físicos de la manera en que se copia un software. Sin embargo, esto se fue resolviendo con la producción distribuida de placas en distintos lugares, las cuales llegaron a tener, en muchos casos, hasta usos comerciales. De esta forma, lo que se puede cobrar es el servicio y el valor agregado, y no el diseño del circuito original.

El hardware libre es un concepto de gran importancia para el mercado, especialmente para acercar la tecnología a los usuarios entusiastas y a los especialistas en informática y electrónica, ya que les permite prescindir de algunas cuestiones que funcionan como limitaciones, tanto desde el punto de vista comercial como técnico. En el contexto del hardware y los circuitos libres, Arduino es una de las plataformas que mejor se posicionó, junto a Raspberry Pi.

Como prácticamente todo hardware, este se complementa por medio del software que se debe escribir para que funcione, que también está al alcance de muchas personas sin que haga falta que tengan profundos conocimientos de programación.

La red de productos sobre tecnología más importante del mundo de habla hispana



Libros

Desarrollos temáticos en profundidad

Coleccionables

Cursos intensivos con gran despliegue visual



Revistas

Las últimas tecnologías explicadas por expertos



RedUSERS redusers.com

Noticias actualizadas minuto a minuto, reviews, entrevistas y trucos



Newsletters

Regístrese en redusers.com para recibir un resumen con las últimas noticias



RedUSERS PREMIUM premium.redusers.com

Nuestros productos en versión digital, con contenido adicional y a precios increíbles



Usershop usershop.redusers.com

Revistas, libros y fascículos a un clic de distancia y con entregas a todo el mundo



El libro de un vistazo

Este libro propone, de manera general, un recorrido por la historia del movimiento de open hardware, que servirá como introducción al desarrollo de las plataformas Arduino y Raspberry Pi, las dos plataformas que mejor se posicionaron en el mercado, junto a sus entornos de desarrollo.

*01



PLATAFORMAS ABIERTAS

Un recorrido por la historia del movimiento de software libre y la relación entre los circuitos y el código fuente. Además, algunos ejemplos de las plataformas de hardware libre más utilizadas en la actualidad, para que el lector pueda continuar explorando dichas tecnologías en caso de que le sea de interés particular.

*02



PLATAFORMA ARDUINO

El surgimiento de Arduino, sus principales características y las aplicaciones en el campo de la electrónica, de la automatización y de la telemetría. El entorno de desarrollo IDE oficial de Arduino, otros IDEs alternativos y los Physical Etoys.

*03



PROYECTO: SISTEMA DE TELEMETRÍA CON ARDUINO

Una aplicación versátil y didáctica: un sistema de telemetría construido a partir de una plataforma Arduino, con todas sus aplicaciones y técnicas posibles. Incluye también, los

módulos para implementar el sistema de telemetría, medir la posición con GPS y transmitirla a través de mensajes de texto.

*04



RASPBERRY PI

Sus características de hardware y software para mejorar su rendimiento, además de su interacción con Arduino. Cómo controlar dispositivos periféricos de entrada y salida, cómo conectarla a una computadora y controlarla desde otra, teniéndolas conectadas en red.

*05



CONECTIVIDAD POR CABLE

Los protocolos de comunicación para generar conexiones efectivas, las técnicas de modulación/demodulación, y estándares como I₂C, SMBus y SPI. Además, la comunicación serie y el estándar RS-232. Los estándares más flexibles, como USB. Por último, algunos programas y utilidades de diagnóstico para el análisis de conexiones de dispositivos electrónicos por cable.

***06****CONECTIVIDAD INALÁMBRICA**

Los fundamentos de la comunicación sin cables: el espectro de frecuencias, el uso de antenas y sus principios de funcionamiento. Además, las tecnologías para interconectar dispositivos: la comunicación por infrarrojo y otras tecnologías, como ZigBee y Bluetooth.

***07****FUENTES DE ALIMENTACIÓN**

Un estudio diferenciado de las fuentes de alimentación, lineales y conmutadas. Además, los circuitos y los dispositivos reguladores de voltaje, y un proyecto de

armado de una fuente completa.

Por último, la generación alternativa de energía eléctrica con los sistemas de alimentación ininterrumpida (UPS) y los generadores eléctricos a combustible.

***08****LAS FUENTES DE ALIMENTACIÓN Y LA ROBÓTICA**

Un repaso por las nociones básicas de la robótica, las diferencias que existen entre los distintos tipos de robots, el funcionamiento de alguno de sus sistemas y las distintas fuentes de energía que se utilizan en los circuitos electrónicos.

**INFORMACIÓN COMPLEMENTARIA**

A lo largo de este manual podrá encontrar una serie de recuadros que le brindarán información complementaria: curiosidades, trucos, ideas y consejos sobre los temas tratados. Para que pueda distinguirlos en forma más sencilla, cada recuadro está identificado con diferentes iconos:



**CURIOSIDADES
E IDEAS**



ATENCIÓN



**DATOS ÚTILES
Y NOVEDADES**



SITIOS WEB

Contenido

Prólogo	4
El libro de un vistazo.....	6
Información complementaria.....	7
Introducción.....	12

* 01

Plataformas abiertas

Historia de un estilo	14
GNU	14
Más allá del software	16
El software libre	17
Modificaciones	20
Linux	21
Copyleft	22
Open hardware.....	22
Filosofía	23
Desafíos	27
Proyecto Arduino	29
Historia y evolución.....	31



Modelo abierto para hardware.....	35
Usos comerciales.....	39
Proyectos de hardware libre	44
Elphel	44
RepRap	46
OpenSPARC	48
OpenBook	53
TuxPhone	53
Oscar	55
Open Graphics Project	57
Arduino	58
Nano Satélite	58
Resumen	59
Actividades	60

* 02

Plataforma Arduino

Historia de Arduino	62
Origen	62
Funcionamiento	63
Entorno	65
Colaboradores	65
Evoluciones, versiones y periféricos	66
Librerías	68
Programación de Arduino.....	70
Lenguajes de bajo nivel	71
Lenguajes de alto nivel	72
Lenguajes de medio nivel	73
Plataforma Arduino	73
Lenguajes C y C++	74
Bibliotecas	78
Entorno de desarrollo	83
Physical Etoys	88
Periféricos y referencias	92
Referencias y recursos	96
Resumen	97
Actividades	98

***03**

Proyecto: sistema de telemetría con Arduino

Telemetría con Arduino.....100

- Telemetría.....102**
- Telemetría con Arduino103**
- Aplicaciones de los sistemas de telemetría.....106**
- Magnitudes físicas medibles con Arduino.....110**

Implementar el proyecto: formas de comunicación y transferencia.....115

- Componentes del sistema de telemetría119**
- ¿Qué vamos a usar?.....120**
- Armado y explicación del código de Arduino.....125**
- Explicación del código final.....129**

Limitaciones y otras alternativas.....136

- Alternativas con otras plataformas.....140**

Resumen145

Actividades146

***04**

Raspberry Pi

Raspberry Pi148

- Características del hardware.....149**
- Rendimiento.....153**

Software necesario.....154

- Raspbian155**

Carga del sistema operativo.....156

Overclock.....157

Integración de Raspberry y Arduino.....158

- Consideraciones sobre Python.....162**

Salida - entrada GPIO.....162

- WiringPi.....166**

Ethernet y Raspberry Pi.....168

Tecnologías VNC y SSH.....169

Resumen171

Actividades172

***05**

Conectividad por cable

Conectividad y protocolos TCP/IP174

Protocolos TCP/IP178

- Capas del modelo OSI180**

Modulación y estándares184

Bus I²C y SMBus.....189

Estándar SPI192

Comunicación asincrónica y RS-232.....197

USB.....201

- Funcionamiento.....203**
- Formato205**
- Diagrama de capas.....206**
- Puerto USB208**
- Conectores209**
- Aplicaciones.....210**

Software y utilidades.....212

Resumen217

Actividades218

***06**

Conectividad inalámbrica

Comunicación sin cables.....220

- Tecnología Wi-Fi221**
- Tecnología WiMAX222**
- Tecnología GPRS y 3G.....222**

Antenas.....223



Patrón de radiación.....	223
Ganancia de las antenas	224
Comunicaciones a corta distancia.....	225
Comunicación infrarroja o IrDA	225
Comunicación en modo semidifuso	227
Comunicación en modo difuso.....	227
Bluetooth.....	228
Características.....	229
Alcance y velocidad de los dispositivos Bluetooth.....	230
Comunicaciones a media distancia.....	232
Clasificación	233
Ventajas y desventajas de la RF	234
Usos de la RF.....	235
VHF.....	236
UHF.....	237
Dispositivos RFID	237
Wi-Fi.....	238
Historia	239
Estándares	239
Seguridad y fiabilidad.....	240
WiMAX.....	242
Funcionamiento.....	243
Comunicaciones a larga distancia.....	244
Sistemas satelitales.....	248
Ventajas y desventajas.....	252
Resumen	253
Actividades	254

07*Fuentes de alimentación**

Fuentes lineales y conmutadas.....	256
Fuentes lineales.....	260
Fuentes conmutadas	269
Las tres configuraciones básicas.....	272
Diseño online de una fuente conmutada	273
Diseño y reguladores	279

Reguladores de voltaje.....	285
Generación alternativa de 220 V.....	291
Generadores a combustible.....	296
Resumen	299
Actividades	300

08*Las fuentes de alimentación y la robótica**

Anatomía de los robots	302
Tipos y características de los robots.....	304
Componentes de un robot.....	305
El controlador	306
Baterías	308
Plomo y ácido.....	308
Níquel-cadmio.....	310
Alcalinas	310
Níquel-hidruro metálico	310
Litio-ion	310
Dispositivos especiales	311
Grados de libertad.....	311
Zona y volumen de trabajo	311
Precisión de los movimientos	312
Configuraciones y morfología	313
Fuentes de alimentación.....	317
Microcontroladores aplicados a la robótica	318
Resumen	319
Actividades	320



BIBLIOTECA DIGITAL



Nuestro catálogo listo para
leer en PC o iPad



¡LEELO COMO
QUIERAS!

Ahora puedes acceder
a los libros USERS a
través de Internet,
no más esperas ni
problemas de envío

¡LEELO DONDE
QUIERAS!

Con la calidad de
siempre, abarcando los
temas del momento,
incorporando servicios
complementarios y
pagando menos

NUEVAS ediciones a  **COLOR**

Accede a un libro de muestra
www.redusers.com/u/2hq

ACCEDE A LA BIBLIOTECA DIGITAL **USERS**



usershop.redusers.com



usershop@redusers.com



+54(011)4110-8700

Introducción



Esta última entrega de la colección Electrónica propone un recorrido por la historia del movimiento de open hardware y software que servirá como introducción para el posterior desarrollo de las plataformas abiertas más utilizadas en la actualidad: Arduino y Raspberry Pi, las dos plataformas que mejor se posicionaron en el mercado, junto a sus entornos de desarrollo.

Hoy en día, compartir el conocimiento y crear proyectos puede ser la base para que otros puedan reproducirlos o mejorarlos, tanto en los circuitos como en los códigos fuente.

Abordaremos también una de las temáticas más apasionantes de la tecnología, la robótica, ya que en ella se puede encontrar la electrónica directamente aplicada. Aquí conoceremos aspectos relacionados con la micromecánica y otros que ayudarán a entender los conceptos más importantes para la creación de equipos que realicen tareas automatizadas.

A lo largo de los capítulos, el lector irá adquiriendo conocimientos teórico-prácticos, en conjunto con una serie de ejemplos y explicaciones prácticas que ayudarán a comprender, de manera cabal, los temas y las necesidades que debe tener en cuenta un técnico al manejarse con dispositivos y circuitos electrónicos.

Recomendamos que, a partir de este enfoque, cada uno pueda tomar los problemas y prácticas planteados como un desafío personal, para que los resultados no constituyan solo un conjunto de plaquetas y pequeños aparatos ensamblados, sino un camino hacia el aprendizaje, que incluya el entretenimiento y la diversión.

De esta forma, la teoría no será solo una cuestión de elementos conceptuales, sino una forma de comprender nuestro mundo y la tecnología que utilizamos a diario. Si nos ponemos a pensar un instante, reconoceremos que todo el tiempo estamos interactuando con decenas de dispositivos electrónicos, desde un simple auricular o un teléfono celular hasta una PC, un monitor, un horno de microondas, una heladera... de hecho, ¡cualquier electrodoméstico!

Plataformas abiertas

Durante los últimos 30 años, el movimiento de código y software libre ha ganado mucha popularidad. En este capítulo haremos una analogía entre los circuitos y el código fuente, para que queden a disposición de cualquier persona, que diseñe otros, o bien los utilice tal como fueron diseñados.

▼ Historia de un estilo.....14	▼ Usos comerciales39
GNU 14	
Más allá del software 16	▼ Proyectos de hardware libre....44
El software libre..... 17	Elphel..... 44
Modificaciones..... 20	RepRap..... 46
Linux 21	OpenSPARC 48
Copleft 22	OpenBook 53
	TuxPhone..... 53
▼ Open hardware.....22	Oscar..... 55
Filosofía..... 23	Open Graphics Project 57
Desafíos..... 27	Arduino..... 58
Proyecto Arduino..... 29	Nano Satellite..... 58
▼ Historia y evolución31	▼ Resumen.....59
▼ Modelo abierto para hardware.35	▼ Actividades.....60



Historia de un estilo

El contenido libre es el concepto que describe cualquier clase de contenido creativo que no posea ningún tipo de restricción legal en relación con el derecho de uso, distribución y modificación. Esto trae consigo la falta de protección ante un posible robo, ya que el contenido libre no posee derechos reservados.

El contenido libre incluye a todo el material del dominio público, que define las obras que pueden ser copiadas y reproducidas por cualquier otra persona o grupo, así como aquellas que posean derechos de autor pero que todavía estén sujetas a una licencia que proteja dichas libertades.

UNA PLATAFORMA
ABIERTA DE
CIRCUITOS UNE
SOFTWARE LIBRE
CON CIRCUITOS
ELECTRÓNICOS

La diferencia entre contenido libre y contenido de código abierto –*open source* en inglés– se encuentra fundamentalmente en sus valores y formas de mirar el mundo, ya que, para el movimiento de código abierto, se trata de una cuestión más práctica que social. Visto de ese modo, el código abierto es una metodología de desarrollo, porque un equipo de desarrollo puede

tercerizar su trabajo haciéndolo de código abierto. De esta manera, se obtienen avances en el proyecto y personas que trabajan en él sin tener que preocuparse por pagar salarios.

GNU

GNU es el proyecto iniciado por **Richard Stallman** con el objetivo de crear un sistema operativo completamente libre. Fue diseñado para ser compatible con UNIX, un sistema muy estable pero no libre. La compatibilidad implica que esté compuesto por pequeñas piezas de software individuales, como el sistema gráfico **X Window**.

Para el movimiento de código abierto, el software que no es libre es simplemente una solución ineficiente. Mientras que, para el movimiento de software libre, si no es libre implica un problema social. Por lo tanto, los seguidores del software libre se encuentran en esa posición por una cuestión ética o social más que por una cuestión de producción.

Este pensamiento es llamado **cultura libre** y es el encargado de promover la libertad en la distribución y modificación de trabajos creativos en base al principio de contenido libre. Este movimiento se opone a las medidas restrictivas de las leyes de **copyright**, que describen los derechos de autor, argumentando que obstaculizan la creatividad y que la idea de restringir legalmente es arcaica.

Dentro de la cultura libre nos encontramos con diferentes **niveles de libertad** debido a los distintos tipos de licencia existentes, como las obras de dominio público, que tienen libertades limitadas bajo las licencias **Creative Commons**. En la cultura libre, Creative Commons ha sido criticada por su falta de estándares de la libertad, ya que estas licencias pueden tener algunos derechos de copia o copyright o carecer de ellos. Estas restricciones de Creative Commons se evitan obteniendo una licencia de software libre a través del proyecto GNU y la incorporación de sus licencias, como GPL. Esta es la más utilizada en el mundo del software libre, ya que garantiza a todos los usuarios finales la posibilidad o libertad de utilizar, compartir o modificar el software en cuestión.

Sin embargo, estas licencias determinan un usuario y un producto final (o bien, lo suficientemente terminado como para que se pueda usar y compartir). De aquí surge el término para definir el software libre que se encuentra en desarrollo.



Figura 1. Linus Torvalds es conocido por iniciar y mantener el desarrollo del kernel Linux.

El **diseño abierto** consiste en el avance de productos en desarrollo que dispongan públicamente de la información de diseño. Por lo general, el diseño abierto también implica el uso o la realización de software libre y abierto, así como de hardware de fuente abierta. Este último es aquel cuyas especificaciones y diagramas esquemáticos son de acceso público y libre. Aunque puede ser de acceso gratuito, últimamente se permite su acceso mediante una donación de una

cantidad determinada de dinero al proyecto. Por ejemplo, el proyecto **OUYA** de consolas de videojuegos de código abierto –hoy hecho realidad– requería una donación para poder acceder al kit de desarrollo, o bien para tener la consola antes del lanzamiento masivo (aunque, lamentablemente, esto no se pudo cumplir).

Más allá del software

Además del contenido en software, sus manuales también deben ser libres por las mismas razones que el software debe serlo y porque, desde cierto punto de vista, el manual es parte del software. El concepto de software libre es aplicable a cualquier tipo de publicación, por lo que se ha extendido a la definición de obras culturales libres. Un ejemplo muy popular de esto último es **Wikipedia**.

El proceso de diseño abierto es bastante masivo gracias a internet. Allí se anuncian en plataformas como **Indiegogo**, una página donde se pueden publicar ideas o proyectos en desarrollo para publicitarlos y que ofrece la posibilidad de donar dentro del mismo sitio.

La filosofía del diseño abierto se suele relacionar con la del movimiento de código abierto, ya que de esta forma se terceriza el trabajo brindando kits de desarrollo para los que deseen participar. Pero, a diferencia del movimiento de código abierto, en este caso se trata de productos físicos: hardware.



Figura 2. Richard Stallman es el fundador del movimiento por el software libre en el mundo.

En la actualidad, el movimiento social de diseño abierto es bastante primitivo pero presenta un crecimiento notorio, ya que en sus inicios era aún más complicado que estos proyectos o ideas se hicieran realidad y terminaran siendo distribuidos en forma masiva. Por ejemplo, **Ubuntu Edge** fue un proyecto de teléfono móvil de alta

gama, que consistía en un teléfono (hardware) con el sistema operativo Ubuntu Mobile (software), que lamentablemente no llegó a su meta de 32 millones de dólares, porque es probable que fuera un poco alta, si se tiene en cuenta que los seguidores y usuarios de Ubuntu no suelen tener que pagar por los productos.

Figura 3. El proyecto GNU fue iniciado por Richard Stallman con el objetivo de crear un sistema operativo completamente libre.



A pesar de las plataformas para difusión y conocimiento alrededor del mundo, existen ciertas barreras que se deben superar dentro del diseño abierto comparándolo con el desarrollo del software de código abierto. Una de ellas es la distribución. El software de código abierto puede distribuir su código sin ningún costo; pero, por el contrario, el hardware de diseño abierto posee un costo de envío. Por otro lado, antes de la distribución están, también, los costos de creación del producto en cuestión; en cambio, el software solo requiere una computadora con la capacidad de correr un IDE para programar y compilar.

En paralelo a la creación del producto, se encuentra la modificación, que implica dificultades o gastos en el caso del hardware de diseño libre. Por ejemplo, si necesitamos hacer algún arreglo o mejora en una placa de circuito impreso, por lo general, debemos construirla de nuevo, aunque depende de la tecnología utilizada para el armado. En el caso del software, basta con unas correcciones en las líneas de programa y una espera en la compilación para que los cambios sean realizados.

El software libre

La definición de software libre establece los criterios que se deben cumplir para que un programa sea considerado libre. Esto implica que se respeta la libertad del usuario para copiar, distribuir, modificar y

mejorar el software. Todas estas libertades están basadas en la definición brindada por el sitio de GNU, aunque podemos deducir que, teniendo el código de un programa, las libertades son mucho más amplias.

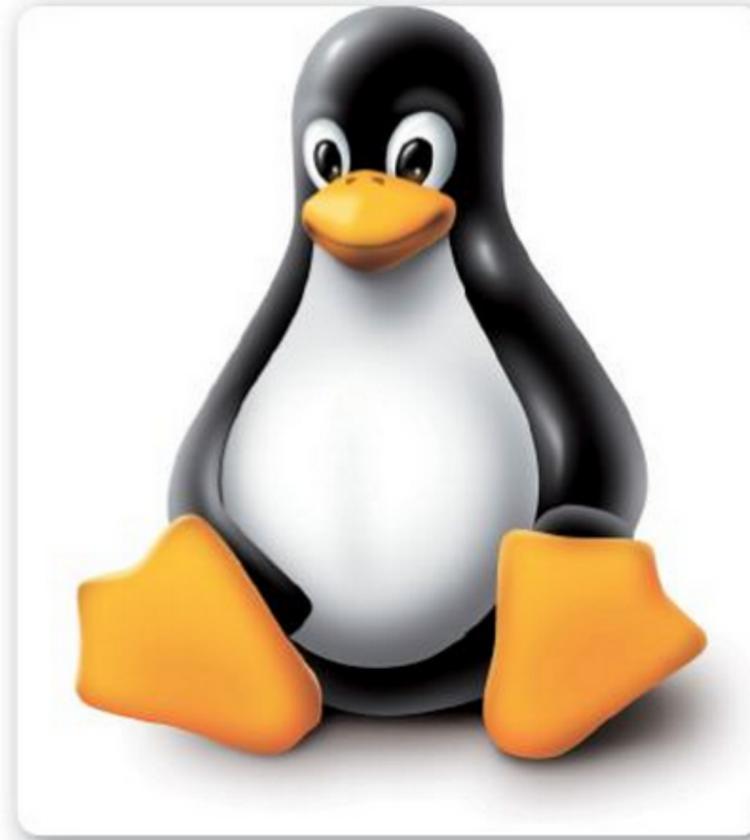


Figura 4. Tux es el nombre de la mascota oficial de los sistemas Linux. Fue creada por Larry Ewing en 1996.

Las libertades esenciales que se suelen mencionar dentro del movimiento del software libre son:

- Libertad 0: Ejecutar el programa para cualquier propósito.
- Libertad 1: Estudiar cómo funciona el programa y cambiarlo para que haga lo que uno desea.
- Libertad 2: Redistribuir copias para ayudar al prójimo.
- Libertad 3: Distribuir copias de sus versiones modificadas a terceros para ofrecer a toda la comunidad la oportunidad de beneficiarse con las modificaciones.

Por lo tanto, un programa que permita todas las características y libertades mencionadas se considera libre. Un usuario puede entonces copiar –con modificaciones o sin ellas– un programa, ya sea de manera gratuita o cobrando una tarifa, para cualquiera en cualquier parte. Vale aclarar también que tener la libertad de hacer estas cosas no

implica que no se debe pedir ni pagar el permiso. Sin embargo, también existe la libertad de hacer modificaciones y usarlas en privado para el propio beneficio, sin tener que notificar a nadie en particular.

La libertad de ejecutar el programa significa que cualquier persona u organización es libre de usarlo en cualquier tipo de sistema de computación, con cualquier propósito, sin obligación de notificar o informar al programador o a ninguna otra entidad específica. Lo importante de esta libertad es el propósito de los usuarios y no del programador. Como usuario, uno es libre de ejecutar un programa para lo que necesite.

La libertad de redistribuir copias debe incluir las formas binarias o ejecutables del programa, así como el código fuente. Aunque se acepta que para algunos programas no existe un modo de producir un formato binario o ejecutable, la distribución en forma ejecutable es necesaria para que los sistemas operativos libres puedan instalar los programas fácilmente.

Para que la primera libertad y la tercera (realizar cambios en el programa y publicar las versiones modificadas) tengan sentido, se debe tener necesariamente acceso al código fuente del programa.

Por lo tanto, el acceso al código fuente es una condición necesaria dentro del software libre.

La primera libertad incluye el uso de versiones modificadas en lugar de la original. En el caso de que el programa se entregara con un producto diseñado para ejecutar versiones modificadas de terceros pero rechazara ejecutar las propias, se estaría realizando una práctica conocida como **tivoización** o **tivoization** (también denominada **lockdown**). Entonces, la libertad 1 se convertiría en una ficción teórica en vez de una libertad práctica. Estos binarios no serían considerados dentro de la categoría de software libre, incluso si fueran compilados desde un código fuente que haya sido libre.

La libertad 3 incluye la libertad de publicar versiones modificadas del programa como software libre. Una licencia libre también puede permitir otras formas de publicación, por lo tanto no es necesario que sea una licencia **copyleft**, que es el método generalmente utilizado para hacer un programa libre, exigiendo que todas las

EL SOFTWARE LIBRE
LE DA LIBERTAD
AL USUARIO PARA
UTILIZARLO SIN PEDIR
NI PAGAR PERMISO



versiones modificadas también sean libres. No obstante, una licencia que requiera que las versiones modificadas no sean libres, también puede ser considerada libre.

Para que dichas libertades sean reales, deben ser permanentes e irrevocables siempre que los usuarios no cometan ningún error. Ahora bien, si el programador tiene el poder de revocar una licencia o añadir restricciones a las condiciones de uso o a alguno de los aspectos vistos anteriormente sin que haya habido ninguna acción de parte del usuario que lo justifique, el software dejará de ser considerado libre.

Ciertos tipos de reglas sobre la manera de distribuir el software son aceptables siempre y cuando no entren en conflicto con ninguna de las libertades principales. De esta forma, las reglas del copyleft, por ejemplo, no entran en conflicto con las libertades principales; por el contrario, las protegen.

A pesar de todas las libertades mencionadas, **software libre** no significa **gratuito** o **no comercial**. Un programa libre también debe estar disponible para el uso, la programación y la distribución comerciales. La programación comercial del software libre ya no es tan inusual. Puede haberse pagado dinero para obtener copias de software libre, o bien pueden haberse obtenido copias sin costo alguno. Sin tener en cuenta cómo se obtuvieron las copias, siempre existe la libertad de copiar y modificar el software, e incluso venderlo.

Modificaciones

Las modificaciones no necesariamente son mejoras; eso es un asunto bastante subjetivo. Si el derecho a modificar un programa se limita, básicamente, a modificaciones que alguna otra persona considera una mejora, el programa no es libre.

No obstante, eventuales reglas sobre cómo se empaqueta una versión modificada son aceptables mientras no limiten de manera sustancial la libertad para publicar versiones modificadas. Así, se acepta que una licencia obligue a cambiar el nombre de la versión modificada, a eliminar el logotipo o a identificar las modificaciones como propias. Estas obligaciones son aceptables siempre y cuando no sean tan agobiantes como para dificultar la publicación de las modificaciones. Como ya se realizaron modificaciones al programa, no le supondría una dificultad al programador hacer algunas más.

Un problema particular se presenta cuando la licencia requiere que a un programa se le cambie el nombre con el cual será invocado por otros programas. De hecho, este requisito dificulta la publicación de la versión modificada para reemplazar a la original, cuando sea invocada por los demás programas. Este tipo de requisitos dentro de una licencia es inaceptable, a menos que exista un instrumento adecuado para la asignación de alias que permitan especificar que el nombre original del programa es un alias de la versión modificada.

Todas estas son cuestiones actuales que surgen por ser el software libre parte de una sociedad. Pero no hace mucho tiempo, entre los años 1960 y 1970, el software no era considerado siquiera un producto, sino un añadido que una computadora aportaba a sus usuarios. En este ambiente, era común que los programadores y desarrolladores de software, cuyos trabajos pertenecían a la compañía que vendía las computadoras, compartieran sus ideas libremente entre sí.

En 1971, las personas hacían uso de la informática, antes de su boom, en ámbitos universitarios y empresariales, creaban y compartían el software sin ningún tipo de restricciones.

Durante la década de los 80, la situación comenzó a cambiar. Las computadoras más modernas empezaron a utilizar sistemas operativos privativos, poniendo a los usuarios en una situación restrictiva, desde el punto de vista del movimiento de software libre, ya que naturalmente no se le podían realizar modificaciones al programa.

Linux

Linux es el núcleo libre del sistema operativo basado en UNIX. Es uno de los ejemplos más populares de software libre. Fue creado en 1991 por un estudiante de Ciencias de la Computación llamado Linus Torvalds. Se encuentra bajo la licencia GPL v2 y se desarrolla con colaboradores de todas partes del mundo. De este núcleo, salen conocidas distribuciones como Ubuntu o Android.

Existe un antecedente que cuenta que, por ese mismo año de 1991, el señor Richard Matthew Stallman trabajaba para un laboratorio donde

EL DERECHO A
MODIFICAR UN
PROGRAMA NO
NECESARIAMENTE SE
LIMITA A UNA MEJORA



tenían una impresora que no funcionaba a la perfección: el dispositivo se atascaba cada cierta cantidad de impresiones. Como la pérdida de tiempo era demasiada, decidió pedirle el código fuente de la impresora a la empresa proveedora para modificarlo y agregar un aviso de red cada vez que esta se atascaba.

La empresa se negó a darle el código pero le ofreció una nueva versión del software propietario. En ese momento, Stallman se vio en un dilema: si firmar acuerdos de no revelación y acabar desarrollando más software propietario con licencias restrictivas o negarse a firmar.

Con este antecedente, en 1984, Richard Stallman comenzó a trabajar en el proyecto GNU y, un año más tarde, fundó la **Free Software Foundation**. Así, introdujo la definición de software libre y el concepto de copyleft, que desarrolló para brindar libertad a los usuarios y para restringir las posibilidades de apropiación del software.

Copyleft

Se trata de la práctica de ejercer derechos de autor y que se permita la libre distribución de copias y versiones modificadas del producto en cuestión. El término surge de las comunidades de **software libre** como un juego de palabras en torno al término **copyright**. Se considera que una licencia libre es **copyleft** cuando, además de otorgar permisos para uso, copia y modificación, contiene una cláusula que dispone una licencia compatible con las copias y las obras derivadas.

Open hardware

Conocemos como **hardware abierto** al conjunto de diseño y especificaciones de dispositivos electrónicos que es de público conocimiento, que permite la libertad de uso, estudio, modificación, distribución y redistribución de las versiones modificadas.

En la mayoría de los casos, no podemos decir que un hardware es 100 % libre de patentes, que es de fácil modificación y, mucho menos, que es de fácil redistribución, aunque existen puntualmente algunas excepciones, tal como el microprocesador de **UltraSparc**.

Este nos permite que quien pueda y quiera utilice el diseño y la especificación para programar software más fácilmente, o diseñe nuevos procesadores en base a él y vuelva a distribuir su diseño.

Podemos clasificar los dispositivos de acuerdo con su naturaleza o la filosofía sobre la cual se asientan.

Así, tenemos el **hardware reconfigurable**, cuya estructura puede configurarse mediante software (lógica programable). Su naturaleza es completamente diferente a la del **hardware estático**, ya que este está ideado para funcionar en un solo sentido o para una sola función. Como el hardware reconfigurable depende del software, se le puede aplicar fácilmente la licencia de software libre.

El hardware estático libre es más complicado de ser desarrollado por su propia naturaleza. Aunque se le puede configurar software para que actúe de manera distinta o tenga otra finalidad, su funcionalidad sigue siendo la misma, pese a que su diseño sea liberado.

EL HARDWARE
LIBRE INTRODUJO
INNOVACIONES
TECNOLÓGICAS EN
PAÍSES IMPENSADOS



Filosofía

En cuanto a la filosofía por la cual podemos guiarnos para una clasificación, existen cuatro tendencias. Una de ellas es el diseño de hardware libre, que especifica que la forma en que se comparte cómo está estructurado el dispositivo debe ser la que se publica como software libre. Esto significa que el diseño debe tener libre disponibilidad para su distribución, estudio, modificación y redistribución, lo que no implica que no pueda comercializarse.



DISEÑO, HARDWARE Y CÓDIGO ABIERTO



Las licencias de código abierto pueden utilizarse en el firmware y el diseño de hardware abierto, mientras que para el hardware deben utilizarse las licencias OHW, sin otra alternativa. Algunas de ellas son lo bastante abiertas como para que los proyectos que las integran puedan ser comercializados; igualmente, depende de cada país que esto sea legal o no lo sea.

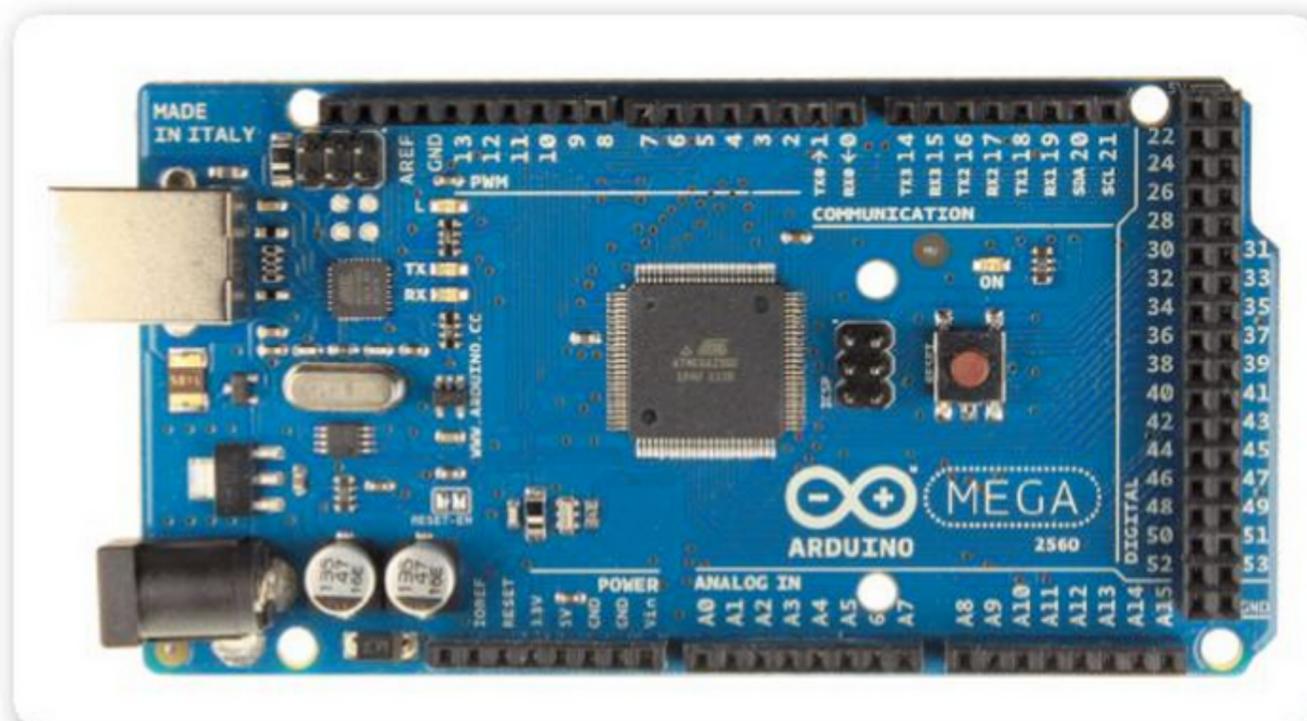


Figura 5. Arduino se utiliza tanto en creaciones simples de componentes como en prototipos de robótica con fines didácticos, lo cual no impide su comercialización intensiva.

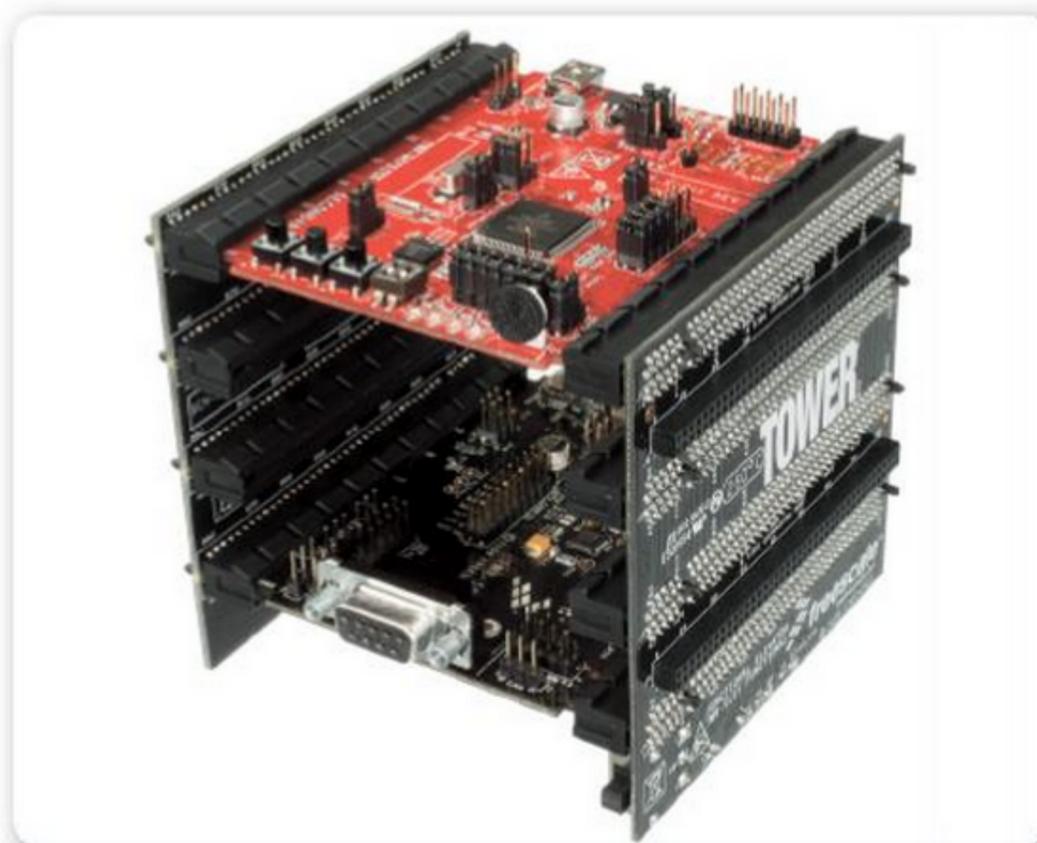


Figura 6. La lógica del hardware reconfigurable es que el mismo dispositivo dentro de distintos entornos pueda efectuar, mediante programación, funciones distintas.

También nos encontramos con la filosofía del hardware de código abierto, que permite compartir su diseño, aunque algunas de sus partes estén como en una **caja negra**, o sea, vedadas a la libre distribución por un tema de patentes.

En otra dirección, tenemos el hardware abierto propiamente dicho, que es una marca registrada del **Open Hardware Specification Program** ; una forma limitada de hardware de código abierto. Básicamente, lo que podemos hacer bajo esta filosofía es generar software (controladores) para el dispositivo, en base a la documentación disponible para el público general, liberada por el fabricante.

Por último, tenemos el hardware libre (en el sentido de ‘gratis’), que es más bien una filosofía utópica, ya que todo hardware tiene costo incluido en sí mismo, excepto cuando se realizan tareas de índole social (*free computers* , por ejemplo).

Desde que este tipo de tecnología llegó a su auge, se han desarrollado un gran conjunto de proyectos que se basan en este concepto. Tenemos desde cámaras reconfigurables, impresoras 3D, diseño de CPU, teléfonos, vehículos y motores de robótica.

Algunos de estos proyectos están actualmente en proceso de desarrollo y otros han sido abandonados. Pero aun los que no han sido sostenidos en el tiempo brindan abundante documentación y conocimientos sobre distintas áreas de la electrónica.

El movimiento de hardware libre está, en la actualidad, íntimamente ligado al movimiento de software libre, por lo cual es muy común tener asociados ambos conceptos en algunos ámbitos, aunque taxativamente provienen de orígenes diferentes.

Algunas empresas han visto con buenos ojos esta tendencia, ya que les permite impulsar la venta de sus productos con finalidades distintas a las que tenían en un principio, y en nuevos mercados. De esta forma, se han desarrollado emprendimientos a la sombra del hardware libre en países en desarrollo, que permiten que la brecha tecnológica sea menor a la que se tendría sin estas herramientas.

El área donde más se está desarrollando el principio de esta filosofía es en la educación, ya sea para la construcción de hardware o para la utilización de los dispositivos electrónicos propiamente dichos.

La facilidad de acceso en los países desarrollados a los elementos de construcción más avanzados (procesadores, microcontroladores, motores) ha permitido que los proyectos más ambiciosos estén evolucionando allí. Sin embargo, las zonas que tienen pocas

HASTA LAS FILOSOFÍAS
RESTRICTIVAS
AYUDARON A
EXPANDIR EL
HARDWARE LIBRE



restricciones de importación están sacando ventaja de la facilidad con que se pueden obtener estos elementos y compiten, casi en igualdad de condiciones, con las primeras economías del mundo en este campo.

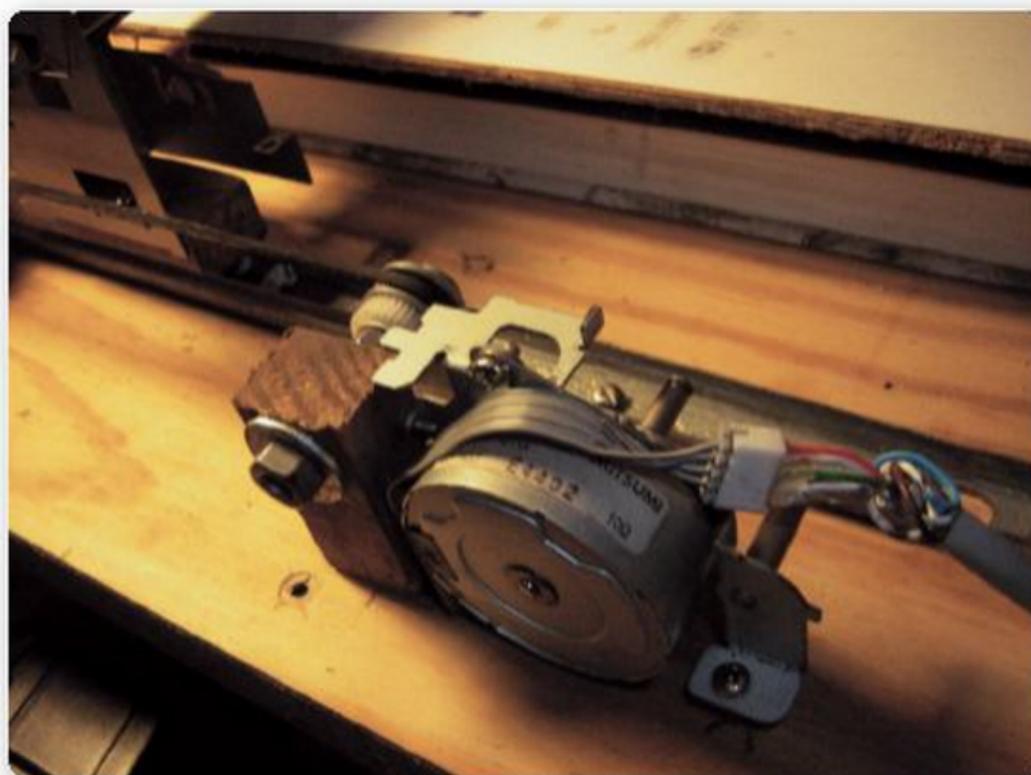


Figura 7. Algunos proyectos de hardware libre son lo suficientemente revolucionarios como para que las empresas comiencen a prestarles atención.

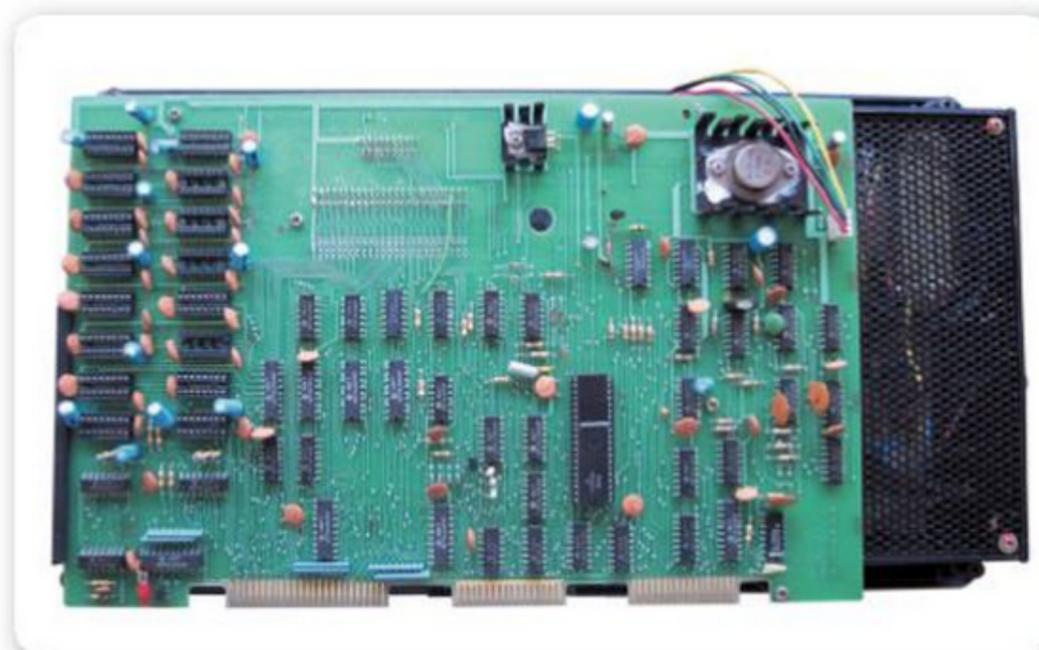


Figura 8. El hardware de los 80 puede reproducirse debido al diseño abierto que se orquestó en esa época en los clubes de electrónica.

Por otro lado, se han creado muchos clubes de desarrollo donde se comparten ideas, proyectos, diseños y otros temas relacionados con el hardware abierto, el software libre e, incluso, el modelo de negocios de

estas filosofías. Estos grupos de internet que discuten libremente sobre el camino a tomar son quienes delinear el futuro de cómo se compartirá este tipo de conocimiento. De la misma forma que sucede en el software libre, hay opiniones contrapuestas sobre qué hay que liberar al público y qué no, cómo se debe colaborar o quién será el dueño del producto resultante final. Muchas de estas cuestiones son las que generan la apertura de nuevas filosofías y la fundación de nuevas formas de licencias, lo que demuestra el vigor de esta corriente de pensamiento.

Desafíos

Entre los principales problemas y desafíos que deben sortear los seguidores del hardware libre están los altos costos de producción, el conocimiento (que poseen pocas empresas), así como que gran parte del tiempo de diseño se destina a proyectos redundantes. Muchos de estos inconvenientes derivan del hecho de que la inversión de tiempo en diseño solo está al alcance de grandes empresas y no de los pequeños emprendimientos. Esto se da particularmente en el ámbito del hardware estático, ya que con la aparición del hardware dinámico este hecho cambió y, en la actualidad, está viviendo una etapa de franca expansión.

Para solventar estos inconvenientes, los clubes y foros creados en internet tratan de obtener, en conjunto con las grandes corporaciones, incentivos para que otros emprendimientos desarrollen nuevos productos en el área educativa o experimental, en base al hardware abierto.

La mayoría del hardware que se logró liberar mediante esta simbiosis se enmarca dentro de productos de enseñanza en el campo de la robótica y se ha dado principalmente en Sudamérica y Asia.



HARDWARE DINÁMICO Y FIRMWARE



Desde que Arduino entró en el mercado del hardware abierto y su firmware permitió que cumpla distintas funciones, se lanzaron muchos proyectos similares. Entre estos, **Tessel**, una placa que puede ser programada en JavaScript, lo que le da mayor flexibilidad porque su lenguaje está más difundido y, además, es simple para desarrollar una interfaz gráfica.

Dentro de estas regiones, lideran el campo Argentina, China (con la excolonia Hong Kong) y Corea, quienes tienen proyectos con apoyo gubernamental para el desarrollo de hardware de enseñanza en colegios secundarios.

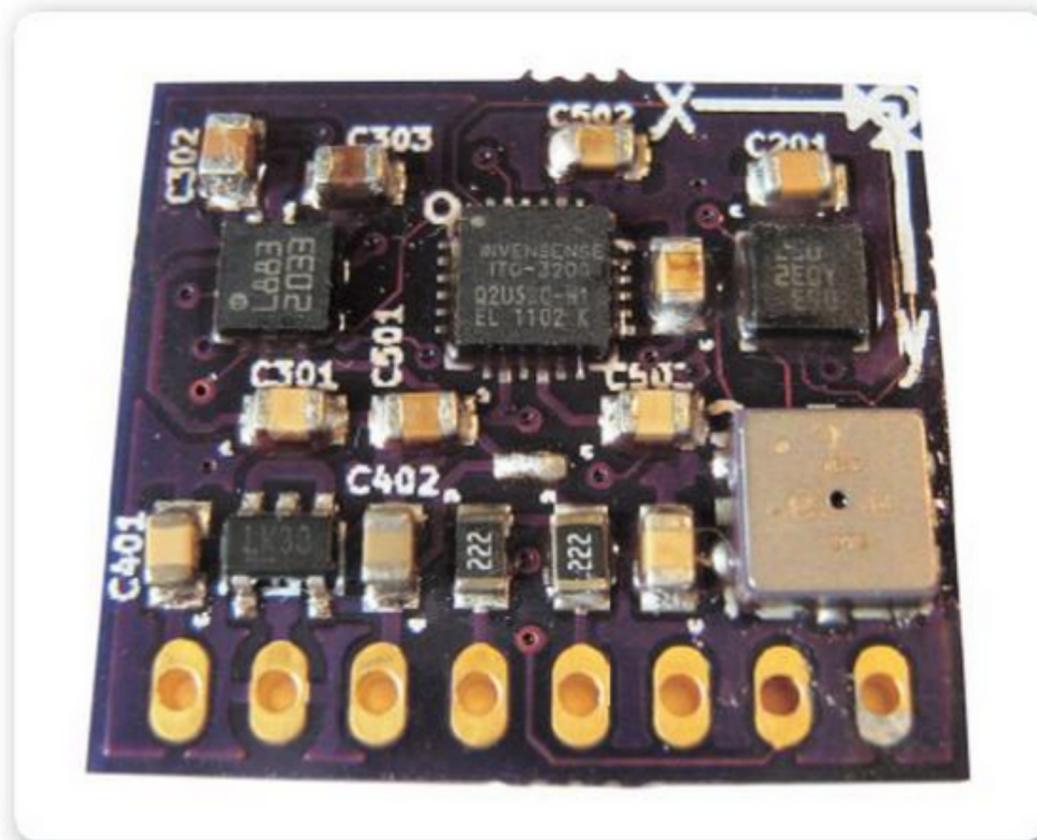


Figura 9. Desde el éxito de Arduino, muchos otros fabricantes comenzaron a producir elementos compatibles con esta plataforma.

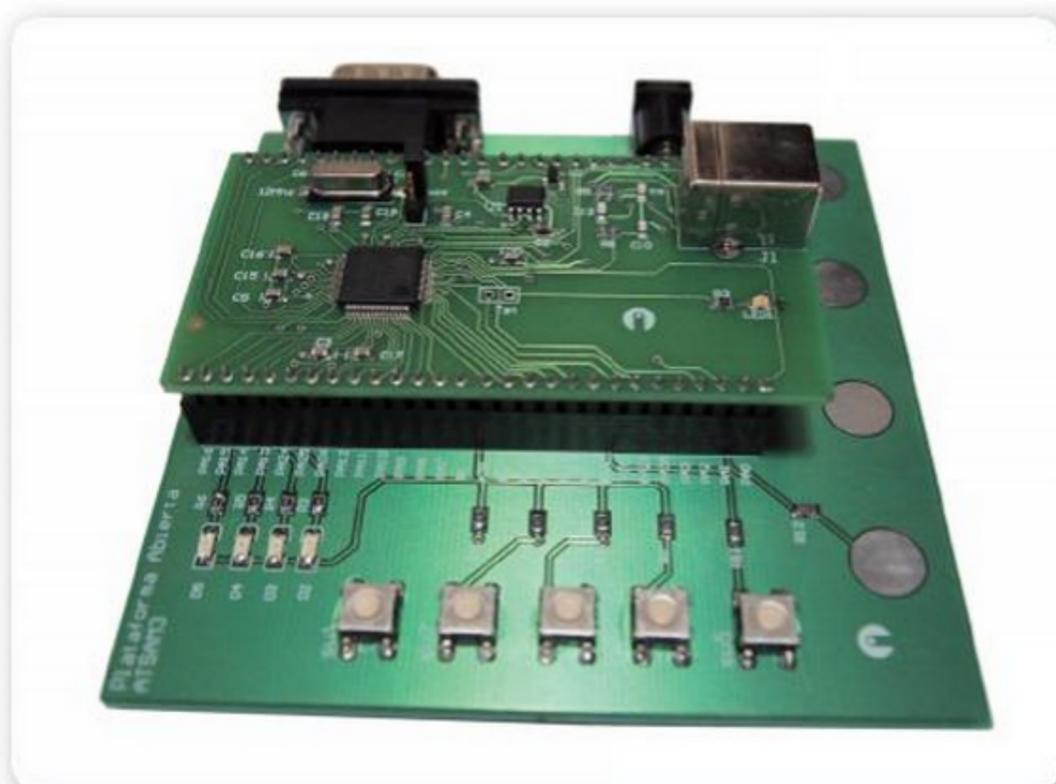


Figura 10. Las licencias del hardware abierto se refieren también al diseño de los componentes, muchas veces protegidos por patentes o derechos de autor.

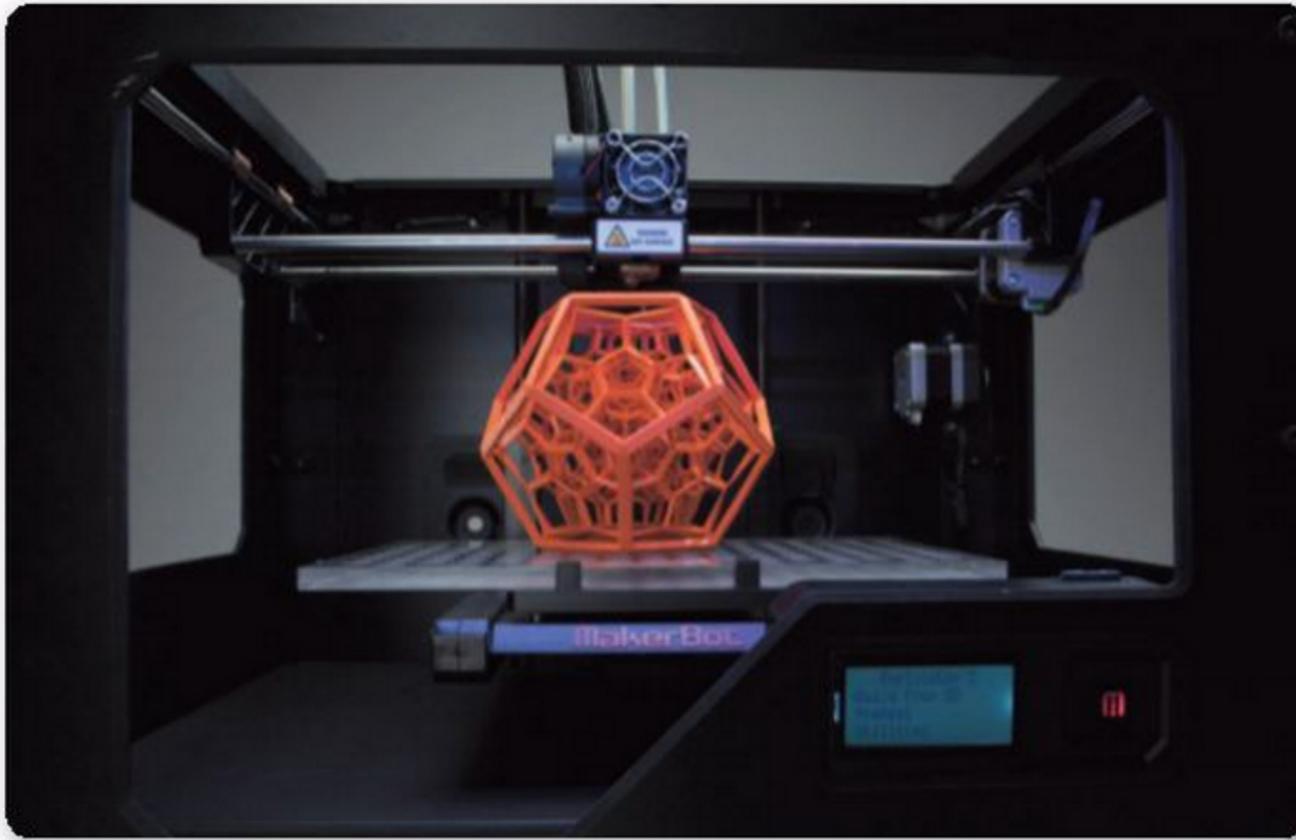


Figura 11. Uno de los últimos productos estrella del hardware abierto es la impresora 3D que se presentó en 2011.

Proyecto Arduino

Especial mención debe recibir el proyecto Arduino, que se compone de una placa con microcontrolador y entorno de desarrollo. Nació en 2005, como los ya comentados, con un fin educativo. El proyecto ha ido creciendo y de él emanan cientos de clones y versiones compatibles que han permitido a miles de estudiantes y autodidactas la creación de nuevas herramientas electrónicas o productos.

Muchas empresas de tecnología han apostado a la compatibilidad de sus productos con esta placa, sin contar la infinidad de software libre que se suma día a día.

Otro de los hitos que logró Arduino fue la fuerte penetración en el ámbito académico internacional. Tanto es así, que estas universidades adoptaron sus componentes para sus propios campus tecnológicos.

En la actualidad, la placa Arduino ya tiene componentes de 32 bits para elaborar y manejar tareas más complejas. Esto se demostró en 2011 con la presentación de la primera impresora 3D, que permite imprimir en resina cualquier modelo tridimensional.

LA RELACIÓN ENTRE
SOFTWARE LIBRE Y
HARDWARE ABIERTO
ES INALIENABLE Y
HASTA CONFUSA



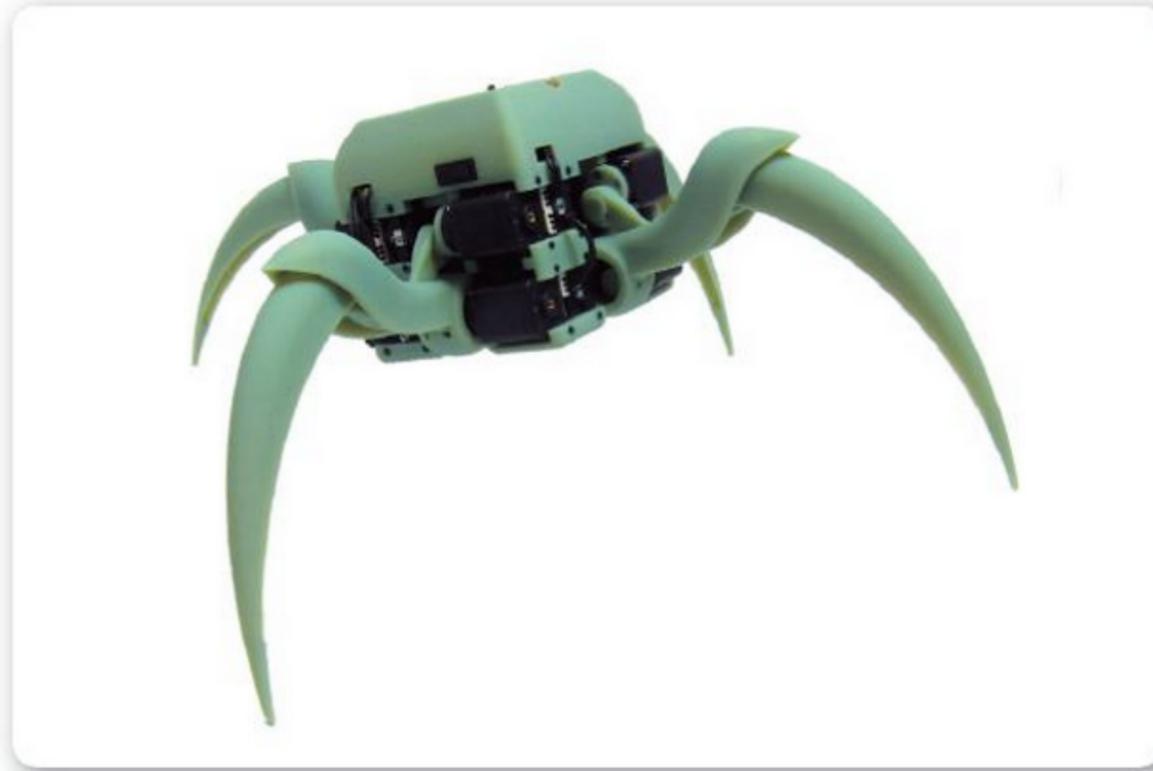


Figura 12. Muchos proyectos novedosos hacen uso del hardware abierto para lograr modelos tan interesantes como este robot araña.



Figura 13. Varias de las piezas que componen la motocicleta conocida como **Brammo Enertia** se fabricaron directamente con impresoras 3D. De esta forma, podemos ver el enorme potencial de estos dispositivos.



CPU OPEN HARDWARE



El microprocesador **UltraSparc 2** de Sun Microsystem es el primer CPU de diseño abierto, lo que permite que otros desarrolladores creen los suyos a partir de este. Las especificaciones fueron definidas en la década del 80, pero su actualización constante lo mantiene a la vanguardia de los avances tecnológicos.



Historia y evolución

Desde los comienzos de la informática, en el siglo XX, se comenzó a gestar el movimiento de hardware abierto. En la década de 1970, en el seno de un club electrónico que amalgamaba un movimiento radical universitario, activistas computacionales y aficionados se escribió el primer texto sobre la computación democrática, texto antecesor del hardware abierto.

Desde aquella época, en los 70 y hasta los 90, la mayoría de las publicaciones sobre hardware abierto era tratada en los círculos de electrónica, de computación y, generalmente, en revistas especializadas. El acceso a la información era muy dificultoso, así como también lo era obtener los materiales necesarios para realizar algún modelo publicado.

Ya en los 90 surgieron los **FPGA** (o campos de arreglo con compuerta programable), que facilitan el acceso a poderosos medios de computación y de electrónica de avanzada en pequeñas pastillas de silicio. También, junto a esto se creó el sitio web de ODC (o circuitos de diseño abierto), que propone formar una comunidad con el mismo espíritu con el cual fue fundado el software libre.

Todo esto estuvo muy bien en la teoría, pero nunca pudo llevarse a cabo por la falta de software libre para el diseño electrónico. Por esta razón, se debatió sobre el desarrollo de una licencia **freeware** pero nunca se puso en práctica la esencia de la proposición.



Figura 14. Un técnico de la NASA con una computadora desarrollada en conjunto con la Universidad de Michigan.

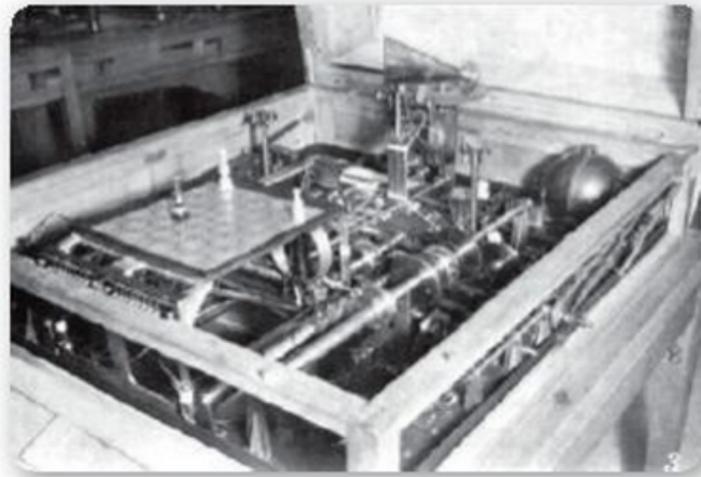


Figura 15. Ejemplo de un hardware transparente, que emula a un jugador de ajedrez y permite ver en su interior todas las piezas que lo componen.

En 1997 se creó oficialmente la definición de hardware abierto, por medio de Bruce Perens, quien ese mismo año lanzó el **Programa de Certificación de Hardware Abierto** (OHCP es su sigla en inglés),

cuyo objetivo era facilitar a los constructores de hardware la disponibilidad de documentación para programar sus dispositivos o crear manejadores de software para ellos. Este programa fue uno de los primeros acercamientos entre las prácticas del software abierto y el hardware abierto.

Más tarde, Perens registró **open hardware** y el dominio **openhardware.org**, donde se podía obtener el programa de certificación.

Poco tiempo después, David Freeman anunció el *Open Hardware Specification Program* (OHSP), una iniciativa tendiente a permitir que el hardware licenciado se interconecte de forma libre, logrando que se cree una plataforma computacional totalmente nueva como contrapartida a los sistemas de cómputo propietarios.

Otras iniciativas similares fueron el *Open Design Circuits* de Reinoud Lamberts y el *Open Design Foundation* de Sepehr Kiani, Ryan Vallance y Samir Nayfeh del MIT.

Luego del nacimiento del concepto, la mayoría de los movimientos mencionados comenzaron a cristalizarse recién a mediados de la primera década del año 2000, con compañías emergentes como OpenCores, RepRap, Arduino, Adafruit o SparkFun.

EL CONCEPTO OPEN
HARDWARE EN ALCANCE
LOS 70, EN CLUBES DE
ELECTRÓNICA
DE LOS EE. UU.

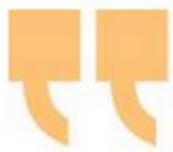




Figura 16. Los clubes de electrónica y los centros de computación y ciencias de los campus de las universidades norteamericanas promovieron el concepto de hardware abierto.

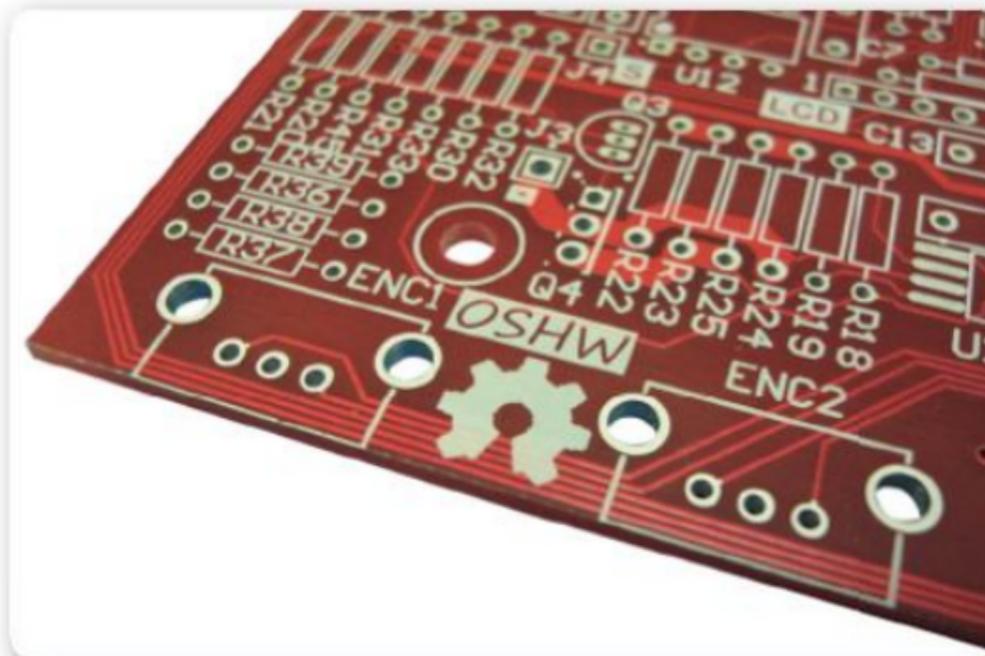


Figura 17. Patrick MacNamara creó la **OHF** (**Open Hardware Foundation**) junto a Transversal Inc., cuya licencia permite comercializar placas PCB de bajo costo y alta tecnología.

En 2011, se creó el *Open Graphics Project (OGP)*, un esfuerzo para diseñar, crear e implementar un conjunto de chips libres y gratuitos de gráficos en 3D. Sin embargo, el primer impedimento se produjo cuando el cálculo financiero para crear los primeros chips era de más de 2 millones de dólares. Como consecuencia de esto, Timothy Miller, el fundador del proyecto, creó la compañía **Transversal Technology Inc.**

Esta compañía, a su vez, se reunió con Patrick McNamara y, juntos, fundaron el *Open Hardware Foundation* (**OHF**) con el fin de resguardar los intereses del OGP. McNamara anunció que Transversal se beneficiaría del bajo riesgo asociado a producir los chips y la comunidad de código abierto obtendría el beneficio de tener hardware disponible a bajo costo o gratis para los desarrolladores.

La organización **TAPR** (*Tucson Amateur Packet Radio Corporation*), fundada en 1982, ofrece ayuda para concretar proyectos dentro de diseños reproducibles logrando que estén disponibles en paquetes

comerciales o productos finales.

En 2005, TAPR comenzó a trabajar en algunos grupos, los cuales desarrollaban software de alto rendimiento y cuyos miembros contribuían con su tiempo libre y su experiencia a la comunidad de radioaficionados. El grupo vio sus esfuerzos coartados por entidades comerciales y, como resultado de esto, nació el **TAPR Open Hardware License**, la primera licencia específica de código abierto.

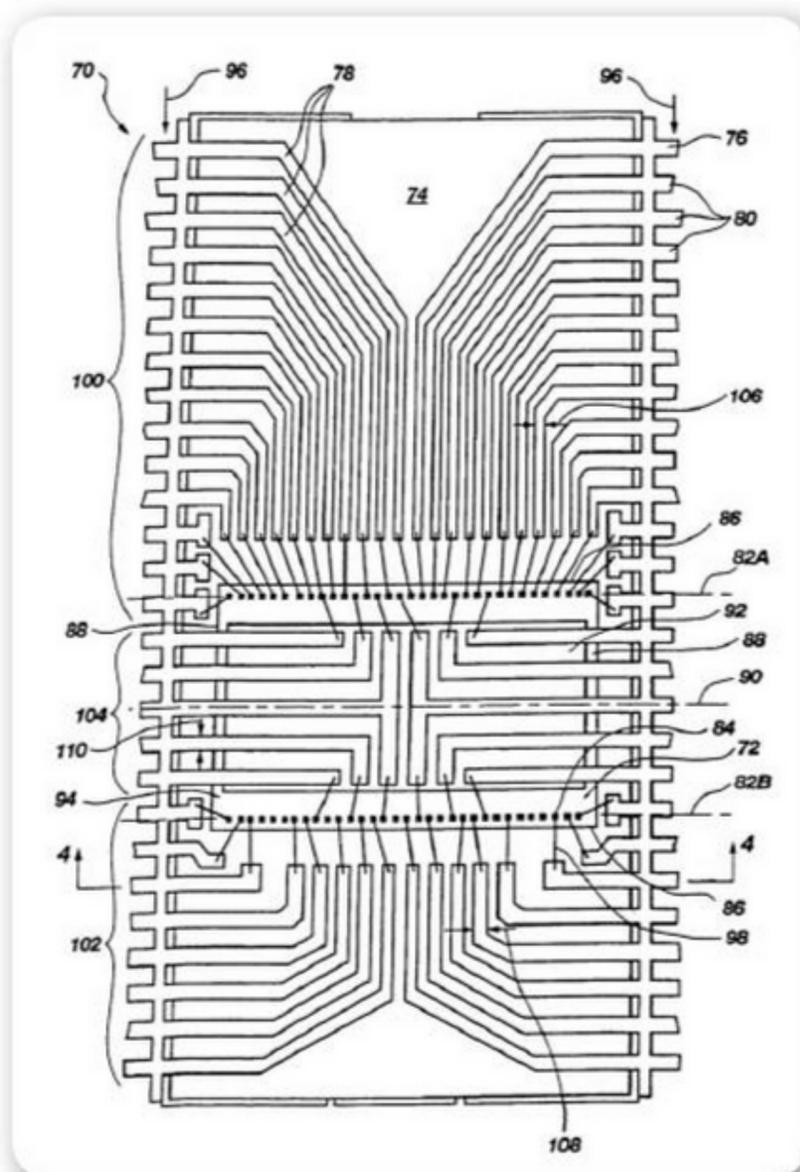


Figura 18. Transversal Technology Inc. produce los primeros chips para los proyectos open hardware, libera su diseño y aporta los fondos necesarios para hacerlos realidad.



OPEN HARDWARE SPECIFICATION PROGRAM

Es una licencia limitada de open hardware, que pone a disposición información para que un programador pueda desarrollar un controlador para un dispositivo en cualquier lenguaje. Es una licencia poco práctica para la comercialización, ya que el periodo de tiempo para que un proyecto se libere es largo en comparación con una producción comercial con patentes y derechos de autor.

Modelo abierto para hardware

Las problemáticas actuales del hardware abierto son diversas y muy distintas a las que podrían darse en el código abierto. Entre ellas, podemos encontrar las siguientes:

- **No hay estándares**, ni unificación, ni mejores prácticas en la creación de documentación de alta calidad. Luego del excelente trabajo de Torrone, Mellis y Seidle para el **Open SourceElectronics**, no hay una clara descripción de qué se debería incluir en la documentación de los proyectos *Open Source Hardware* (OSHW).
- **En internet hay muchas piezas desconectadas de documentación** de hardware, pero la mayoría son pobres en calidad o claridad. Guías claras para clasificación y estructuración pueden ayudar rápidamente con esto. Se podría proponer también un conjunto inicial de normas y guías para ser debatidas y refinadas.
- **La documentación para OSHW está dispersa** a través de muchas plataformas, sitios web, wikis y blogs. Como el número de proyectos está constantemente en aumento, la dificultad para encontrar información se acelera de manera exponencial. La solución es crear una clasificación para identificar fácilmente el hardware y los documentos de módulos, dentro de un repositorio común.
- **No hay un alcance claro sobre qué se define como OSHW**. Si bien los electrónicos de código abierto han sido la faceta más visible del OSHW, hay mucho más detrás de esas siglas.



DEPENDENCIA TECNOLÓGICA

Un problema en los países donde se quiere desarrollar el movimiento de hardware abierto es la importación, ya que se imponen barreras arancelarias. La solución, muchas veces, consiste en crear el mismo componente con otros más simples (por ejemplo, usar transistores en vez de microprocesadores), y esto implica que los proyectos sean más costosos que si se utilizara tecnología de punta.

- **La falta de formatos de normalización** , de una organización clara y la aplicación de toda una jerga técnica hace muy difícil el acceso y la comprensión de la documentación existente por parte del público en general.
- **El lenguaje** es una barrera para la expansión de los planes de open hardware.
- **No hay un formato simple** para mezclar y reunir el hardware subyacente.
- **El trabajo derivado de otro proyecto es difícil de continuar** . Mientras que la proliferación de proyectos es relativamente común, lo complicado es continuar ese proyecto debido a que los usuarios y los desarrolladores deben identificar y decidir qué curso seguir sin tener documentación apropiada al respecto.
- **La falta de software adecuado** para diseñar, mostrar y compartir planes realizados en forma colaborativa.
- **La documentación consume mucho tiempo** , por lo cual, a veces, escasea.
- **El licenciamiento no muy claro** y el miedo a infringir leyes de propiedad intelectual desanima a la gente a producir documentación.



Figura 19. Los diseños de hardware abierto son intercambiables con otros proyectos por su modelo de producción abierto. En este ejemplo, vemos a **Zoybar** , la guitarra eléctrica desensamblable y ergonómica.

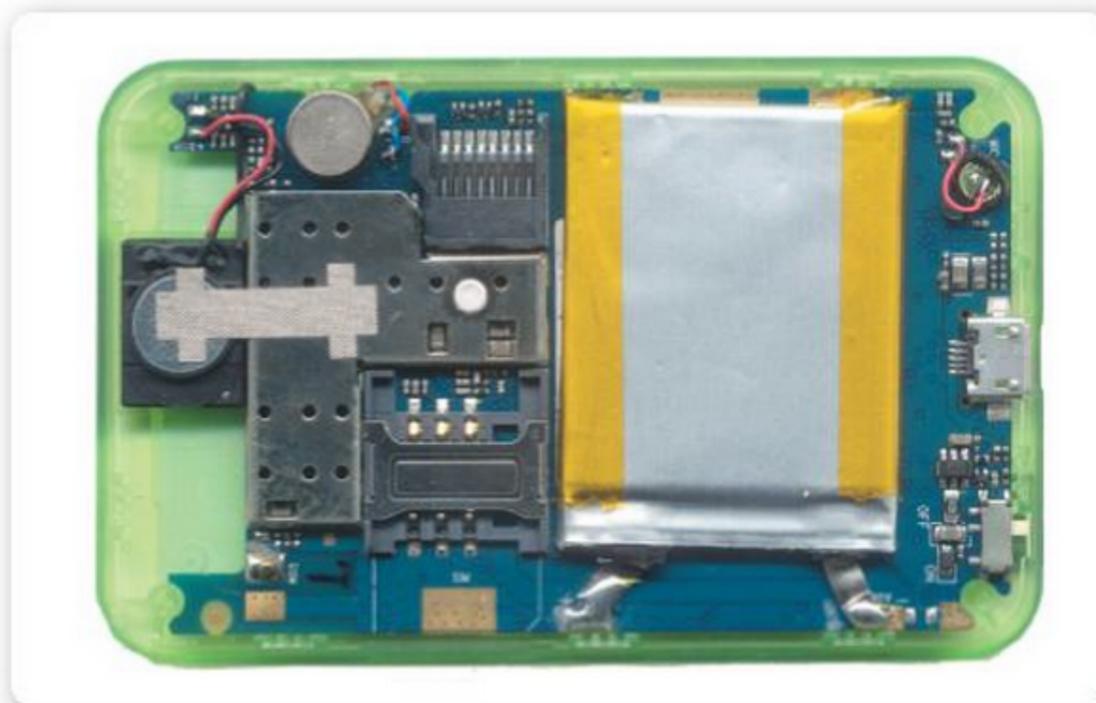


Figura 20. Un elemento de videojuego de diseño abierto publicado por Atmel que permite conectarlo a un televisor y usar algunos juegos simples, como el Tetris.

Sin embargo, dentro de las ventajas que el hardware abierto trae, tenemos muchas más, como, por ejemplo, las siguientes:

- **Fomenta la calidad de los componentes** , el diseño, la estructura y la programación, ya que se basa en estándares abiertos y económicos. La mayoría de los diseños pueden realizarse con componentes básicos de electrónica, aunque el open hardware no se enfoca solamente en este campo, si no en todo lo que sea hardware.
- **Permite la reutilización y la adaptación de diseños** . En este caso, estas técnicas son bastantes anteriores al concepto, ya que la mayoría de ellas fueron creadas en los 70. Así, tenemos el GridBeam que utiliza patrones huecos repetitivos; el Bit Beam que es básicamente un GridBeam a escala minimalista; el Open Beam y el MakerBeam que son sistemas de construcción basados en reglas simples; y, por último, el MakeBlock, que incluye múltiples aproximaciones y módulos eléctricos. Por último, en 2007 se creó el Open Structures, que es un sistema que ofrece muchas más posibilidades.
- **Existen comunidades de diseño** , programación, pruebas y soporte que crecen día a día en forma dinámica y participativa. Esto permite que muchos de los prototipos que salen a la luz como productos ya estén ampliamente testeados y probados, y en la

mayoría de los casos, esto redundará en una mayor velocidad de producción de nuevas tecnologías.

- **En muchos casos, existen clubes que se encargan** de impedir la alianza conocida como **Trusted Computing**, que trata principalmente sobre el control de lo que se permite hacer en una computadora y lo que no; y, a veces, por extensión, en un equipo, microchip o, incluso, en hardware externo.

NO DOCUMENTAR
UN PROYECTO
OBSTRUYE LA
DISEMINACIÓN DEL
HARDWARE ABIERTO



Esto limita el uso legal que se le puede dar a determinados equipamientos, software o hardware de base. Por eso, estos clubes han promocionado el uso del software libre y del hardware libre como medio de impulsar su libre utilización, mejorando el rendimiento de cada equipamiento y optimizando el uso del software. Otro ámbito sobre el que actúa directamente el hardware libre es en la llamada **gestión digital de derechos (DRM)**, una iniciativa de las empresas privadas para limitar el uso de medios o dispositivos digitales. El hardware y el software abiertos permiten que se utilicen sin restricciones la copia y reproducción de elementos multimedia.



Figura 21. Grid Beam en acción: aquí tenemos una carreta manual con múltiples agujeros para reconfigurar el diseño y darle nueva funcionalidad.



Figura 22. **Open Beam** es un modelo de diseño repetitivo que se utiliza en construcción. En esta imagen, vemos un techo con múltiples asociaciones arquitectónicas.

Usos comerciales

Las licencias de hardware abierto que se utilizan para la creación de proyectos nuevos generalmente son confundidas con otras similares tales como GPL, Creative Commons, MIT, BSD. Mientras que algunas de estas se adaptan perfectamente para el firmware o los dibujos CAD, no hacen mucha diferencia en lo que respecta al hardware (en particular, en lo referido a patentes y trabajos relacionados).

Muchos de los miedos que comúnmente aquejan a los elaboradores de un proyecto open hardware se vinculan al hecho de que este sea



PROGRAMAS DRM



El DRM está incluido en todos los dispositivos de reproducción de contenidos digitales. En 1996, el DVD Forum solo permitía reproducir DVD en productos propios. Más tarde, se creó el AACS para HD DVD y Blu-ray, pero ambos sistemas fueron vulnerados. Por lo general, estas prácticas solo logran que el público tenga problemas para reproducir sus archivos y abandone los dispositivos.

utilizado por grandes conglomerados de empresas para sus propios productos y que finalmente lo pongan bajo alguna patente o tipo de restricción hacia el público en general.

Algunos de los licenciamientos que pueden aplicarse para aminorar este tipo de temores es la licencia TAPR OHL, aunque todavía es muy abierta hacia las grandes empresas. Otra licencia es la OSI, que es más restricta, pero va en contra del concepto OSHW.

UN PRODUCTO
OPEN HARDWARE
PUEDE SER
COMERCIALIZADO,
COMO ARDUINO



Lo más realista sería utilizar una licencia restrictiva si lo que se quiere es tener control sobre el proyecto; sin embargo, algunos de los siguientes puntos pueden ayudar en este aspecto mucho más que solo una declaración de licencia.

Entre otros puntos que pueden ayudar a evitar un plagio por parte de empresas, se recomienda: colocar el e-mail, la dirección web o el nombre en la placa PCB de cobre, así como el nombre del proyecto. También sirve hacer un esquema del proyecto y del código. Escribir

pequeños párrafos sobre la contribución realizada en el circuito también es de gran utilidad.

Hay que tener cuidado al utilizar una licencia sin uso comercial, ya que puede también ir en contra de la divulgación propia del proyecto. Un ejemplo muy esclarecedor dentro del ámbito del código libre es Linux. Hoy en día, la mayoría de las computadoras de bajo costo llevan Linux como sistema operativo; si este hubiera sido publicado bajo una licencia sin uso comercial, las empresas ensambladoras de computadoras personales nunca lo hubieran integrado y, por lo tanto, su divulgación hubiera sido más limitada. Otro ejemplo es Firefox.



EL PROBLEMA DE LAS PATENTES



El registro de patentes de una nueva pieza o producto final es un problema para su creador. La delgada línea que separa a los partidarios del hardware abierto de los que abogan por una semiapertura de los productos derivados se cruza con estos inconvenientes. La licencia TAPR OHL acota un poco este problema, aunque han habido muchas controversias a la hora de publicar hardware nuevo.

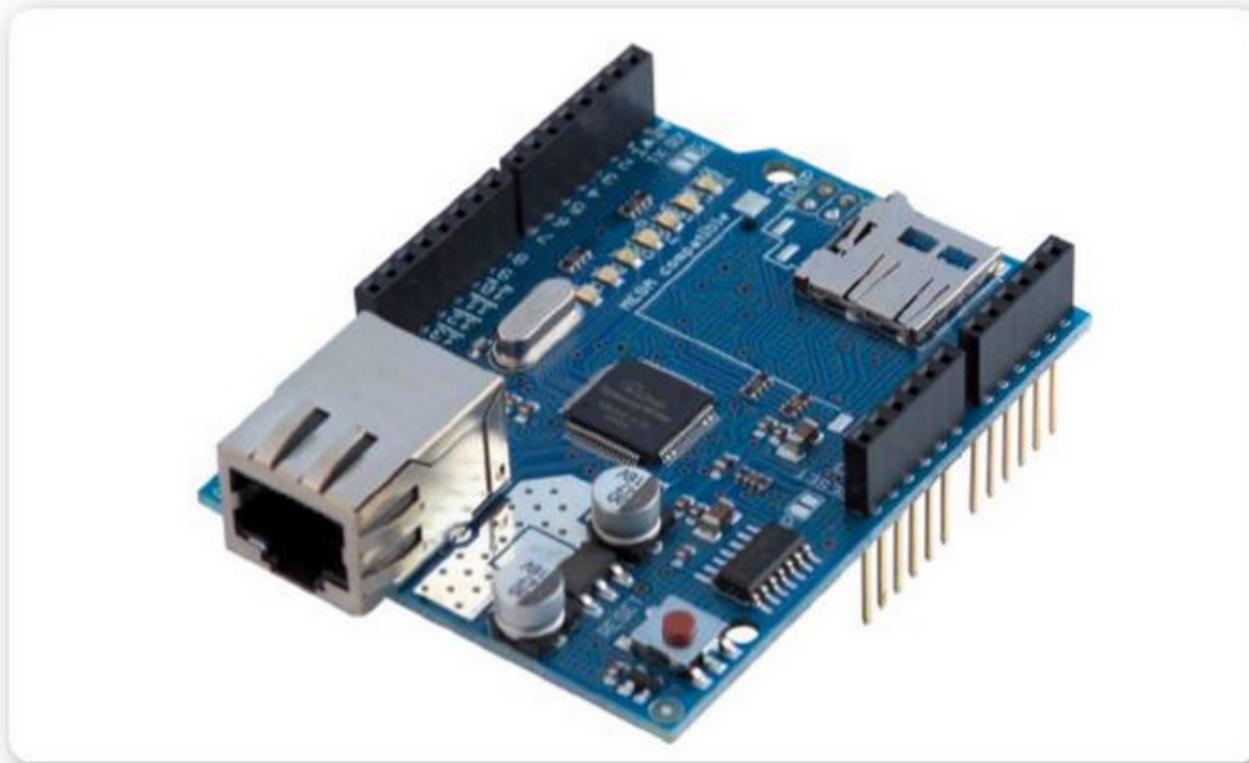


Figura 23. Arduino es un ejemplo de cómo preservar un proyecto de los plagios. En su placa PCB figura el nombre del proyecto y sus componentes.

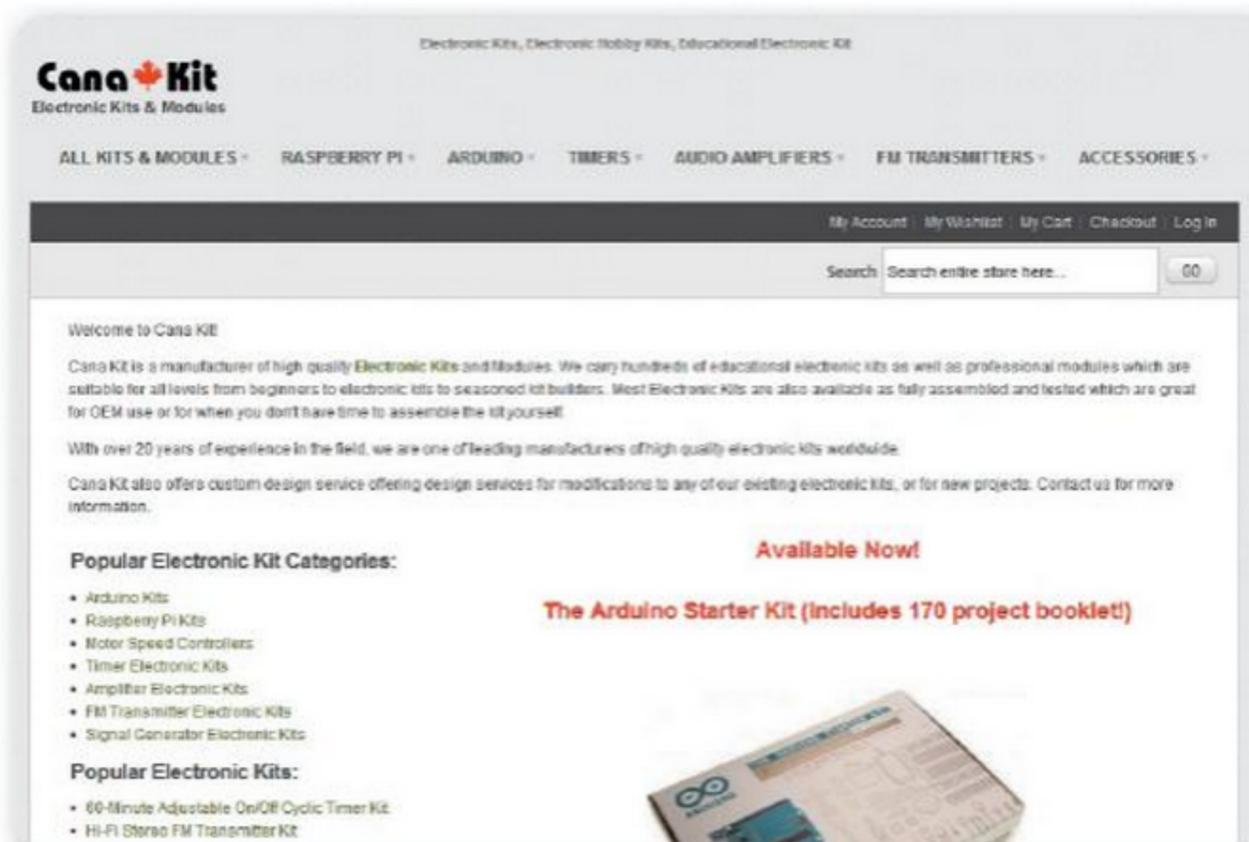


Figura 24. En www.canakits.com se comercializan componentes para Arduino. En la actualidad, el 40 % de su facturación corresponde al open hardware.

En el ámbito del hardware abierto, hay empresas que, al comercializar pequeños proyectos de electrónica, promocionan la utilización del concepto en la creación de nuevos proyectos más

avanzados o con otra funcionalidad. Entre estas, están Arduino, Michigan Tech OHW, canakit.com y muchas otras que perciben ganancias y, sin embargo, sus partes son de hardware abierto.

La mayoría de las licencias sin uso comercial (*non-commercial* en inglés) se restringen para el uso académico.

En lo referente al patentamiento, todos los componentes de las partes o proyectos open hardware deben estar libres de este tipo de restricciones. Esto es solo algo conceptual, ya que, en la mayoría de los casos, algunos componentes están bajo la protección de patentes. En otros casos, los inventores, si bien patentan sus logros, los ponen a disposición del público en general sin cobrar, lo que permite que su invención se propague con mayor fuerza debido al éxito comercial que esto implica. En el sitio **OpenElectronics** (www.open-electronics.org), podemos ver cuán complejo es el tema de las licencias y de su aplicación en proyectos. En el caso de las aplicaciones comerciales, podría ser un inconveniente que algunas piezas de hardware estén disponibles para su modificación o replicación, si son cruciales en un nuevo producto. Otro tanto puede suceder con el equipamiento con intencionalidad libre, ya que puede haber componentes que estén protegidos tanto por derecho de autor (en el caso del diseño o código fuente del firmware) como por patentes (en el caso de algún componente novedoso).

Por lo general, en cualquier proyecto actual nos encontramos con una mezcla de ambos mundos: por un lado, hay que ver los costos asociados al desarrollo de determinados componentes, por lo cual es de esperar que estén protegidos, y, por otro lado, algunos componentes son desarrollados por universidades o centros de investigación públicos, que ponen a disposición del público estos descubrimientos.

Además, tenemos la legalidad dentro de los ámbitos donde se trabaja; por ejemplo, lo que es ilegal en un estado o país, podría



ICE TUBE CLOCK KIT



Adafruit es una empresa creada a partir de componentes de open hardware. Algunos de sus productos más vendidos son los gadget para tecnófilos y las partes para otros productos de hardware abierto. El más llamativo de todos ellos es el **Ice Tube Clock Kit**, una simple pieza de resina de cristal con seis display de siete segmentos unidos a un reloj digital que vendió más de 100.000 unidades en pocos meses.

bien ser legal en otro. Como las patentes no son universales, sino que están enmarcadas en algunos países por acuerdos bilaterales, su alcance no va más allá.

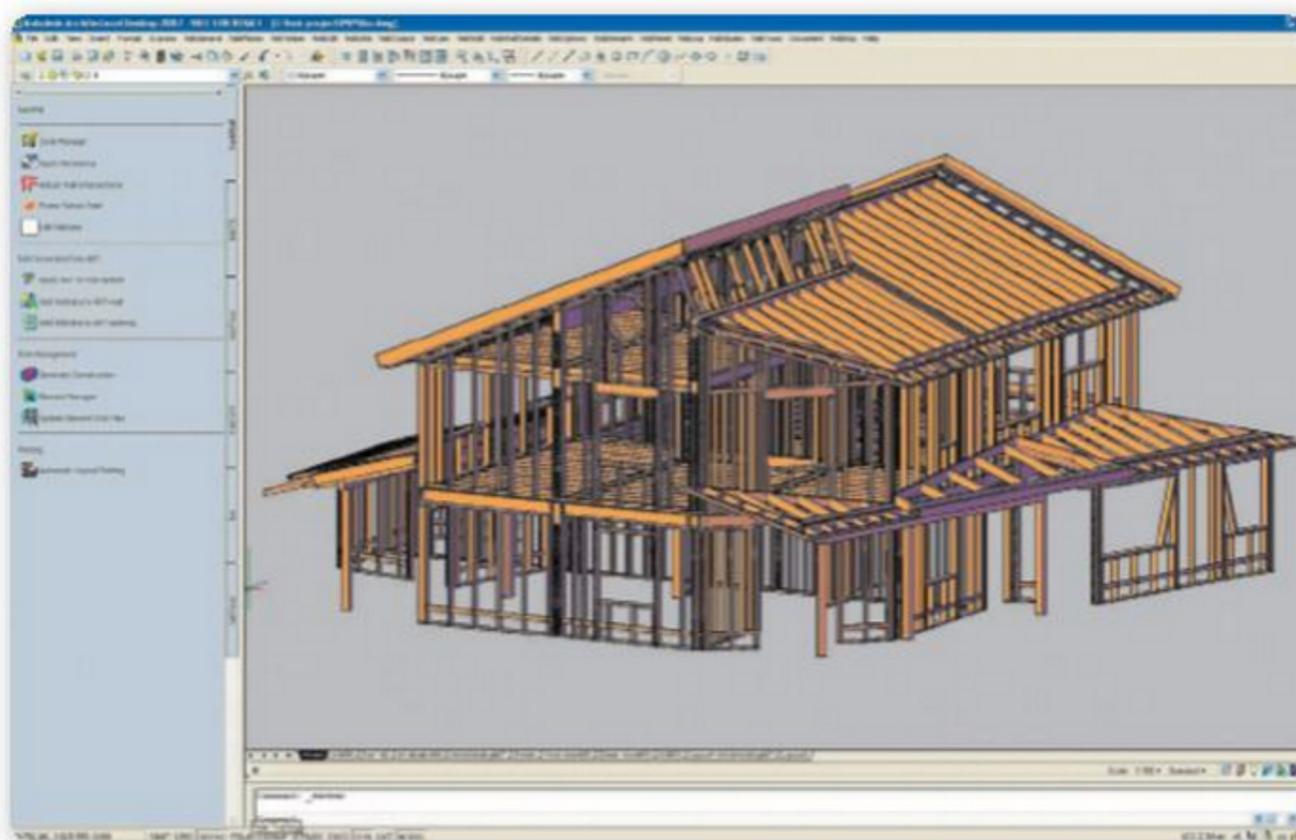


Figura 25. A los diseños CAD se les aplican las licencias de open source (código abierto) o similares, ya que lo que se quiere liberar es el derecho de autor.



Figura 26. Las licencias TAPR OHL permiten la comercialización de los productos de hardware y su diseño asociado, además de prevenir que las empresas patenten sus componentes.

Proyectos de hardware libre

Entendemos por **hardware libre** a todos aquellos dispositivos de hardware (dispositivos tangibles) cuyas especificaciones y diagramas esquemáticos son de acceso público, aunque no siempre sean gratuitos. La filosofía del software libre se aplica a la del hardware libre; entonces, ambas forman parte de la **cultura libre**, cada vez más extendida.

Desde el punto de vista histórico, el concepto de hardware libre se inició en la década de 1960 y, tanto en la década del 70 como en la del 90, se extendió a otros ámbitos de la tecnología. Hoy en día, se han creado licencias específicas de uso en hardware libre que intentan proteger algún aspecto de un dispositivo bajo esta filosofía.

Desde el punto de vista comercial, lo único realmente importante es mantener el diseño libre y compartirlo con diseñadores e interesados a lo largo y ancho del mundo, quienes aportan sus recursos técnicos, energía y creatividad de forma dinámica y participativa. Dada la naturaleza física del hardware, no se pueden aplicar directamente las cuatro libertades, ya mencionadas, del software libre.

En las páginas siguientes, haremos referencia a algunos de los proyectos de hardware libre más notables, que se benefician del trabajo colaborativo bajo un modelo de cooperar y compartir.

Elphel

La iniciativa **Elphel** consiste en una cámara open source y software libre diseñado por la empresa Elphel Inc. para aplicaciones científicas. En este sentido, se utiliza para capturar imágenes generadas por las iniciativas Google Street View y Google Books y por la NASA, en el vehículo Global Hawk UAV (vehículos aéreos o no tripulados). El laboratorio Moss Landing Marine utiliza cámaras Elphel en el proyecto de exploración antártica mediante robots submarinos (SCINI).

En cuanto a las herramientas que soportan este proyecto de hardware libre, para la serie de cámaras Elphel 353, en http://wiki.elphel.com/index.php?title=Elphel_353_series_quick_start_guide disponemos de una guía de inicio rápido, mientras que el software para las cámaras está disponible en <http://sourceforge.net/projects/elphel/>. Además, la herramienta que nos ayuda a definir el tipo de

lente necesario para una aplicación en particular, cuando ciertos parámetros pueden ser importantes, utilizando diferentes formatos de sensores, la encontramos en <http://blog.elphel.com/2012/12/sensorlens-tool>. En el sitio del fabricante, www3.elphel.com, hay información actualizada sobre esta iniciativa.

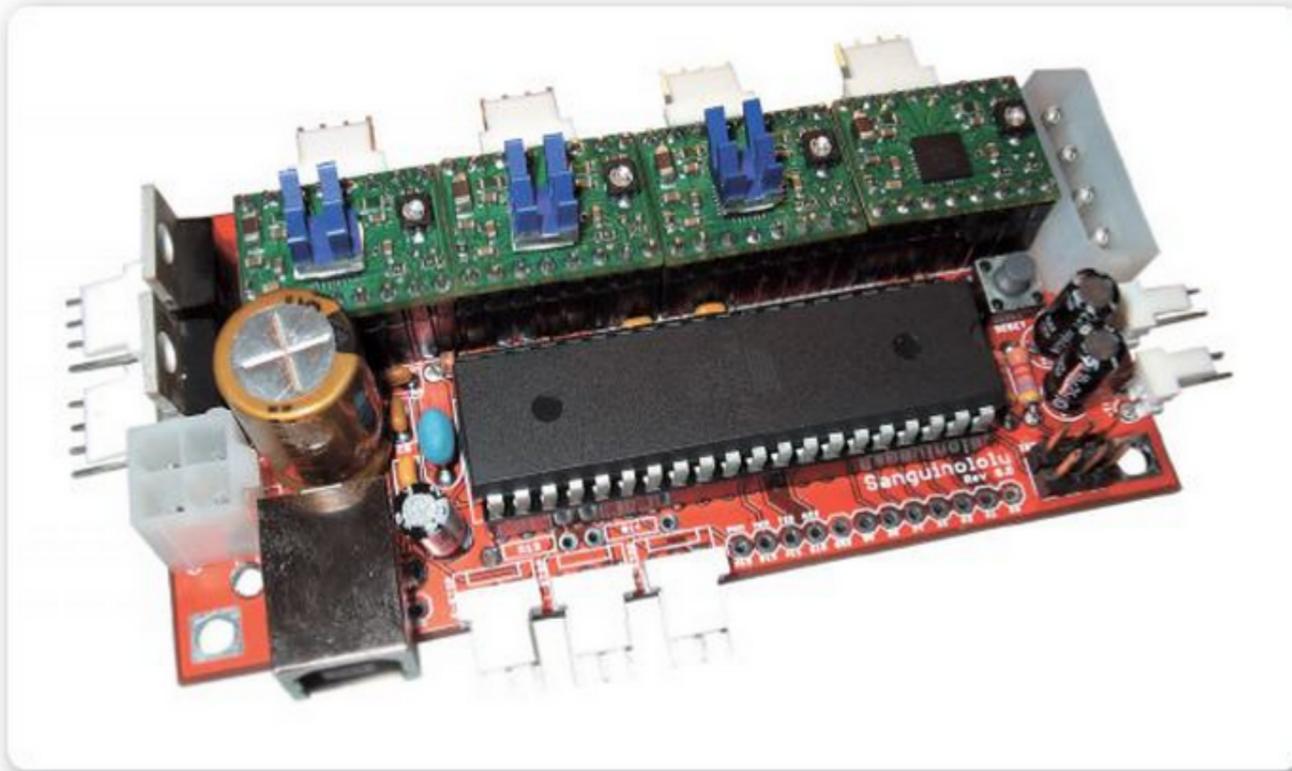


Figura 27. El hardware libre lleva la filosofía del software libre al mundo del hardware. Podemos encontrar proyectos de hardware libre sostenidos por comunidades de usuarios y empresas.



Figura 28. La cámara NC353L-369 de la empresa Elphel Inc. es open source y free software, diseñada para aplicaciones científicas.



Figura 29. Elphel Inc. nos facilita una aplicación útil para especificar el lente que necesitamos al emplear distintos tipos de sensores.

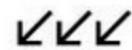
RepRap

RepRap, un proyecto iniciado durante el 2004 en Inglaterra, es la primera máquina de uso general de prototipo rápido con la habilidad de replicarse a sí misma bajo la forma de una impresora 3D libre, apta para imprimir objetos plásticos. El concepto de autoreplicación implica que una impresora RepRap 3D puede imprimir un kit de sí misma, además de otros objetos. Esta facilidad posibilitaría distribuir a bajo costo máquinas RepRap a personas y comunidades, permitiéndoles crear productos y objetos complejos sin la necesidad de disponer de herramientas industriales costosas. Los modelos de los objetos se generan mediante una computadora.

RepRap está disponible bajo una licencia del tipo **GNU GPL** (*General Public License* o **Licencia Pública General GNU**) y nos



IMPRESORA MAKERBOT 3D



Makerbot Industries fue constituida en 2009 con el apoyo del fundador de RepRap y se inició ofreciendo impresoras 3D bajo el formato de hardware libre y abierto. Consistía en un kit para construir impresoras utilizando herramientas sencillas, lo que generó una comunidad muy activa que las fue mejorando.

permite compartir (copiar), analizar, distribuir y mejorar los diseños de la impresora, así como el código fuente. En la actualidad, podemos considerar un éxito a este proyecto, ya que se ha cumplido su objetivo básico y resta resolver cómo incrementar el número de partes reproducibles y la calidad del tipo de objetos producidos. En su sitio web (en español), http://reprap.org/wiki/RepRap_en_español_-_RepRap_in_Spanish, encontramos información sobre el proyecto relacionada con el software y la construcción de la impresora en tres dimensiones.

REPRAP ES LA PRIMERA MÁQUINA DE PROTOTIPADO RÁPIDO QUE PUEDE AUTORREPLICARSE

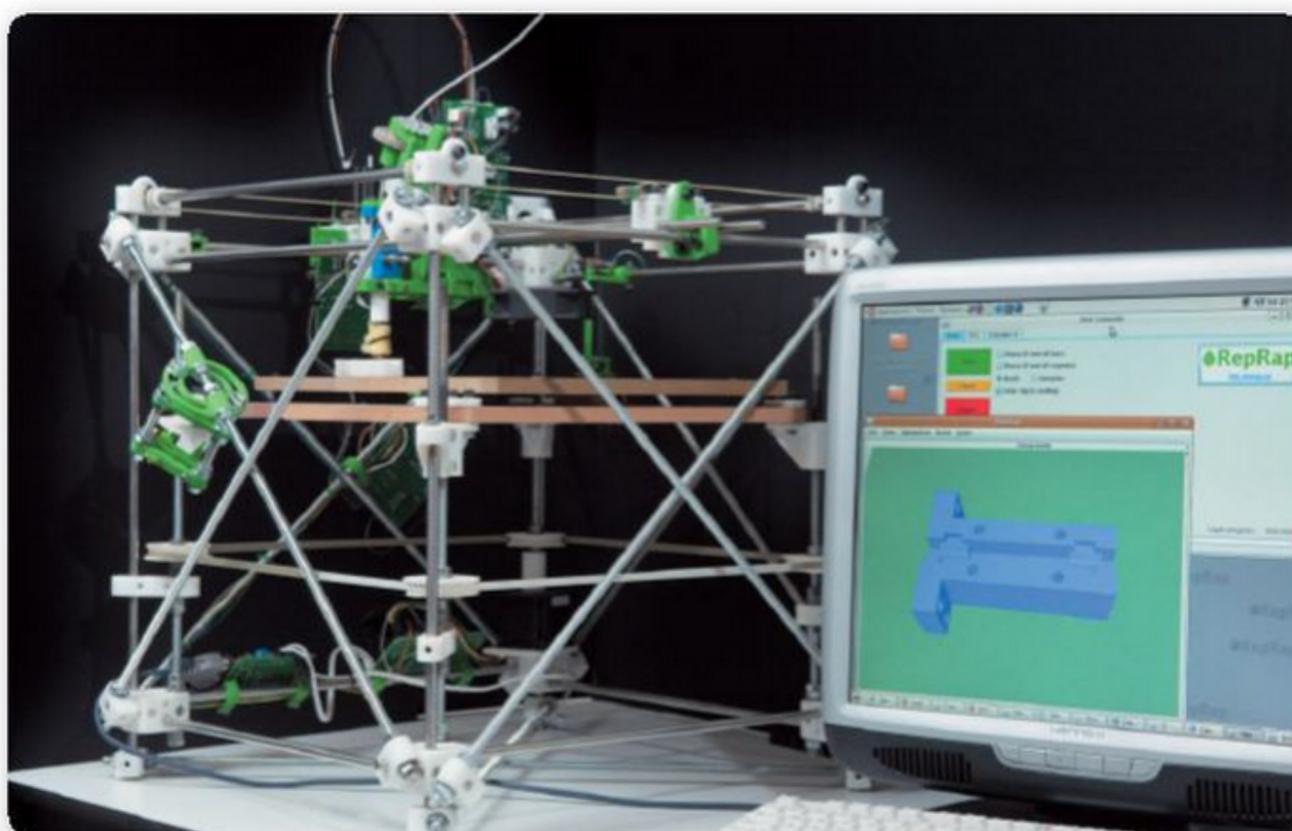


Figura 30. RepRap es la primera máquina de uso general de prototipado rápido que puede replicarse a sí misma bajo la forma de una impresora 3D.



THINGIVERSE

En el sitio www.thingiverse.com se ofrece un espacio en el cual podemos compartir nuestras creaciones de forma libre y colaborativa. Una de las últimas novedades de Makerbot, la impresora **3D Replicator 2**, mejora la anterior, aunque no es libre ya que su hardware y software son cerrados y propietarios.

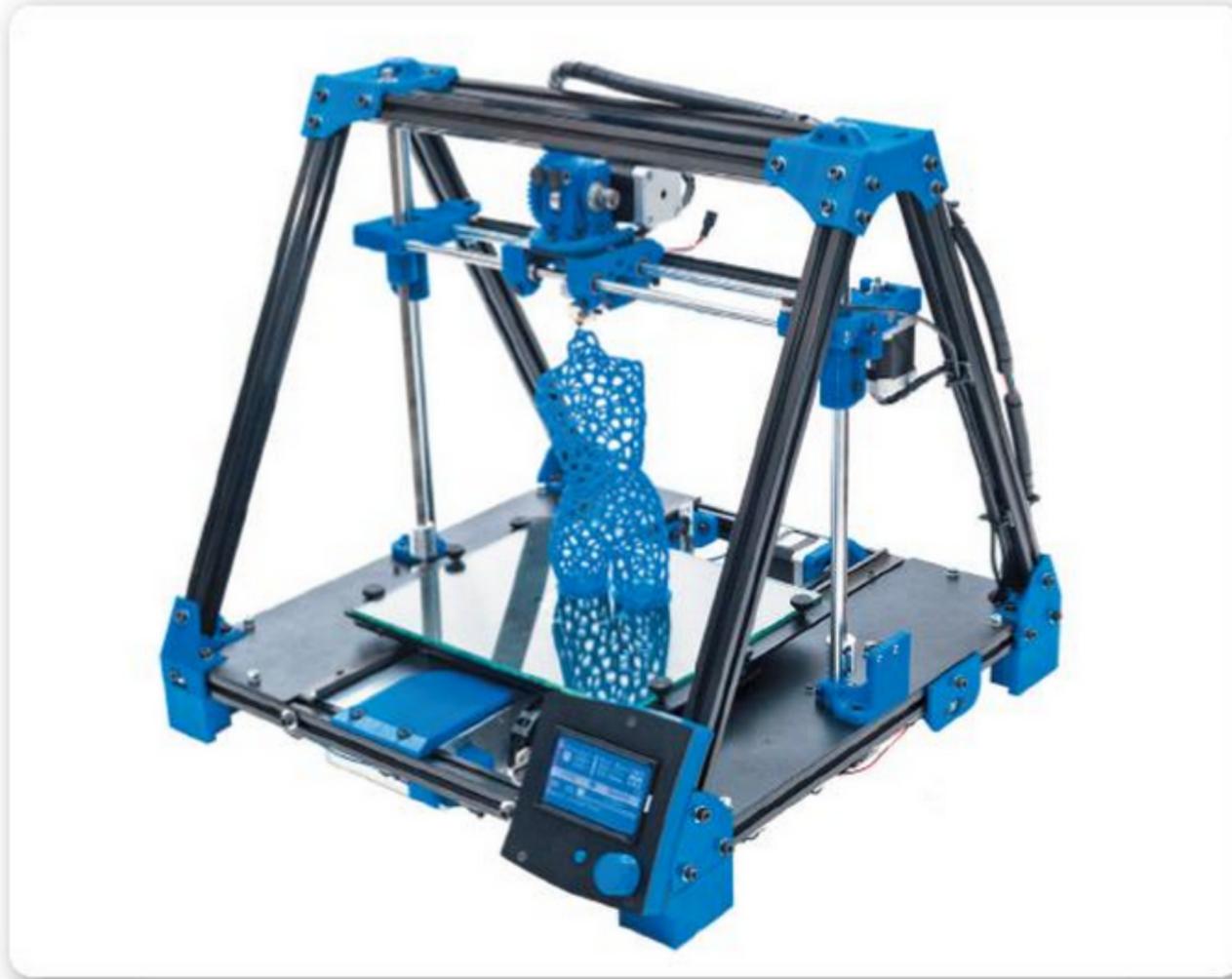


Figura 31. Dada su facilidad de autorreplicarse, podemos distribuir a bajo costo máquinas **RepRap** y crear objetos complejos.

OpenSPARC

OpenSPARC (en inglés, *Open Scalable Processor ARChitecture*) es otro proyecto de hardware libre, aunque esta vez orientado a una arquitectura con un conjunto de instrucciones reducidas. El esquema original fue diseñado por Sun Microsystems, que otorgó la licencia a fabricantes como Cypress Semiconductor, Fujitsu y Texas Instruments, entre otros. Se trata de la primera arquitectura RISC abierta, basada en una arquitectura RISC diseñada en la Universidad de California en Berkeley, por lo que las especificaciones de diseño están disponibles para que otros fabricantes de microprocesadores puedan desarrollar sus propios dispositivos.

En marzo de 2006, SUN Microsystems presentó el primer microprocesador open hardware, el **UltraSPARC T1**, seguido en 2008 por el microprocesador **OpenSPARC T2**, el primero de 64 bits. También es el primer microprocesador open hardware con capacidad multihilo (en inglés, *chip multithreaded*, CMT). Tanto estos diseños,

como el código fuente del diseño del procesador, las herramientas de simulación, las suites verificadoras del diseño, el código fuente Hypervisor y otras herramientas de ayuda están disponibles de manera gratuita. A partir de 2009, Sun transfirió la propiedad de la especificación SPARC a SPARC International, que continúa licenciando la tecnología y gestionando el proyecto hoy en día.

Encontramos valiosa información sobre este proyecto en www.sparc.org, el sitio de SPARC International, una organización independiente sin fines de lucro dedicada a supervisar y guiar la evolución de SPARC y otros estándares tecnológicos abiertos. Las principales ventajas de la arquitectura SPARC se relacionan con la escalabilidad del precio y las funciones implementadas; además, provee un rendimiento entre los más poderosos, su certificación SPARC facilita la compatibilidad con generaciones anteriores y, por último, cuenta con una gran cantidad de aplicaciones disponibles para los usuarios a lo largo y ancho del mundo.

Estas ventajas se sostienen en el tiempo, ya que las especificaciones SPARC están disponibles para ser licenciadas a cualquier persona o empresa. La competencia se basa en la implementación y acceso igualitario al núcleo de la tecnología. El set de instrucciones SPARC es la base para el estándar abierto IEEE 1754-1994, y el control sobre la arquitectura SPARC está depositado en las manos de una organización no gubernamental independiente cuya membresía está abierta para todos los que tengan interés en incorporarse. De esta manera, SPARC International se asegura que las ventajas de su arquitectura sean completamente accesibles para la totalidad de la industria. También se encuentra disponible una plataforma en base a FPGA, OpenSPARC FPGA, que nos proporciona una serie de ventajas en la investigación de microprocesadores SPARC.



VENTAJAS DEL HARDWARE LIBRE



Contribuye a la independencia tecnológica de un país, favorece la calidad del hardware y los estándares abiertos, facilita la reutilización y adaptación de los diseños, reduce costos y tiempos de diseño. Estas ventajas se fortalecen certificando el hardware con **Open Hardware Specification Program**, lo que nos asegura la compatibilidad de diversos componentes informáticos con el SO Linux.



Figura 32. OpenBook es un proyecto de Via Technologies con un diseño open source para netbooks con licencia Creative Commons.

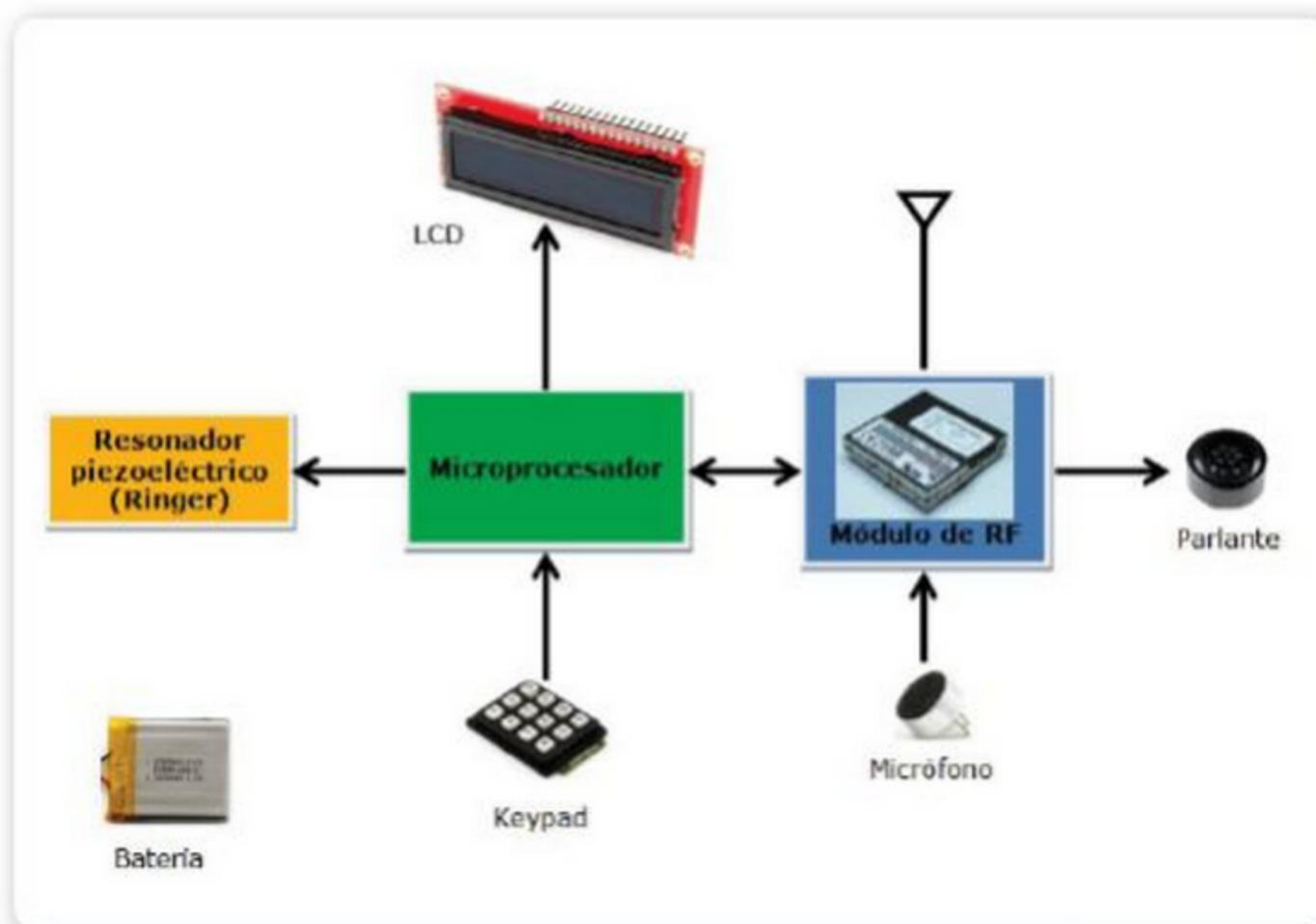


Figura 33. Arquitectura del proyecto de hardware libre y teléfono móvil **TuxPhone** sobre una placa OTS y SO Linux.

PAP: CONSTRUCCIÓN DE UN MODELO DE OPEN HARDWARE

01 La impresora 3D de Cooking Hacks es ideal para quienes se inician en el mundo de la impresión 3D, y la obtenemos en forma de un kit fácil de ensamblar.



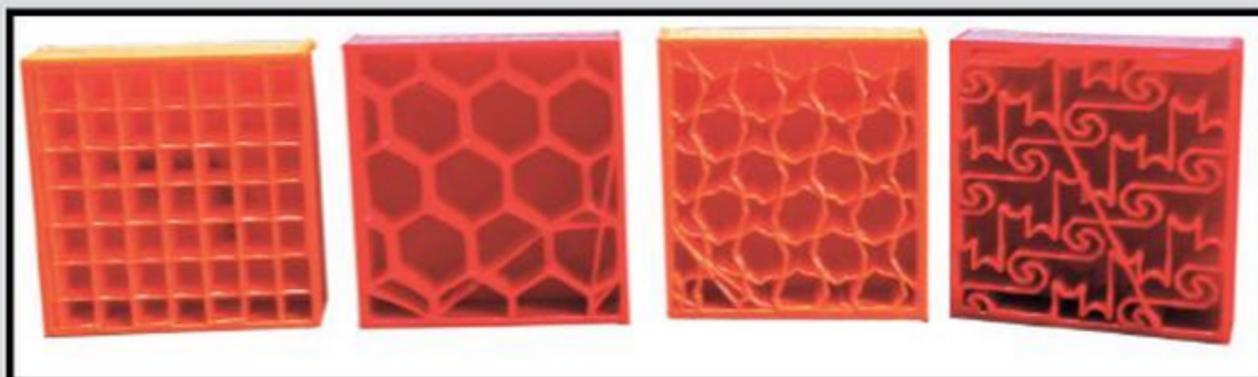
02 En www.cooking-hacks.com, encuentre la guía de ensamble de la impresora 3D, el software necesario y los archivos CAD para imprimir.



03 Además del IDE Arduino, con Marlin, Arduino interpreta los comandos desde la PC y Cura traslada el diseño 3D al lenguaje 3D de la impresora, GCODE.



04 Con esta impresora 3D, el software indicado y la guía de ensamble y uso, rápidamente estará en condiciones de imprimir objetos 3D como los de la imagen.



OpenBook

OpenBook es un proyecto de **Via Technologies**, www.via.com.tw, con un diseño open source para netbooks que funcionan bajo licencia Creative Commons. Una de sus principales aplicaciones es facilitar la creación de dispositivos genéricos a productores **OEM** (*Original Equipment Manufacturer*) que utilizan marcas propias para identificar sus productos.

Como iniciativa hardware libre, VIA OpenBook pretende facilitar el camino hacia la próxima generación de notebooks ultraportátiles basadas en la plataforma de ultramovilidad VIA. Introduce una gran cantidad de innovaciones y características para los usuarios: una carcasa delgada con esquinas curvas, un peso menor a 1 kg, acceso libre a los archivos CAD para modificar el diseño de la carcasa, aspectos que abren a los diseñadores alternativas para personalizar el producto.

Una mejora muy importante es en el campo de la conectividad extendida, ya que no solo nos proporciona una conexión Wi-Fi, sino que también incorpora módulos WiMAX y opciones HSDPA o EV-DO/W-CDMA para conectividad celular en redes 3G, de modo de proporcionarnos un acceso instantáneo a internet de banda ancha y, virtualmente, desde cualquier lugar. La notebook VIA OpenBook aporta una pantalla de 8.9 que soporta una resolución de 1024x600 mediante VIA Chrome9 HC3 y gráficos 3D DirectX 9.0 para dar al usuario una gran experiencia multimedia.

Estas facilidades se combinan con una aceleración de video avanzada para los formatos de video MPEG-2, MPEG-4, WMV9 y VCI, además de capacidades de procesamiento de video HD mediante VMR. La placa madre se construye alrededor del procesador VIA C7-M ULV (con velocidades desde 1.0 GHz a 1.6 GHz) y del nuevo Chipset VIA VX800 Digital Media. En cuanto al sistema operativo, VIA OpenBook soporta tanto Windows XP como Vista y Linux, además de conectividad Bluetooth, Wi-Fi, AGPS y Ethernet.

OPENSPARC ES
OTRO PROYECTO DE
HARDWARE LIBRE
CON INSTRUCCIONES
REDUCIDAS



TuxPhone

TuxPhone es un proyecto de hardware libre, específicamente, un teléfono móvil que soporta software libre y open source bajo

el sistema operativo Linux. El concepto original es que cualquier persona, con una herramienta no más compleja que un soldador, sea capaz de construir un teléfono móvil, utilizando software libre y open source, SO Linux y el soporte de una comunidad de desarrolladores de ringtones, música y fondos de pantalla.

TUXPHONE ES UN
TELÉFONO MÓVIL
BAJO HARDWARE
LIBRE Y SISTEMA
OPERATIVO LINUX



Está basado en una placa OTS con Linux que nos proporciona una serie de ventajas: open source, gratuito, soporte excelente para diferentes pilas de protocolos, una comunidad de desarrolladores activa y un amplio soporte para el desarrollo de aplicaciones. Algunas desventajas que plantea esta solución se relacionan con una baja eficiencia energética, un SO que no funciona en tiempo real y el requerimiento de una amplia cantidad de recursos. Como microprocesador, emplea un pequeño procesador Intel XScale

PXA255 400 MHz basado en SBC Linux.

En una configuración posible para este teléfono móvil, el módulo de comunicaciones requiere que sea capaz de operar en RF y proporcionarnos una interfaz para comandos AT de modo de comunicar el módulo con el microprocesador. La selección recae en el módulo GM862 de la empresa Telix, que funciona en las bandas de 900 MHz, 1800 MHz y 1900 MHz a una tasa de datos de 57.6 kbits/s y, además, soporta tanto el modo normal como manos libres. El display utilizado es SEIKO Color STN 1.5 (RNH942209R1A), específico para teléfonos celulares.

Se requieren tres componentes de audio: un parlante de 16 Ohm de impedancia, 30 mW de potencia de salida máxima y 4 cm de longitud como el Panasonic EAS-4D02C0, un micrófono balanceado



COMUNIDADES DE HARDWARE LIBRE



En internet podemos encontrar algunos proyectos como **OpenCores** (<http://opencores.org>) y **Open-Hardware** (www.openhardware.de), donde se recopilan esquemas y desarrollos de componentes de hardware enviados por profesionales independientes, que colaboran con el mismo espíritu de la comunidad del software libre.

y aislado de la carga con sensibilidad de -45 dB y un resonador piezoeléctrico (ringer). Este conjunto se complementa con un teclado 3x4. Respecto de la alimentación eléctrica, se utilizan baterías livianas, seguras, recargables (varios cientos de veces) y disponibles fácilmente. Por su gran capacidad de corriente, se deben considerar baterías de NiMH o de litio.

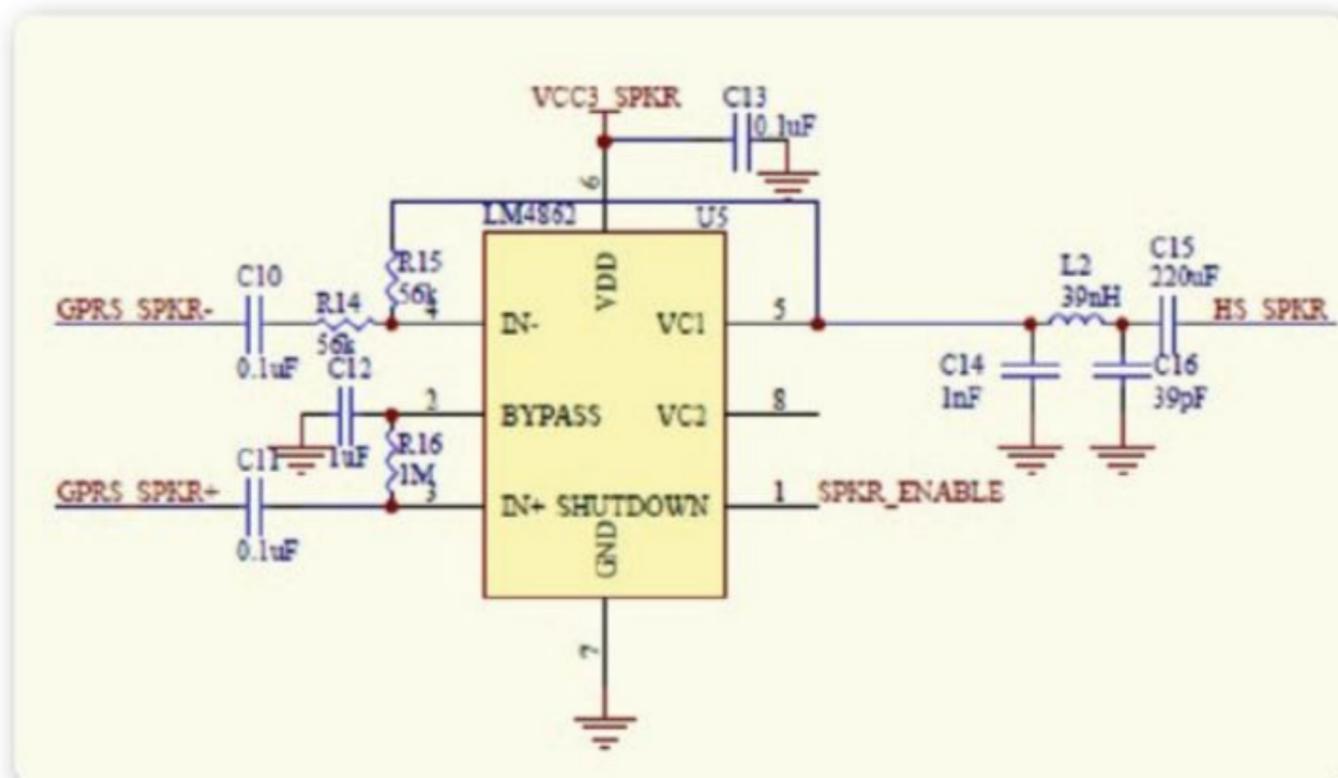


Figura 34. Sección de radiofrecuencia del teléfono móvil **TuxPhone** con el módulo GM862 de la empresa Telix tribanda y con comandos AT Hayes.

Oscar

Oscar es un proyecto de automóvil ecológico bajo código abierto al que la comunidad de ingenieros y diseñadores que lo impulsa define muy bien como el intento de diseñar un automóvil usando los principios de los productos físicos de código abierto (*open source hardware*). El modelo de código abierto Oscar pretende sencillez entendida como número de piezas mínimo, chasis y carrocería resistentes y seguros. La forma denota la función, internacional, modular y con bajo costo de mantenimiento.

OSCAR ES UN
PROYECTO DE
AUTOMÓVIL
ECOLÓGICO BAJO
CÓDIGO ABIERTO





Figura 35. OScar es un automóvil diseñado siguiendo los principios de los productos físicos de código abierto.

Pese a que el proyecto fue acotado en un principio al diseño y producción de un modelo de automóvil de código abierto, cuyas especificaciones pudieran ser replicadas por cualquiera, existen ahora grupos de trabajo paralelos para diseñar otros vehículos: un autobús, una camioneta y una motocicleta, también de código abierto.

OScar se inició como un proyecto de bajo presupuesto en 1999 y, a fines de 2005, se lanzó la versión 2.0. Dada su naturaleza de proyecto open hardware, cualquier persona interesada puede aportar voluntariamente sus conocimientos al proyecto de manera de contribuir a la filosofía original de OScar: desarrollo de un producto basado en la colaboración, el intercambio de conocimiento y las experiencias de cada integrante del grupo de trabajo.

Por ahora, OScar es un diseño realizado en computadora mediante software CAD de código abierto, mientras espera recursos para construir los primeros prototipos tridimensionales en base a las especificaciones actuales: cuatro puertas, peso máximo de 1.000 kilogramos, 4 metros de longitud, 1.75 metros de ancho y 1.55 metros

de alto con una distancia entre ejes de 2.50 metros. Respecto del consumo, uno de los aspectos más importantes para considerar en el diseño de un automóvil, OScar se propone utilizar un motor eléctrico con una batería con suficiente capacidad para recorridos interurbanos a una velocidad máxima de 145 km/h.

Open Graphics Project

Open Graphics Project (OGP), iniciativa conocida como **Proyecto de Gráficos Abierto**, se orienta al diseño de hardware libre de arquitectura abierta y estándares para placas gráficas utilizando sistemas operativos de software libre. Entendemos por Open Graphics Project (OGP) al grupo de personas que desarrollan el OGA, su documentación escrita y sus productos, mientras que Open Graphics Architecture (OGA) es la marca registrada para las arquitecturas gráficas abiertas especificadas por el Open Graphics Project.

Inicialmente, el proyecto desarrollará un prototipo de tarjeta gráfica PCI mediante FPGA para ir mejorando su diseño en versiones AGP y PCI Express hasta alcanzar un desarrollo de tarjeta gráfica completa y competitiva. El usuario tendrá disponibles tanto las especificaciones completas como los controladores de dispositivos de código fuente abierto.

El código abierto de los controladores del dispositivo y el BIOS serán liberados bajo la licencia MIT y la licencia BSD. El RTL (en Verilog) usado para el FPGA y el RTL para el ASIC están planeados para ser liberados bajo la Licencia Pública General de GNU (GPL). La placa tendrá 256 MB de memoria RAM DDR, enfriado pasivamente y siguiendo los estándares VESA: DDC, EDID, DPMS y VBE. Además, se propone una salida de TV (TV-out).



DESVENTAJAS DEL HARDWARE LIBRE

De acuerdo con la Fundación Software Libre (**Free Software Foundation**), las libertades del software libre no se aplican al hardware: los diseños específicos y únicos dificultan su reproducción, debemos probar los componentes antes de construir el dispositivo, no disponemos de los CI en todos los países ni la infraestructura de diseño, simulación, producción e implementación.



Figura 35. Oscar es un automóvil diseñado siguiendo los principios de los productos físicos de código abierto.

Pese a que el proyecto fue acotado en un principio al diseño y producción de un modelo de automóvil de código abierto, cuyas especificaciones pudieran ser replicadas por cualquiera, existen ahora grupos de trabajo paralelos para diseñar otros vehículos: un autobús, una camioneta y una motocicleta, también de código abierto.

Oscar se inició como un proyecto de bajo presupuesto en 1999 y, a fines de 2005, se lanzó la versión 2.0. Dada su naturaleza de proyecto open hardware, cualquier persona interesada puede aportar voluntariamente sus conocimientos al proyecto de manera de contribuir a la filosofía original de Oscar: desarrollo de un producto basado en la colaboración, el intercambio de conocimiento y las experiencias de cada integrante del grupo de trabajo.

Por ahora, Oscar es un diseño realizado en computadora mediante software CAD de código abierto, mientras espera recursos para construir los primeros prototipos tridimensionales en base a las especificaciones actuales: cuatro puertas, peso máximo de 1.000 kilogramos, 4 metros de longitud, 1.75 metros de ancho y 1.55 metros

de alto con una distancia entre ejes de 2.50 metros. Respecto del consumo, uno de los aspectos más importantes para considerar en el diseño de un automóvil, OScar se propone utilizar un motor eléctrico con una batería con suficiente capacidad para recorridos interurbanos a una velocidad máxima de 145 km/h.

Open Graphics Project

Open Graphics Project (OGP), iniciativa conocida como **Proyecto de Gráficos Abierto**, se orienta al diseño de hardware libre de arquitectura abierta y estándares para placas gráficas utilizando sistemas operativos de software libre. Entendemos por Open Graphics Project (OGP) al grupo de personas que desarrollan el OGA, su documentación escrita y sus productos, mientras que Open Graphics Architecture (OGA) es la marca registrada para las arquitecturas gráficas abiertas especificadas por el Open Graphics Project.

Inicialmente, el proyecto desarrollará un prototipo de tarjeta gráfica PCI mediante FPGA para ir mejorando su diseño en versiones AGP y PCI Express hasta alcanzar un desarrollo de tarjeta gráfica completa y competitiva. El usuario tendrá disponibles tanto las especificaciones completas como los controladores de dispositivos de código fuente abierto.

El código abierto de los controladores del dispositivo y el BIOS serán liberados bajo la licencia MIT y la licencia BSD. El RTL (en Verilog) usado para el FPGA y el RTL para el ASIC están planeados para ser liberados bajo la Licencia Pública General de GNU (GPL). La placa tendrá 256 MB de memoria RAM DDR, enfriado pasivamente y siguiendo los estándares VESA: DDC, EDID, DPMS y VBE. Además, se propone una salida de TV (TV-out).



DESVENTAJAS DEL HARDWARE LIBRE

De acuerdo con la Fundación Software Libre (**Free Software Foundation**), las libertades del software libre no se aplican al hardware: los diseños específicos y únicos dificultan su reproducción, debemos probar los componentes antes de construir el dispositivo, no disponemos de los CI en todos los países ni la infraestructura de diseño, simulación, producción e implementación.

Arduino

Otra de las iniciativas nacidas bajo el concepto de hardware libre es el **proyecto Arduino**, creado por Massimo Banzi y David Cuartielles en Italia. Esta plataforma consiste en una placa de circuitos electrónicos con diversas entradas y salidas, que permite el desarrollo de una enorme cantidad de proyectos para el control de objetos interactivos autónomos, al mismo tiempo que sirve de plataforma para estudiantes como iniciación al mundo de la electrónica. En www.arduino.cc podemos obtener información respecto de los diferentes modelos de placas Arduino, su configuración y sus aplicaciones.

Un proyecto de hardware abierto basado en Arduino es **ArduSat** (en inglés, *Arduino satellite*), un satélite lanzado en agosto de 2013 a una altura de 300 km. Se ha constituido en la primera plataforma abierta disponible para el público en general con el objeto de diseñar y ejecutar sus propias aplicaciones espaciales, juegos y experimentos, controlar las cámaras satelitales para tomar imágenes bajo demanda y enviar mensajes personalizados de regreso a la Tierra. El satélite tiene una vida útil de dos años y, durante ese tiempo, cualquier persona en la tierra podrá comunicarse con él mediante los servidores del fabricante para hacer experimentos durante una semana.

En <http://tinyurl.com/ArduSatAppIdeas>, encontramos diferentes ideas para utilizar la plataforma ArduSat. La forma en que controlamos el satélite es generando nuestro propio código y subiéndolo a la página web de los desarrolladores para probarlo durante una semana en el satélite, previo pago de un monto determinado de dinero por el uso. Encontramos detalles adicionales en www.nanosatisfi.com/store y en el sitio www.kickstarter.com/projects/575960623/ardusat-your-arduino-experiment-in-space obtenemos una descripción del proyecto ArduSat.

Nano Satélite

Otro proyecto interesante, también de hardware abierto, es el primer nano satélite argentino, el **CubeBug-1**, más conocido como **Capitán Beto**, un dispositivo de 2 kg de peso, concebido, diseñado y fabricado tanto bajo el concepto de software como de hardware abierto (open software y open hardware). Estos estarán disponibles

para aficionados, universidades e institutos de investigación y la comunidad internacional.

Entre las múltiples aplicaciones de los nano satélites, tenemos observaciones científicas de la Tierra (clima y atmósfera) y las estrellas, ensayo de dispositivos tecnológicos como sensores, investigación biológica, nuevos materiales y funciones educativas. Finalizada la vida del satélite, quedará a disposición de la comunidad de aficionados.

El Capitán Beto dispone de 1 W de potencia de salida en la frecuencia de 437.438 MHz, modulación AFSK (FM) y, dependiendo del modo del satélite, la señal de las balizas se transmite cada 10 a 30 segundos. La información de telemetría del satélite está disponible en **<http://1.cubebug.org/coms/telemetry>** .

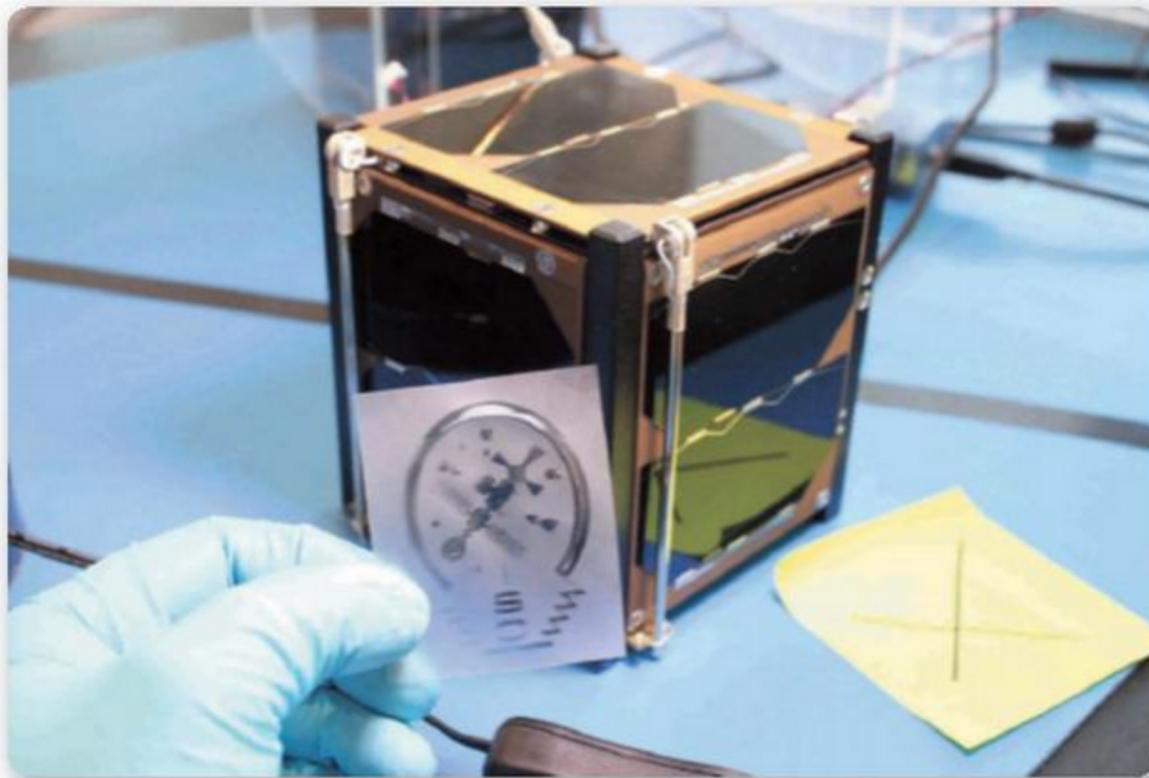


Figura 36. ArduSat es un proyecto que nos permite controlar desde la Tierra un satélite por medio de la página web de los desarrolladores.



RESUMEN

En este capítulo, desarrollamos una analogía entre los circuitos y el código fuente, para que puedan quedar a disposición de cualquier persona. Vimos también que la problemática, en este caso, está enfocada en la dificultad de copiar elementos físicos de la manera en que se copia un software. Por último, presentamos algunos ejemplos de las plataformas de hardware libre más utilizadas en la actualidad, para que el lector pueda continuar explorando dichas tecnologías en caso de que le sea de interés.

Actividades

TEST DE AUTOEVALUACIÓN

- 1 Explique la diferencia entre contenido libre y contenido de código abierto.
- 2 Desarrolle las bases del proyecto GNU.
- 3 ¿Cuáles son las bases del software libre con respecto a las libertades que un usuario tiene?
- 4 ¿Cuáles son las libertades esenciales que se suelen identificar dentro del movimiento de software libre?
- 5 ¿Cuáles son las características y particularidades de la placa Arduino?
- 6 ¿Cuáles son los principales problemas y desafíos que deben sortear los seguidores del hardware libre?
- 7 ¿Para qué se utilizan los licenciamientos? Mencione algunos ejemplos.
- 8 ¿Qué se entiende por proyecto de hardware libre?
- 9 ¿Cuáles son algunos de los proyectos de hardware libre más notables?
- 10 Enumere las situaciones problemáticas con las que debe lidiar, en la actualidad, el hardware abierto.



PROFESOR EN LÍNEA



Si tiene alguna consulta técnica relacionada con el contenido, puede contactarse con nuestros expertos: profesor@redusers.com



Plataforma Arduino

Dada su facilidad de uso, su gran comunidad de usuarios, su flexibilidad y su eficiencia, Arduino ostenta un lugar privilegiado entre las plataformas de hardware abierto. La idea de Arduino es segmentar los circuitos al máximo, para que puedan ser ensamblados y combinados con el fin de crear todo tipo de dispositivos que cumplan funciones técnicas sin requerir, siquiera, de un profundo conocimiento de su funcionamiento interno.

▼ Historia de Arduino.....62	Plataforma Arduino 73
Origen..... 62	Lenguajes C y C++..... 74
Funcionamiento..... 63	Bibliotecas 78
Entorno..... 65	Entorno de desarrollo 83
Colaboradores..... 65	▼ Physical Etoys.....88
▼ Evoluciones, versiones	▼ Periféricos y referencias.....92
y periféricos66	Referencias y recursos..... 96
Librerías..... 68	▼ Resumen.....97
▼ Programación de Arduino70	▼ Actividades.....98
Lenguajes de bajo nivel 71	
Lenguajes de alto nivel 72	
Lenguajes de medio nivel 73	



Historia de Arduino

Según sus autores, **Arduino** es una plataforma de creación de prototipos electrónicos de código abierto basada en la flexibilidad: hardware y software fáciles de usar. Está dirigida a artistas, diseñadores, aficionados y a cualquier persona interesada en la creación de objetos o entornos interactivos.

ARDUINO ES LA
PLATAFORMA DE
HARDWARE LIBRE
MÁS POPULAR DE
LOS ÚLTIMOS AÑOS

Se presenta como una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de electrónica en proyectos multidisciplinarios.

Para nosotros, es una placa con un microcontrolador que tiene entradas y salidas, tanto digitales como analógicas, que podemos programar para que realicen cualquier actividad que queramos. Además, es una plataforma de hardware libre (open source), por lo que podemos

disponer abiertamente de sus planos, comprar los componentes y fabricar nuestra propia Arduino sin infringir ninguna ley.

Es un dispositivo en el que es posible interactuar con una gran cantidad de objetos, desde simples LEDs, interruptores y sensores hasta objetos más complejos como visores LCD, GPS, GSM, impresoras, lectores de proximidad, entre otros tantos, de una manera fácil y sencilla. Básicamente, le decimos a la placa qué es lo que tiene que hacer, en qué tiempo y, si quisiéramos, bajo qué parámetros.



Figura 1. Este es el logo de Arduino, una marca registrada por el equipo y que representa el hardware open source.

Origen

Según cuenta su historia, Arduino nació por el año 2005 como un proyecto para estudiantes en el Instituto de Diseño Interactivo Ivrea de Italia (IDII), donde los alumnos experimentaban con distintos

tipos de microcontroladores. La idea era crear una herramienta moderna, simple, barata y sencilla de usar.

Fue así como empezaron a desarrollarla bajo la licencia open source, para que todo el mundo pudiese ayudar y contribuir, y, además, porque la escuela no pasaba por sus mejores años y se murmuraba que estaba por cerrar; de esta manera, el proyecto sobreviviría al cierre y no quedaría olvidado en cajones de armario.

Las primeras Arduino se conectaban mediante el puerto RS232 (el puerto COM de la PC), que rápidamente fue reemplazado por un puerto USB.

El hecho de ser open source ha permitido que mucha gente colabore con el proyecto y que en muy poco tiempo su uso explote por todo el mundo. Arduino es una representación física de todas las cosas geniales del software open source, que ahora empiezan a llegar al hardware.



LA IDEA ORIGINAL
FUE LA DE HACER
UNA PLATAFORMA
SIMPLE, BARATA Y
SENCILLA DE USAR



Funcionamiento

Existen varios modelos de Arduino, pero básicamente todos funcionan de la misma manera. Se diferencian por su tamaño físico, la capacidad de almacenamiento y la cantidad de entradas y salidas que poseen. En la actualidad, encontramos desde las más pequeñas, del tamaño de una moneda, hasta las más grandes, del tamaño de una billetera, representadas por la Arduino Mega (con 54 pines digitales y 16 analógicos) y la Arduino Nano (con 14 digitales y 8 analógicos).

Las más grandes cuentan con mayor número de conectores.

En principio, debemos conocer algo (muy poco) de programación para decirle a la placa qué es lo que queremos que haga. Para eso, tenemos que bajar de su sitio (<http://arduino.cc>) el software de desarrollo que nos permitirá codificar en el lenguaje del microcontrolador las órdenes que deberá llevar a cabo.

Actualmente, la última versión estable es la Arduino 1.0.5, tanto para 32 como para 64 bits, que es multiplataforma (se puede usar en Windows, Mac y Linux) y open source.

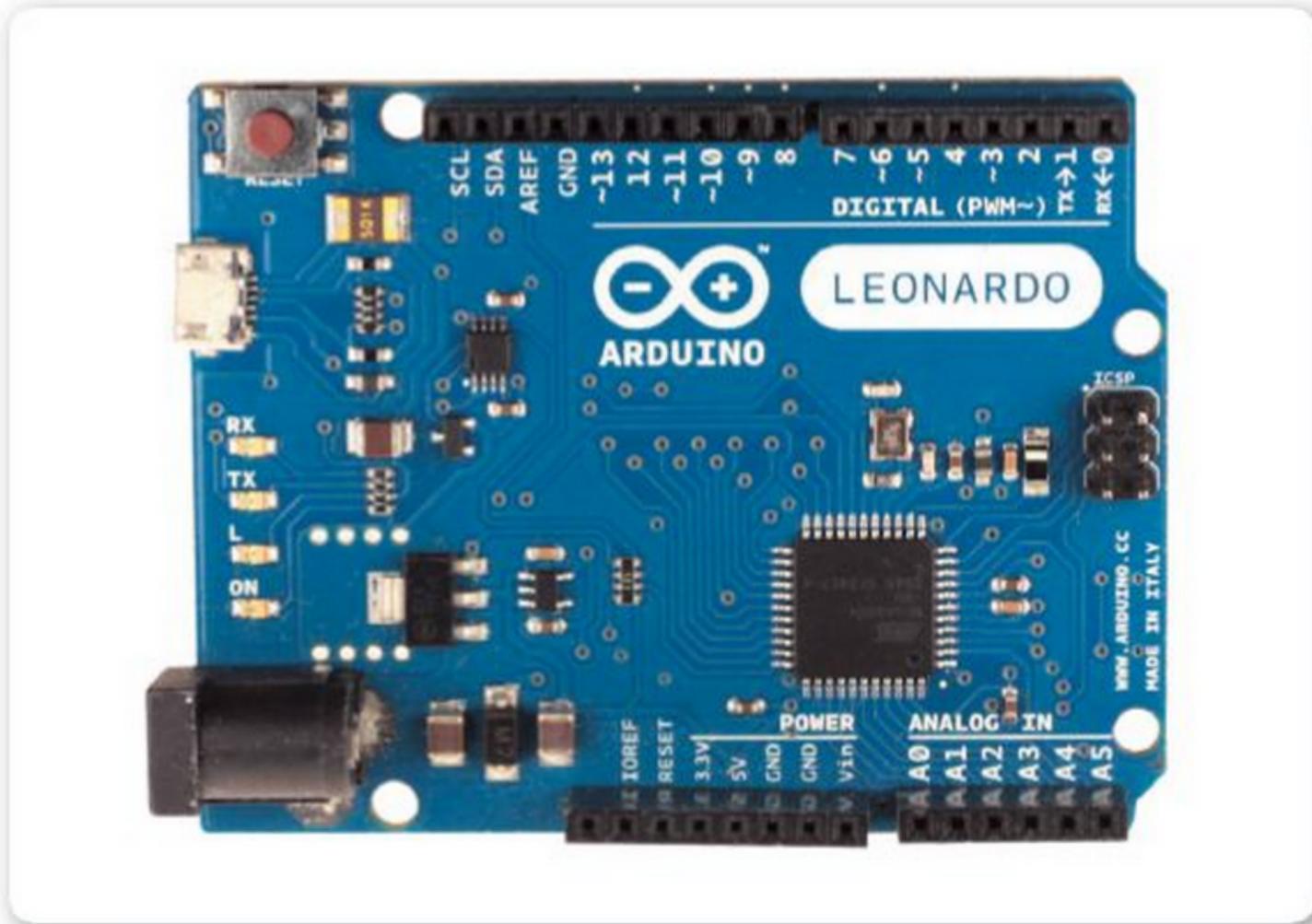


Figura 2. Arduino Leonardo , la placa novata del equipo, consta de 20 entradas/salidas digitales y 12 analógicas.



Figura 3. Arduino Mega es la más grande de la familia, no solo por su tamaño sino también por sus 54 entradas/salidas digitales y 16 analógicas.

Entorno

El entorno, como no podía ser de otra manera, es de código abierto. Arduino facilita escribir el código y cargarlo a la placa E/S. Funciona en Windows, Mac OS X y Linux. Está escrito en **Java** y basado en Processing, AVR-GCC y en otros programas también de código abierto.

La sintaxis de programación (**Processing**) es bastante parecida a C/C++, lo que hace que el que posea conocimientos en este lenguaje tenga alguna ventaja. En el lenguaje, es importante reconocer que hay dos partes: un **SETUP** para establecer las configuraciones iniciales y los estados de los pines y un **LOOP**, que es el bucle que se repite constantemente y es donde más se programa.

Además, disponemos de estructuras de control (if, for, switch, while, return, etcétera), variables (HIGH, LOW, INPUT, OUTPUT, etcétera), operadores matemáticos (+, -, =) y conversores (Char(), int(), float()), entre otras tantas funciones que se asemejan a cualquier lenguaje de programación y que iremos viendo a medida que las vayamos usando.

Una gran virtud que tiene el proyecto Arduino es la gran documentación que existe en la Web, tanto en español como en inglés, donde podemos encontrar innumerables ejemplos listos para funcionar. Este proyecto cuenta con una comunidad que ayuda tanto a desarrollar productos nuevos como a solucionar problemas técnicos.

Colaboradores

Varias personas colaboraron con el proyecto; en principio, fueron Massimo Banzi (ingeniero y profesor en IDII), David Cuartielles (ingeniero e investigador invitado en IDII), Tom Igoe (profesor de la Universidad de Nueva York), Gianluca Martino y David Mellis (estudiante en IDII). También colaboraron Nicholas Zambetti, Yaniv Steiner y Giorgio Olivero, quienes trabajaron con la plataforma Instant Soup.



Figura 4. Al ser hardware open source, el equipo se autofinancia con la venta y comercialización de su propia motherboard.

CAPACIDADES SEGÚN SU MICROCONTROLADOR			
	▼ ATMEGA168	▼ ATMEGA328	▼ ATMEGA1280
Voltaje operativo	5 V	5 V	5 V
Voltaje de entrada recomendado	7 - 12 V	7 - 12 V	7 - 12 V
Voltaje de entrada límite	6 - 20 V	6 - 20 V	6 - 20 V
Pines de entrada y salida digital	14 (6 proporcionan PWM)	14 (6 proporcionan PWM)	54 (14 proporcionan PWM)
Pines de entrada analógica	6	6	16
Intensidad de corriente	40 mA	40 mA	40 mA
Memoria flash	16 kB (2 kB reservados para el bootloader)	32 kB (2 kB reservados para el bootloader)	128 kB (4 kB reservados para el bootloader)
SRAM	1 kB	2 kB	8 kB
EEPROM	512 bytes	1 kB	4 kB
Frecuencia de reloj	16 MHz	16 MHz	16 MHz

Tabla 1. Niveles de funcionamiento según el tipo de microcontrolador.

➤ Evoluciones, versiones y periféricos

Las placas han pasado por diferentes versiones, ya sea porque han mejorado sus conectores o adaptadores, porque han aparecido necesidades específicas de desarrollo (por ejemplo, en robótica) o porque los fabricantes de los microcontroladores han actualizado sus componentes haciéndolos más pequeños y con mayores capacidades (lo que comúnmente sucede con todo tipo de aparato electrónico: lo mejoran y lo hacen más pequeño).

La primera placa fue **Severino (S3V3)**, que tenía conectores RS232 y estaba impresa de un solo lado (*single-sided board*). Luego, apareció la Arduino Serial, que estaba impresa de ambos lados y tenía un

menor tamaño. Al poco tiempo, y con los avances de los puertos USB, nació la **Arduino USB**, con la cual se hicieron varias versiones, principalmente, para que el puerto USB fuera compatible con otros dispositivos. Luego, le siguieron Extreme, Nuova Generazione, Diecimila y Duemilanove, cada una con sus respectivas mejoras, hasta llegar hoy a la **Arduino Leonardo**, que es la última.

Entre ellas, existen otras versiones que podemos clasificar por su tamaño: Stamp, Mini, Nano, BT (con Bluetooth incorporado), LilyPad (para prendas de vestir) y Fio (diseñada para conectarle productos de otro fabricante de dispositivos, XBee). Para finalizar, están la Mega y la Mega 2560, que son las que tienen más capacidad y entradas y salida digitales y analógicas. Por otro lado, el equipo Arduino y la comunidad desarrollan periféricos, entre ellos podemos encontrar: motores DC, motores paso a paso, relés, servos, joysticks, Bluetooth, Ethernet, GSM, radiofrecuencias, LCD, LCD TFT, LED, láseres, sensores, impresoras, entre otros.

De manera constante, salen productos (periféricos) nuevos al mercado; algunos están desarrollados específicamente para Arduino y se diferencian porque llevan impresa la denominación **Shield** y tienen la particularidad de ser encastrables entre sí y también con Arduino. Esto simplifica la tarea a la hora de desarrollar y testear, ya que elimina el protoboard y las conexiones mediante cables.



Figura 5. Equipo Arduino, de izquierda a derecha: Gianluca Martino, David Cuartielles, Massimo Banzi. Atrás: Dave Mellis, Tom Igoe (foto: flickr David Cuartielles).



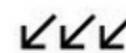
Figura 6. IDE de desarrollo V1.0.5: aquí programamos el código que queremos que reproduzca la placa.

Librerías

Además de todo el hardware desarrollado, tanto por el equipo Arduino como por su comunidad, existen innumerables dispositivos que, si bien no han sido diseñados especialmente para Arduino, pueden funcionar sin ningún tipo de problema gracias a las librerías. Las librerías son programas creados por terceros, que podemos utilizar libremente y que nos permiten, sin conocer un determinado periférico, hacerlo funcionar correctamente.



INTRODUCIRSE EN EL MUNDO ARDUINO



Para introducirnos en el proyecto, lo primero que debemos hacer es comprar una placa Arduino: recomendamos la **Uno R3** o la **Leonardo**. Luego, tenemos que bajar del sitio oficial (<http://arduino.cc>) el IDE de desarrollo para poder codificar el microcontrolador. Por último, debemos probar y probar. También existe un foro oficial con una comunidad muy solidaria donde podemos hacer consultas.

Cabe destacar que estas librerías se aplican a todos los dispositivos conectados a la Arduino, sean específicamente fabricados o no para esta placa.

Las más conocidas son **Ethernet**, que permite la conexión a una red; **LiquidCrystal**, que hace funcionar los visores LCD, los Servos y SPI, y que es muy utilizada en comunicaciones como, por ejemplo, en lectores de radiofrecuencia, en Software Serial para impresoras y en sensores; **XBee** para conexiones Wi-Fi; **SD** para interactuar con tarjetas flash; **Keyboard** para simular un teclado, entre otras tantas librerías que son de uso más específico (**Mouse**, **Robot**, **Esplora**, **TFT**, etcétera).

Las posibilidades son muchísimas, el límite lo da la imaginación: desde el simple encendido y apagado de un diodo hasta hacer domótica, desde medir la temperatura ambiente hasta controlar un brazo robot.

En la Web podemos ver videos de proyectos como, por ejemplo, alarmas, controles de presencia, medidores de temperatura, aperturas de puertas, impresoras 3D, impresoras que se activan cada vez que reciben un twit, remeras con luces psicodélicas, entre otros.

Cada día surgen nuevos proyectos. Si bien la mayoría son de aficionados, también están apareciendo proyectos más elaborados de empresas que están incorporando la placa Arduino a sus productos.

Un ejemplo: Blink

Analicemos un ejemplo bien simple, como puede ser prender y apagar un diodo. En este caso, trabajamos con la placa Arduino Uno R3, que es la más difundida y estándar.

Esta placa consta de 14 entradas o salidas digitales y 6 entradas o salidas analógicas. Además, tiene la particularidad de traer, en el pin 13, un diodo con una resistencia incorporado a la placa, lo que hace que podamos disponer de un diodo para pruebas.

A continuación, presentamos el código adecuado:

```
//Aquí le indicamos al microcontrolador que usaremos el pin 13
int led = 13;
//Seteamos el pin13 como salida
void setup() {
  pinMode(led, OUTPUT);
```

```
}  
//Bucle infinito  
void loop() {  
digitalWrite(led, HIGH); // Prendo la luz  
delay(1000); // Espero un segundo  
digitalWrite(led, LOW); // Apago la luz  
delay(1000); // Espero un segundo  
}
```

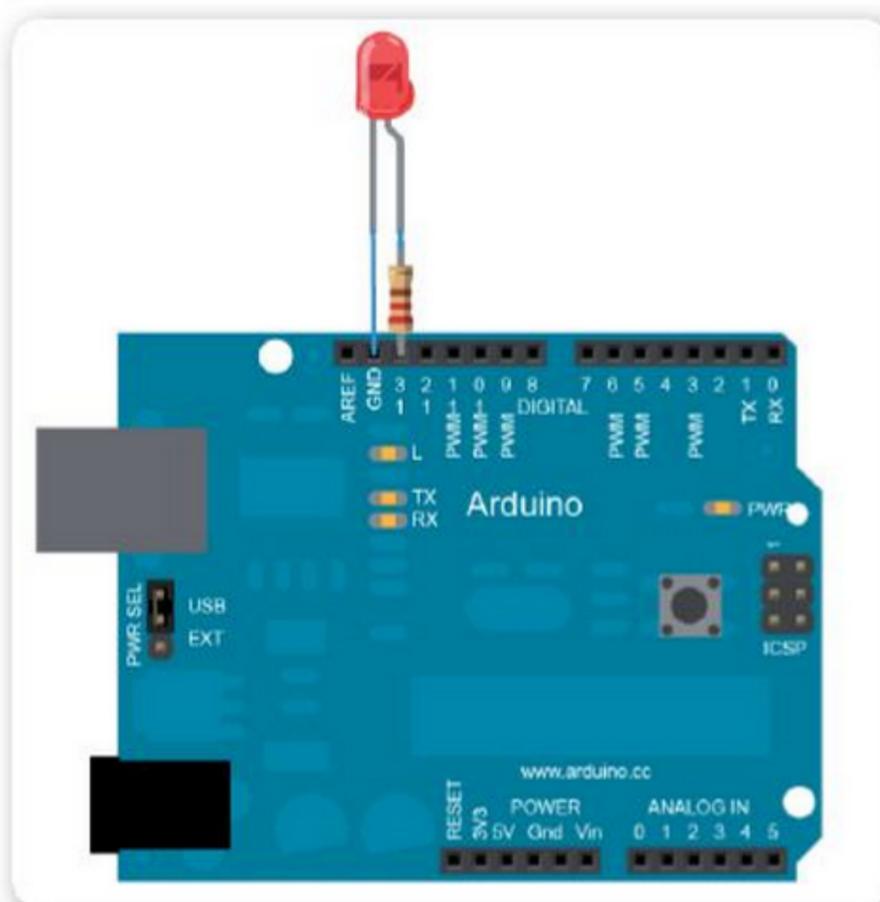


Figura 7. Como vemos, es muy fácil conectar el dispositivo para realizar el ejemplo que proponemos.



Programación de Arduino

Un lenguaje de programación es un conjunto de reglas, símbolos y palabras, formalmente establecido, cuyo propósito es expresar procesos específicos que una máquina puede ejecutar de una manera ordenada.

Estos lenguajes pueden expresar programas, pueden decidir el comportamiento lógico y físico de una máquina, pueden generar algoritmos, o bien ser una manera de comunicación humana.

Dentro de los lenguajes de programación, existe el proceso en el que se escribe, depura, compila, ejecuta y se mantiene el código fuente de un programa informático. Este proceso recibe el nombre de programación. Los lenguajes de programación se pueden clasificar de acuerdo con niveles dados: cuánto se asemeja el lenguaje al lenguaje máquina o, en el otro extremo, al lenguaje natural que todos conocemos y manejamos.

Figura 8. Java es un lenguaje de programación de propósito general, orientado a objetos y basado en clases, diseñado para tener pocas dependencias de implementación.



Lenguajes de bajo nivel

Son lenguajes totalmente dependientes de la máquina, es decir que el programa que se realiza con este tipo de lenguajes no se puede migrar o utilizar en otras máquinas. Al estar prácticamente diseñados a medida del hardware, aprovechan al máximo las características de este.

```

org 0x0000          ;org coloca el programa en el origen.
movlw 0x07          ;Deshabilito los comparadores
movwf CMCON

bsf STATUS, RP0    ;entra a Bank 1 para poder acceder a TRISB
clrf TRISB         ;Configuro PortB como salida
bcf STATUS, RP0    ;retorna a de nuevo el Bank 0

Loop
movlw b'00000001'  ;enciendo LED
movwf PORTB
nop                ;nop no hace nada
nop
call Delay         ;Llamo a la rutina de retardo de 200ms
movlw b'00000000'  ;apago LED
movwf PORTB
call Delay         ;Llamo la rutina de retardo de 200ms
goto Loop          ;Inicia de nuevo el ciclo.

;****SUBROUTINA DE RETARDO****;
; Esta rutina retarda 200ms. ;
;*****;
Delay
movlw d'200'       ;Espero 200 ms (Para un reloj de 4 MHz)
movwf count1
d1 movlw 0xc7
movwf counta
movlw 0x01
movwf countb
Delay_0
decfsz counta, f
goto $+2
decfsz countb, f
goto Delay_0

decfsz count1, f
goto d1
retlw 0x00

end

```

Figura 9. Ensamblador es un lenguaje de bajo nivel para computadoras, microprocesadores, microcontroladores y otros circuitos integrados programables.

Lenguajes de alto nivel

Se trata de lenguajes independientes de la arquitectura de la computadora, por lo que, en principio, un programa escrito en un lenguaje de alto nivel se puede migrar de una máquina a otra sin ningún tipo de problema.

Estos lenguajes permiten al programador olvidarse por completo del funcionamiento interno de la máquina para la que están diseñando el programa; tan solo necesitan un traductor que entienda el código fuente y las características de la máquina.

```

1 package tresnumerosordenados;
2 import java.util.Scanner;
3
4 /**
5  *
6  * @author tutorias.co
7  */
8
9 public class TresNumerosOrdenados {
10
11     static Scanner entrada = new Scanner(System.in);
12
13     public static void main(String[] args) {
14         // TODO code application logic here
15
16         int numero1, numero2, numero3, menor, medio, mayor;
17
18         System.out.print("Favor ingresar el numero 1: ");
19         numero1 = entrada.nextInt();
20         System.out.print("Favor ingresar el numero 2: ");
21         numero2 = entrada.nextInt();
22         System.out.print("Favor ingresar el numero 3: ");
23         numero3 = entrada.nextInt();
24
25         if((numero1 <= numero2)&&(numero1 <= numero3)){
26
27             menor = numero1;
28
29             if(numero2 <= numero3){
30                 medio = numero2;
31                 mayor = numero3;
32             }else{
33                 medio = numero3;
34                 mayor = numero2;
35             }
36         }else if((numero2 <= numero1)&&(numero2 < numero3)){
37
38             menor = numero2;
39
40             if(numero1 <= numero3){
41                 medio = numero1;
42                 mayor = numero3;
43             }else{
44                 medio = numero3;
45                 mayor = numero1;
46             }
47         }else{
48             menor = numero3;
49
50             if(numero1 <= numero2){
51                 medio = numero1;
52                 mayor = numero2;
53             }else{
54                 medio = numero2;
55                 mayor = numero1;
56             }
57         }
58     }
59 }

```

Figura 10. Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis limpia y en favorecer un código legible.

Suelen usar tipos de datos para la programación y hay lenguajes de propósito general (cualquier clase de aplicación) y de propósito específico (como **FORTRAN** para trabajos científicos).

Lenguajes de medio nivel

Estos lenguajes se encuentran en un punto intermedio entre los dos anteriores. Dentro de este grupo podría situarse el lenguaje C, ya que, por un lado, puede acceder a los registros del sistema y trabajar con direcciones de memoria –características de lenguajes de bajo nivel– y, a la vez, permite realizar operaciones de alto nivel.

Plataforma Arduino

Ahora hablaremos sobre los lenguajes de programación que se pueden utilizar en la plataforma Arduino. Al tratarse de una plataforma de hardware libre, en Arduino se podrán encontrar distintos y muy variados lenguajes y entornos de programación para desarrollar diferentes proyectos. A continuación, mencionamos algunos de ellos:

- 3DVIA Virtools (aplicaciones interactivas y de tiempo real)
- Adobe Director
- BlitzMax (con acceso restringido)
- C
- C++ (mediante libSerial o en Windows)
- C#
- Cocoa/Objective-C (para Mac OS X)
- Flash (mediante ActionScript)
- Gambas
- Isadora (interactividad audiovisual en tiempo real)
- Instant Reality (X3D)
- Java
- Liberlab (software de medición y experimentación)
- Mathematica
- Matlab
- MaxMSP (entorno gráfico de programación para aplicaciones musicales, de audio y multimedia)
- Minibloq (entorno gráfico de programación; corre también en OLPC)
- Perl
- Php
- Physical Etoys (entorno gráfico de programación usado para proyectos de robótica educativa)
- Processing

- Pure Data
- Python
- Ruby
- Scratch for Arduino (S4A) (entorno gráfico de programación, modificación del entorno para niños Scratch, del MIT)
- Squeak (implementación libre de Smalltalk)
- SuperCollider (síntesis de audio en tiempo real)
- VBScript
- Visual Basic .NET
- VVVV (síntesis de video en tiempo real)

Arduino se comunica mediante la transmisión de datos en formato serial, el que es soportado por la mayoría de los lenguajes anteriormente nombrados. Para los que no admiten el formato serie de forma nativa, se puede utilizar software que traduzca las instrucciones enviadas por ambas partes para permitir una comunicación compatible.

La lista de lenguajes de programación contiene programas opcionales para utilizar de una manera más específica la plataforma Arduino, ya que el más utilizado es Processing.

Processing es un lenguaje de programación de código abierto, lo cual quiere decir que cualquier persona puede hacer uso de él, ya que nadie cuenta con los derechos exclusivos y no se cobra por el desarrollo de programas. Está basado en Java, es fácil de aprender y utilizar y sirve para enseñar y producir proyectos interactivos y multimedia de diseño digital.

Lenguajes C y C++

C es un lenguaje de programación de propósito general que ofrece economía sintáctica, control de flujo, estructuras sencillas

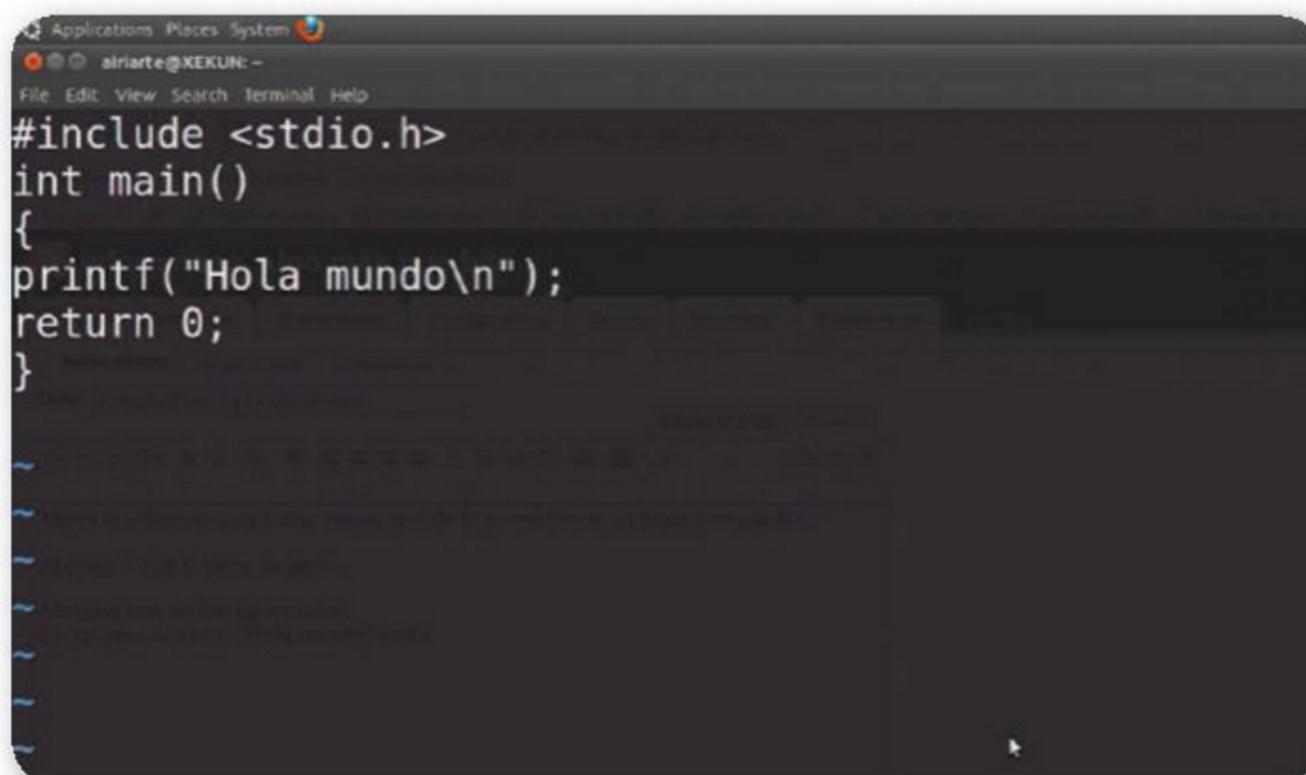


VISUAL BASIC



Es un lenguaje de programación dirigido por eventos, desarrollado por Alan Cooper para Microsoft. Es un dialecto de BASIC, con importantes agregados. Su primera versión, presentada en 1991, tuvo la intención de simplificar la programación utilizando un ambiente de desarrollo completamente gráfico para la creación de interfaces gráficas y la programación misma.

y un buen conjunto de operadores. No es de muy alto nivel, es un lenguaje pequeño, sencillo, que no está especializado en ningún tipo de aplicación. Esto lo convierte en un lenguaje potente, con un campo de aplicación ilimitado y, sobre todo, que se aprende rápidamente. En poco tiempo, un programador puede utilizar la totalidad del lenguaje C.



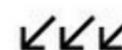
```
Applications Places System
alriarte@XEKUN: ~
File Edit View Search Terminal Help
#include <stdio.h>
int main()
{
printf("Hola mundo\n");
return 0;
}
```

Figura 11. C es un lenguaje de programación creado en 1972 por Dennis M. Ritchie en los Laboratorios Bell, como evolución del anterior lenguaje B, a su vez basado en BCPL.

La base del C proviene del BCPL, escrito por Martin Richards, y del B, escrito por Ken Thompson en 1970 para el primer sistema UNIX en un DEC PDP-7. Ambos son lenguajes sin tipos, al contrario del C, que proporciona varios tipos de datos. Los tipos que ofrece son caracteres, números enteros y en coma flotante, de varios tamaños.



LENGUAJE C++



Es un lenguaje de programación diseñado a mediados de los años 80 por Bjarne Stroustrup con la intención de extender al lenguaje de programación C, con mecanismos para la manipulación de objetos. Luego, se sumaron facilidades de programación genérica a los otros dos paradigmas: la programación estructurada y la programación orientada a objetos. Por esto, C++ es un lenguaje de programación multiparadigma.

Además, se pueden crear tipos derivados mediante la utilización de punteros, vectores, registros y uniones. El primer compilador de C fue escrito por Dennis Ritchie para un DEC PDP-11, quien escribió el propio sistema operativo en C.

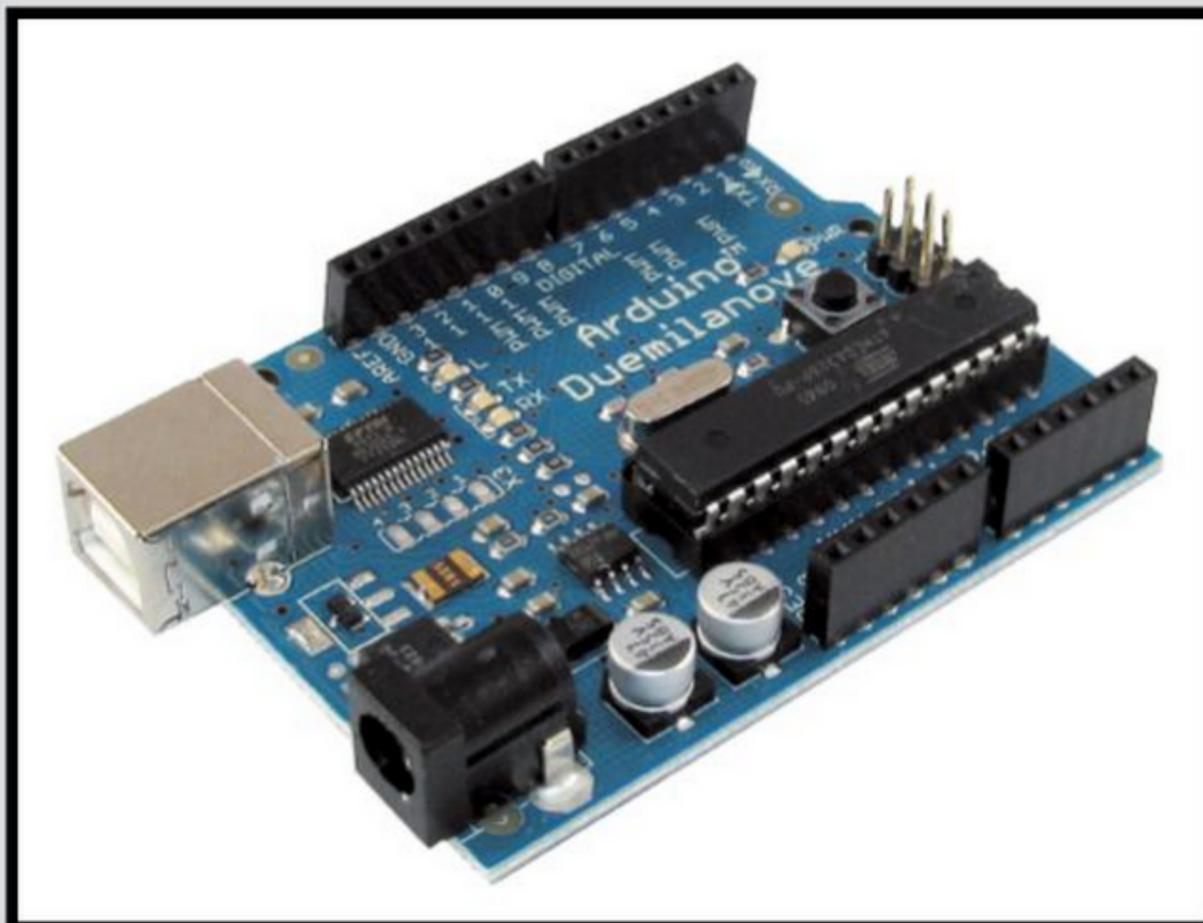
C++ es un lenguaje imperativo orientado a objetos, derivado del C. En realidad, es un superconjunto de C, creado para añadirle cualidades y características.

El resultado es que C++, como su ancestro, sigue muy ligado al hardware subyacente, mantiene una considerable potencia para programación a bajo nivel, pero se la han añadido elementos que le permiten también un estilo de programación con un alto nivel de abstracción. Claro que C++ es un lenguaje de programación extremadamente largo y complejo; cuando nos adentramos en él, parece no acabar nunca.

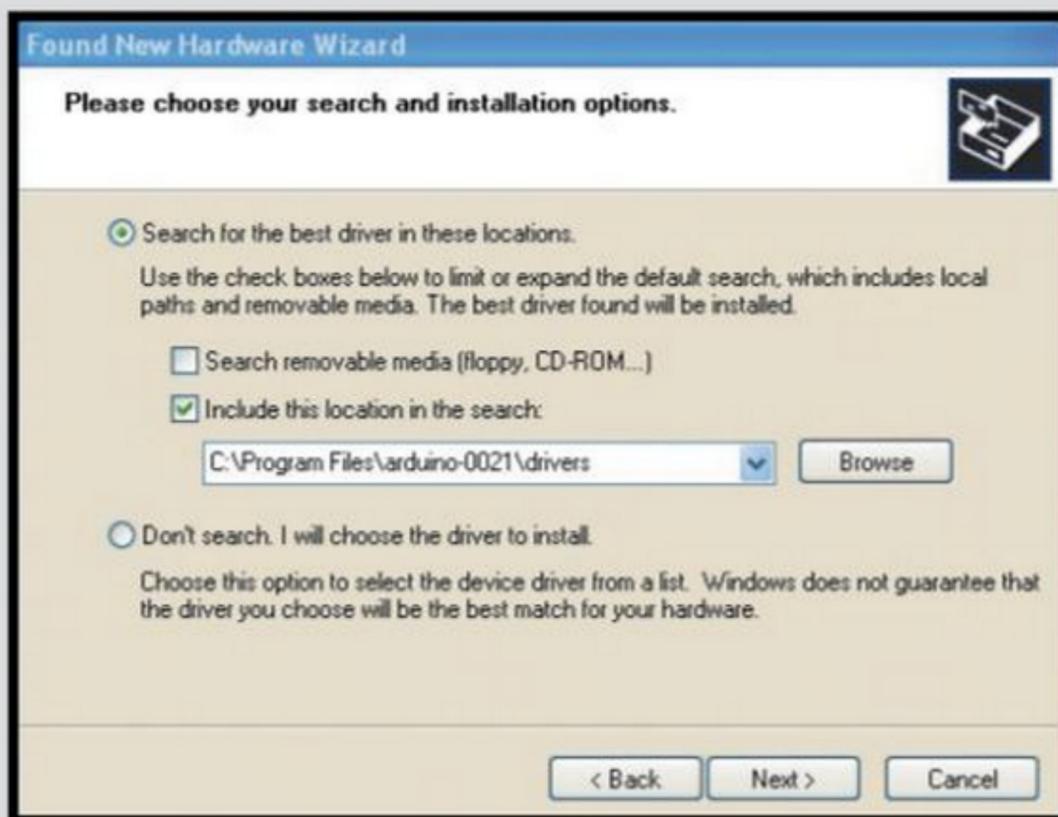
PAP: INSTALACIÓN Y CONFIGURACIÓN DEL IDE



01 Consiga una placa Arduino y un cable USB para llevar a cabo la conexión, una vez terminada la instalación del IDE de Arduino.



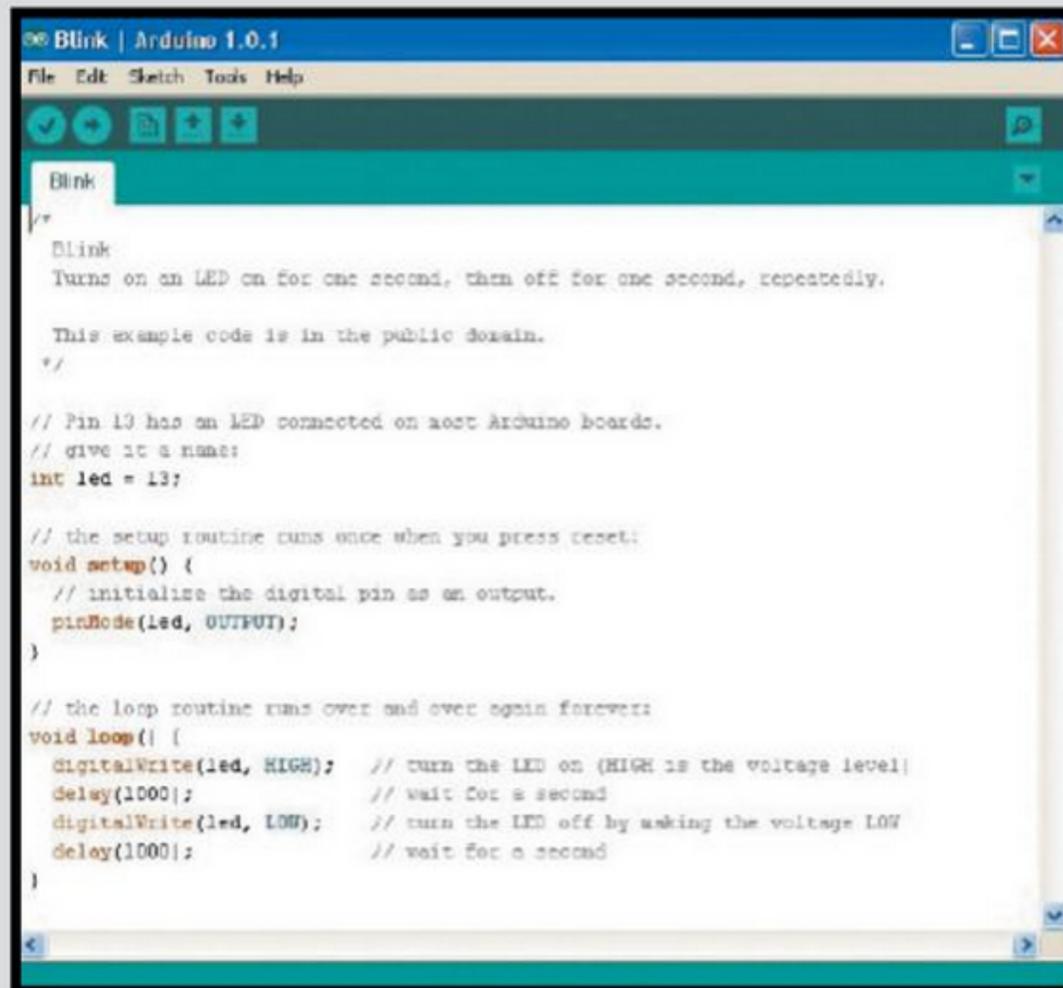
02 A continuación, descargue los drivers de la placa de Arduino con la que cuenta para que se pueda comunicar.



03 Conecte la placa Arduino por medio del cable USB y, después, proceda a instalar los drivers que haya descargado previamente.



04 Proceda a ejecutar la aplicación del IDE, seleccionando el tipo de tarjeta, el puerto serie y, por último, suba el programa que muestra Blink en Upload to I/O Board .

A screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.0.1". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for opening files, saving, and uploading. The main text area contains the following code:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by asking the voltage LOW
  delay(1000);             // wait for a second
}
```

Bibliotecas

Una biblioteca es, de alguna manera, un programa que realiza un cierto proceso que se puede llamar desde el programa que se esté escribiendo. No es propiamente un programa, ya que su función principal no es comportarse de una manera autónoma, sino realizar un cierto proceso en cuanto se la invoca desde el programa que el usuario está definiendo.

Con ayuda de las bibliotecas se pueden realizar procesos bastante más complejos, ya que estas evitan que se tenga que definir un cierto proceso que pudiera entenderse como un estándar para poder realizar un programa.

Width and Height.

The 'width' and 'height' variables contain the width and height of the display window as defined in the size() function.

```
void setup() {
  size(640, 360);
}

void draw() {
  background(127);
  noStroke();
  for (int i = 0; i < height; i += 20) {
    fill(129, 206, 15);
    rect(0, i, width, 10);
    fill(255);
    rect(i, 0, 10, height);
  }
}
```

Figura 12. Processing es un lenguaje de programación, un entorno de desarrollo y una comunidad online desde 2001. Fue creado para servir como un sketchbook de Dot Software.

Las bibliotecas pueden vincularse a un programa (o a otra biblioteca) en distintos puntos del desarrollo o la ejecución, según el tipo de vínculo que se quiera establecer.

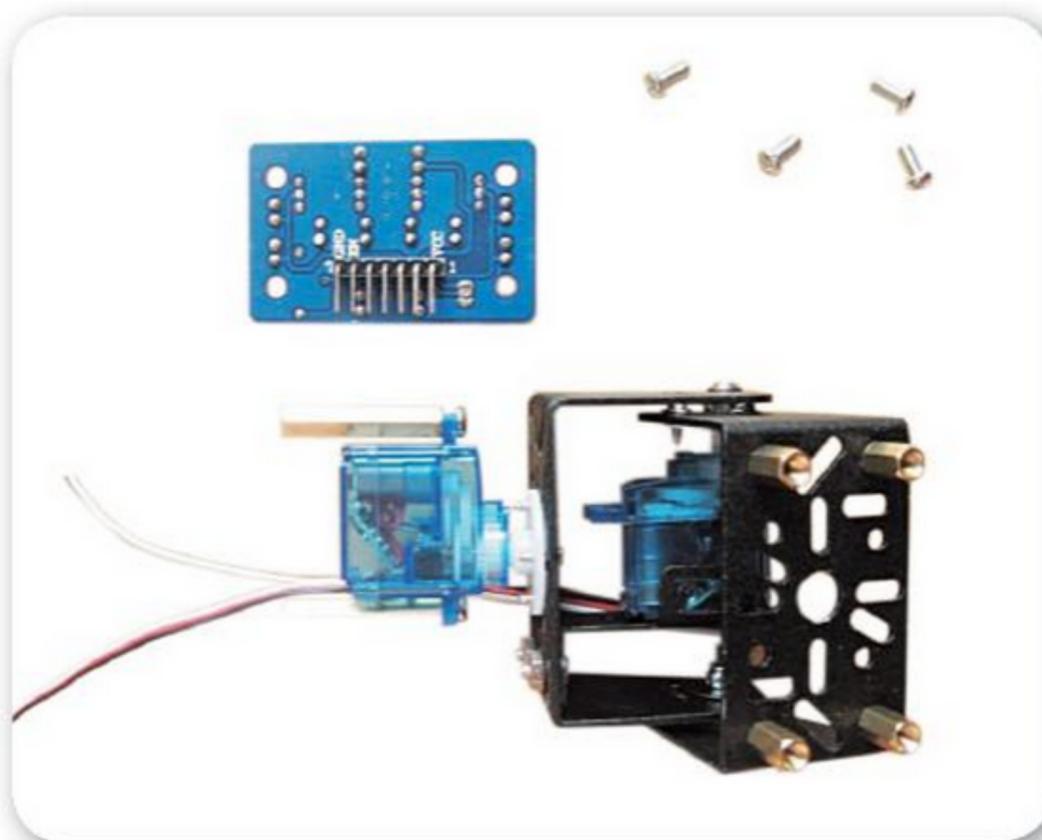


Figura 13. Una aplicación de Arduino en el área de redes, específicamente, en una aplicación de monitoreo y seguridad por medio de video.

UNA BIBLIOTECA
SE VINCULA A UN
PROGRAMA EN
DISTINTOS PUNTOS
DEL DESARROLLO



La mayoría de los sistemas operativos modernos proporcionan bibliotecas que implementan los servicios del sistema. De esta manera, estos servicios se han convertido en una materia prima que cualquier aplicación moderna espera que el sistema operativo ofrezca. Como tal, la mayor parte del código utilizado por las aplicaciones modernas se ofrece en estas bibliotecas.

Retomando las bibliotecas incluidas en Processing, se pueden encontrar las siguientes, de una manera nativa:

Video: la biblioteca de video le permite a Processing mostrar y ejecutar archivos de video, tomar datos de video desde una cámara y hacer directamente videos desde un programa en ejecución. El video puede ser capturado desde cámaras USB, cámaras con Fireware (estándar IEEE 1394) o tarjetas de video con entrada de cable compuesto o de s-video conectadas a la computadora. El video puede cargarse en archivos locales de la computadora o en alguna parte de internet.

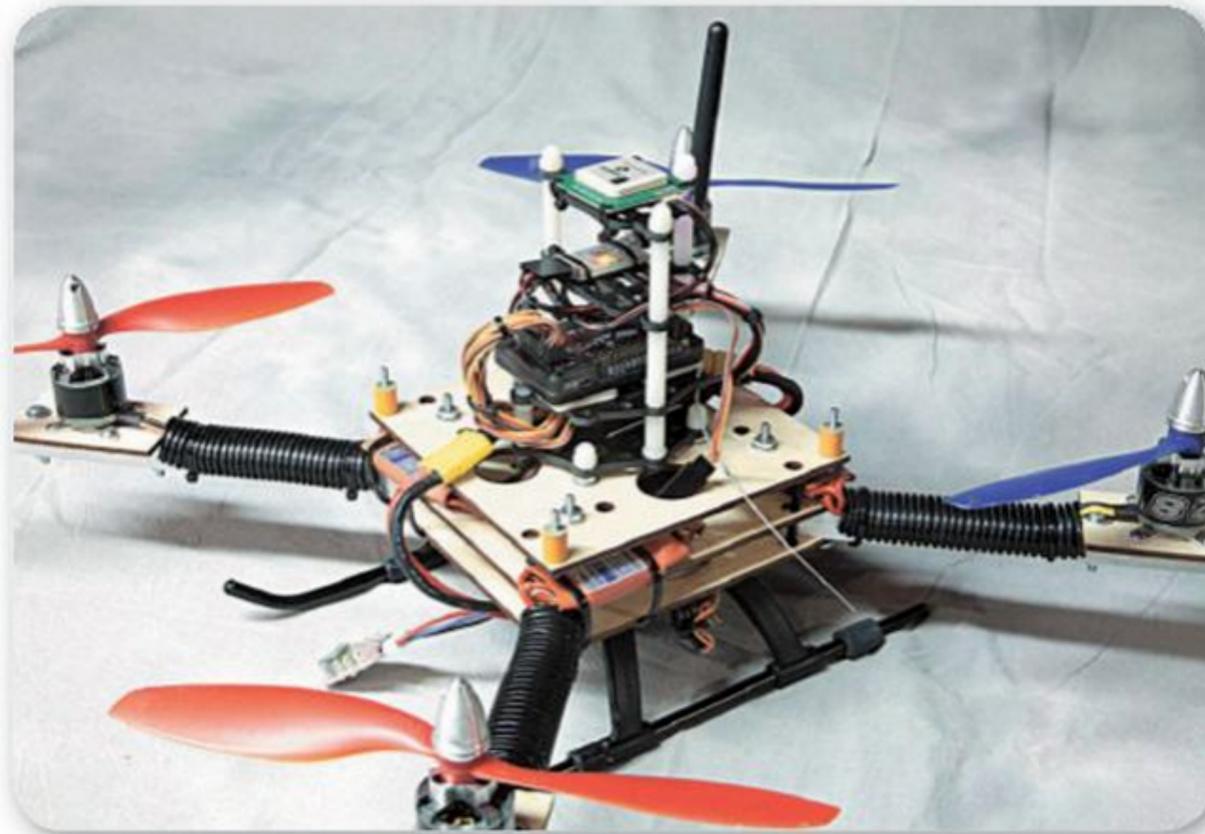


Figura 14. Existen varias aplicaciones de Arduino posibles de realizar con las bibliotecas. Una de ellas es el **ArduPilot**, un cuadricóptero funcional.

Serial: es utilizada para leer datos de una fuente externa de dispositivos y para escribir datos en ella, un byte a la vez. Permite a dos computadoras enviar y recibir datos. Tiene la flexibilidad de comunicarse con controladores personalizados y permite usarlos como dispositivos de entrada o de salida para programas de Processing. El puerto serial es un puerto de nueve pines de E/S (entrada/salida) que existe en distintas computadoras y que también puede emularse mediante USB.

Network: facilita leer y escribir información a través de máquinas en internet. Permite la creación de clientes y servidores. Un servidor se conecta a una lista de clientes para leer y escribir datos. Un cliente es capaz de leer y escribir datos en un servidor.

DXF Export: escribe todas las gráficas basadas en triángulos (polígonos, cajas, esferas, etc.) en un archivo DXF, funciona con `beginRaw()` y `endRaw()`. Estas funciones pueden tomar y guardar la información de las formas justo antes de ser renderizadas en la pantalla. En esta etapa, la escena completa no es más que una lista de líneas y triángulos. Esto significa que una forma creada con la función `sphere()` estará hecha de cientos de triángulos, a diferencia de un simple objeto.

PDF Export: permite escribir archivos PDF directamente desde Processing. Estos archivos de gráficos vectoriales pueden ser escalados a cualquier tamaño y rendimiento en resoluciones muy altas. Esta biblioteca se utiliza frecuentemente con el tamaño de la función central de Processing, `size()`, con una combinación de `beginRecord()` y `endRecord()` o con `beginRaw()` y `endRaw()`. La función `createGraphics()` también puede ser útil.

Minim: es una biblioteca de audio que utiliza la API JavaSound, un poco de Tritonus y MP3SPI de Javazoom para facilitar el uso de la biblioteca de audio a las personas que desarrollan proyectos en el entorno de Processing. Esta API permite realizar la integración de audio dentro de los bocetos de una forma sencilla. Además, ofrece una cantidad razonable de flexibilidad a los usuarios más avanzados. Podemos obtener Minim en dos versiones: una con los archivos JAR necesarios para utilizar Minim, y una distribución completa que incluye los javadocs, ejemplos y código fuente.

EN PROCESSING
HAY BIBLIOTECAS
DE PUERTO SERIAL,
VIDEO, COMPILACIÓN
Y ANIMACIÓN



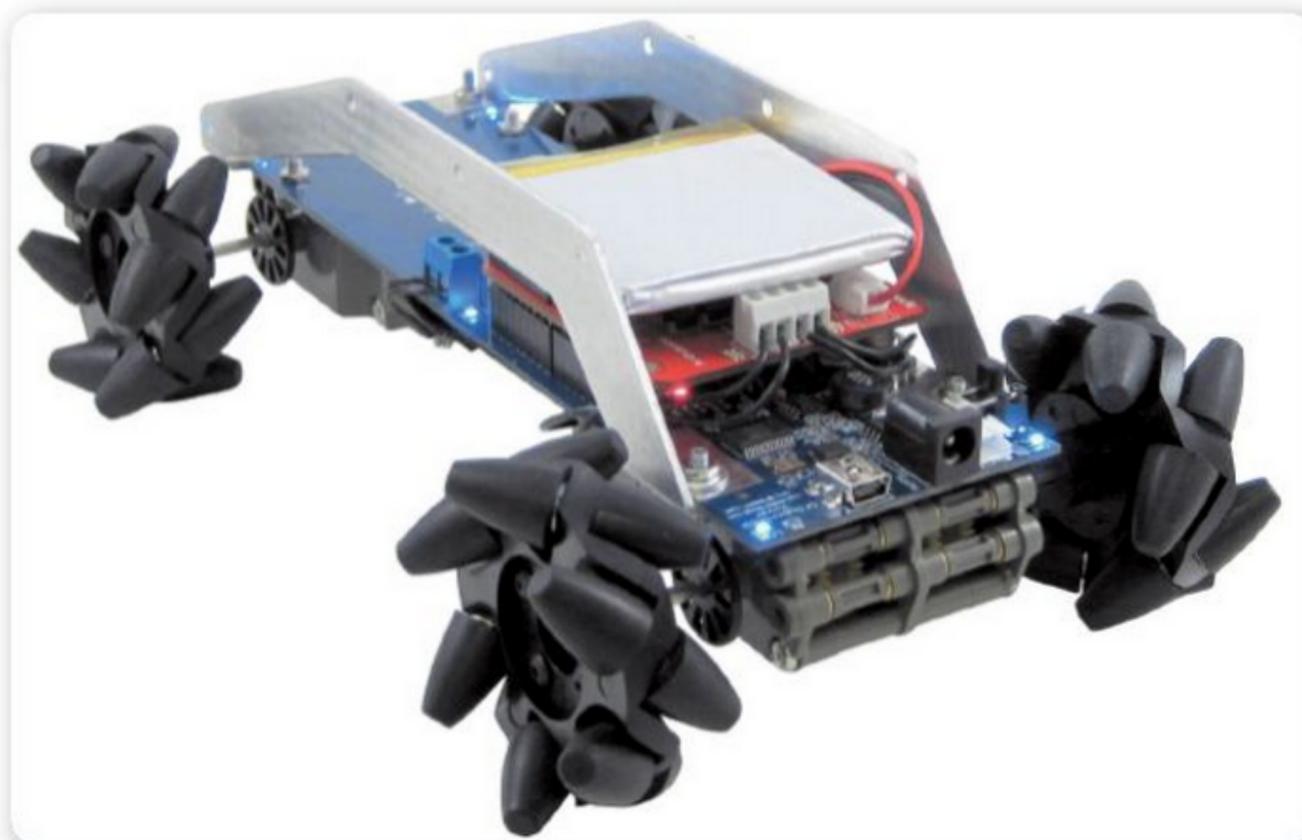


Figura 15. Ejemplo de Arduino utilizado en robótica. La placa es capaz de comunicarse con distinto hardware para realizar operaciones concretas.

Existen otras bibliotecas conocidas como contribuciones, que están documentadas, desarrolladas y mantenidas por miembros de la comunidad de Processing. A continuación, mencionamos algunas de ellas:

- Dentro de la categoría 3D: ProScene, ODC (Objective Camera Direction), Stereo, HE_Mesh, Patchy, SimpleOpenNI, PeasyCam, OBJLoader, iGeo, Picking, Shapes3D.
- Dentro de la categoría de animación: Ani, Sprites, Frames, Motion, Hermes, ijeomamotion.
- Dentro de la categoría de compilación: gicentreUtils, GenerativeDesign, Hermes.



INFORMACIÓN ADICIONAL SOBRE ARDUINO



Las aplicaciones que nos ofrece Arduino son múltiples. Mediante sensores, podemos crear aplicaciones sencillas enfocadas a la docencia, para estudiantes de electrónica, proyectos más elaborados para la industria o, incluso, sistemas dirigidos simplemente al ocio. Esta plataforma es muy utilizada también en los entornos artísticos para crear obras más elaboradas, dada su facilidad de programación.

- Dentro de la categoría de datos: GML4U, YahooWeather, ShortMessage, SFTP, Share, Xlsreader, QRCode, Shaer, Carnivore, MapThing, oscPS, UDP, SoundCloudBezierQLib, SigantFilter, GNet.
- Dentro de la categoría de GUI: G4P, Guido, ControlPS.
- Dentro de la categoría geometría: Gemoerative, Hermes, Computational Geometry, point2line.
- Dentro de la categoría hardware: MostPixelsEver, Ketai, LeapMotion, Sudden Motion Sensor, Blink Stick, Apple Light Sensor, dmxPS12, Leap Motion for Processing.
- Dentro de la categoría E/S: Collada Loader for Google Sketchup, Gamepad, Simple Multi Touch, exGestures, proIMS, IgnoCodeLib, NXTComm Processing, OBJExport, Tablet, MindSet Processing, SelectFile.

EL ENTORNO DE
DESARROLLO DE
ARDUINO ESTÁ
ESCRITO EN JAVA
Y PROCESSING



Existen otras categorías en donde se encuentran listadas las demás bibliotecas, ya que cada vez se agregan más a la lista de las bibliotecas de contribución.

Entorno de desarrollo

El entorno de desarrollo de Arduino es un entorno de código abierto en donde resulta sencillo escribir código para posteriormente cargarlo a la placa de E/S. Funciona en las principales plataformas de computación, como Windows, Mac OS X y Linux.



MICROCONTROLADORES UTILIZADOS EN ARDUINO

Los microcontroladores más usados son el **ATmega168**, **ATmega328**, **ATmega1280** y **ATmega8**, ya que, por su sencillez y bajo costo, permiten el desarrollo de múltiples diseños. Por otro lado, el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque (bootloader) que corre en la placa.

Está escrito en **Java** y su base es el lenguaje **Processing**, aunque también cuenta con otros elementos de código abierto.

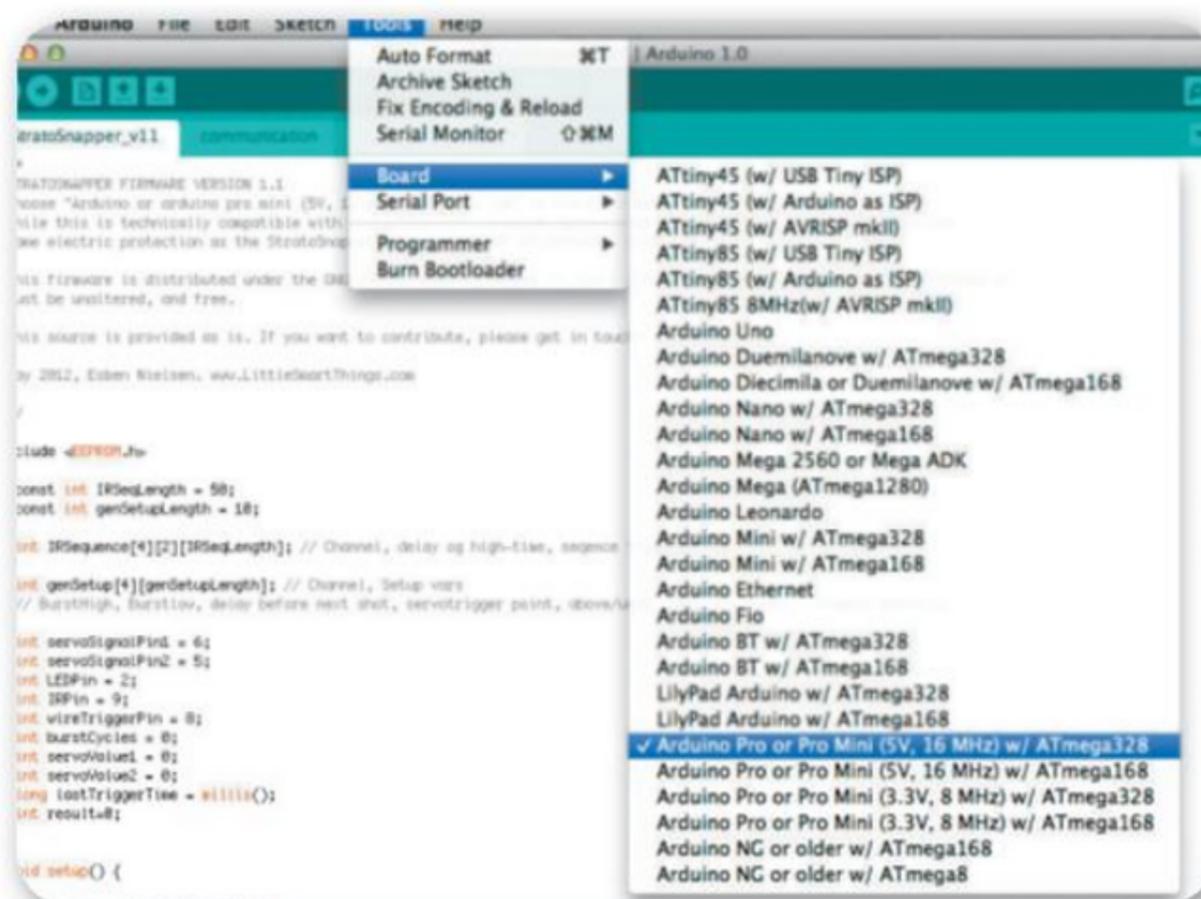


Figura 16. Imagen de cómo luce el **Arduino IDE**, para realizar sketches y, posteriormente, cargarlos a la placa Arduino.

Antes de empezar a hablar del IDE de Arduino, es necesario saber exactamente qué es un IDE. Un programa es un conjunto concreto de instrucciones, agrupadas y ordenadas en una forma adecuada que tiene como objeto realizar un proceso determinado. Cuando se dice que un microcontrolador es programable, nos referimos a que su memoria puede contener, de manera permanente, el programa que deseemos que contenga para que el microcontrolador lo ejecute; de lo contrario, si el microcontrolador no contiene ningún programa, no sabrá qué tipo de tarea deberá desempeñar.

Las siglas IDE se refieren a Entorno Interactivo de Desarrollo o *Integrated Development Environment*. En cuanto a lo que se refiere exactamente al entorno de desarrollo, esta es una manera de llamar al conjunto de herramientas de software que permite a los programadores realizar desarrollos, escribir sus códigos y hacer pruebas para el proyecto que necesitan crear. Para el caso de Arduino, se requiere un IDE que permita la realización de los programas, para escribirlos y editarlos.

Dentro del mundo de Arduino, el programa que se escribe y edita se conoce como **Sketch**, que hace referencia a una versión de prueba o prototipo de lo que sería el proyecto que vamos a realizar. De igual manera, necesitamos que el IDE nos permita comprobar que nuestro proyecto no tenga errores, asegurar que sea correcto y que todo se encuentre en orden para proceder a grabarlo en la memoria del microcontrolador de la placa Arduino; en ese momento, pasa a ser el ejecutor autónomo de dicho programa.

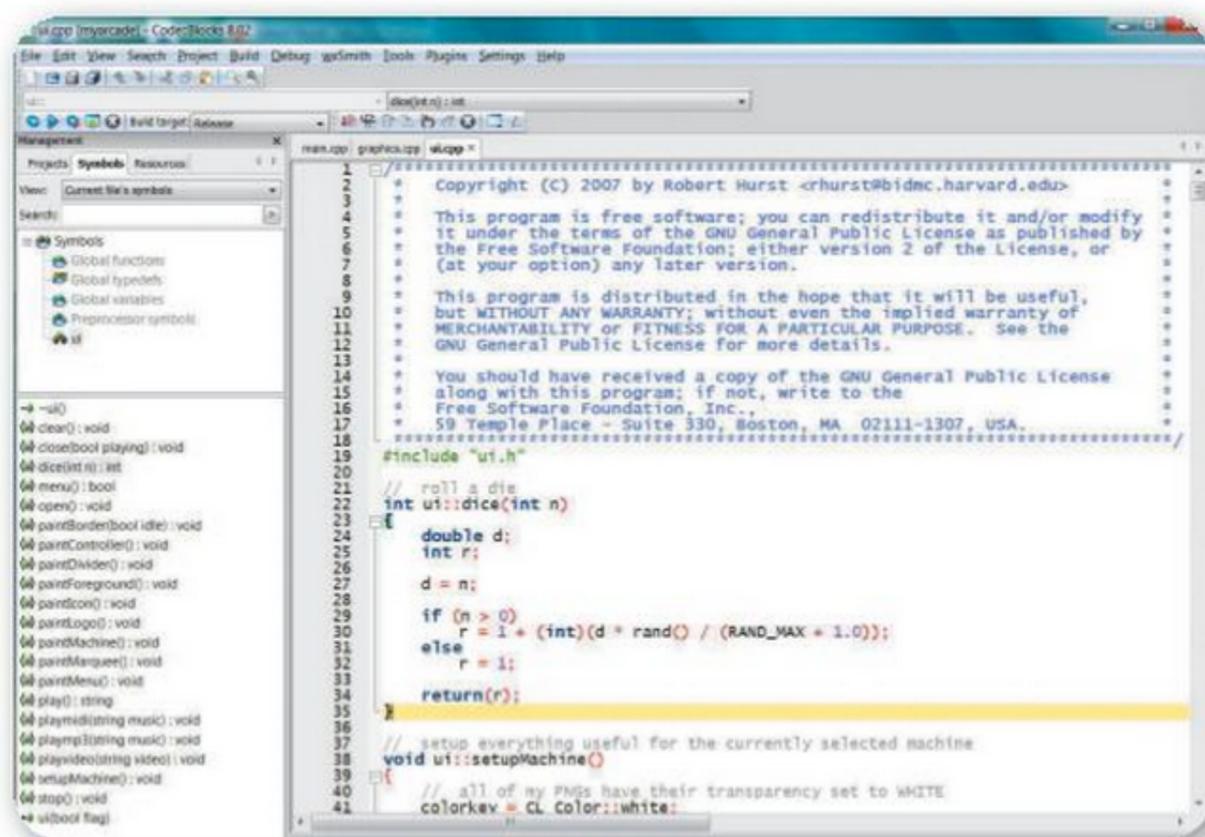


Figura 17. CodeBlocks es un IDE o entorno de desarrollo opcional al entorno de desarrollo oficial de Arduino, con algunas características extra.

Existen también IDEs alternativos al IDE oficial de desarrollo de Arduino, ya que hay usuarios que no necesitan opciones más avanzadas dentro del IDE oficial. Un ejemplo de esto es el autocompletado de sentencias o, también, debido a la dependencia del lenguaje Java, ya que se deben tener instalados más paquetes de los que realmente se utilizan para la compilación y carga de los sketches.

Otra de las razones para tener un IDE alternativo es que los desarrolladores pueden estar familiarizados con un IDE en particular, que no sea el oficial para Arduino y quieran seguir utilizándolo para sus desarrollos en esta plataforma. Por tales motivos es que existe una gran variedad de opciones de IDEs alternativos, que pueden aportar características adicionales o cuya manera de trabajo es distinta.

A continuación, describimos algunos de los IDEs alternativos:

CodeBlocks : es un IDE libre y multiplataforma para el desarrollo de aplicaciones escritas en lenguaje C y C++. Sin embargo, también se pueden escribir programas directamente en el lenguaje Arduino (y que sean cargados en la placa) mediante una versión modificada del software, mejor conocida como CodeBlocks Arduino Edition.

Gnoduino: es un IDE libre que intenta imitar en aspecto y funcionalidad al oficial, pero que evita su dependencia de Java, ya que está escrito en el lenguaje Python. Esto hace que sea especialmente ligero y rápido, aunque tiene la desventaja de funcionar solo en Linux (específicamente, en el escritorio Gnome).

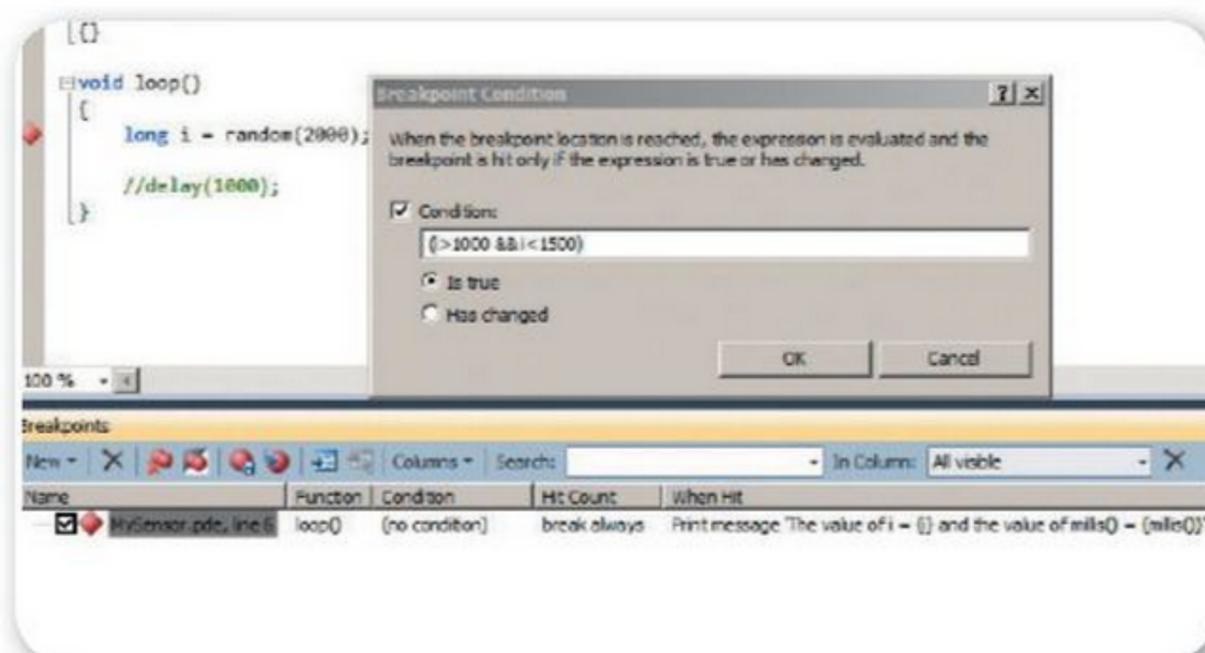


Figura 18. **Visual Micro** es un entorno de desarrollo opcional al oficial de Arduino, un plugin que permite utilizar Visual Basic como plataforma para los desarrollos de Arduino.

Codebender : en realidad es una aplicación web, por lo que funciona completamente online dentro del navegador, sin necesidad de realizar alguna instalación de software adicional. Incluye un completo editor de textos, un compilador y un cargador de sketches, todo sin salir del navegador. Una característica adicional que brinda es que se puede realizar previamente un registro gratuito para almacenar, en la nube, el conjunto de códigos que se realice.

Visual Micro : en realidad se trata de un plugin para utilizar el lenguaje Arduino dentro del entorno de programación Visual Studio de Microsoft. Una desventaja es que no funciona con la versión Express,

que se trata de una versión gratuita del programa, por lo que es necesario adquirir una versión superior.

EmbedXCode: al igual que Visual Micro, se trata de un plugin o complemento para el entorno de programación oficial de Apple, llamado XCode. Permite trabajar con un IDE “todo en uno”, de tal manera que se puede programar para distintas plataformas, como Arduino, ChipKIT, Maple, MSP430, Wirig, entre otros, con el lenguaje de programación propio de cada uno.

Existen también otros tipos de entornos de desarrollo para la plataforma Arduino, enfocados específicamente a la programación visual de los sketches, por ejemplo: en vez de utilizar instrucciones de texto en cierto lenguaje de programación, se utilizan estructuras visuales de colores para el acoplamiento gráfico que representan ciertas acciones y estructuras de control.

Su objetivo es facilitar a cualquier persona que no cuente con experiencia previa su iniciación en el mundo de la programación y al mismo tiempo, de la electrónica. Algunos de los más importantes son:

Scratch for Arduino: se trata de un plugin de entorno visual de programación Scratch para interactuar con placas Arduino.

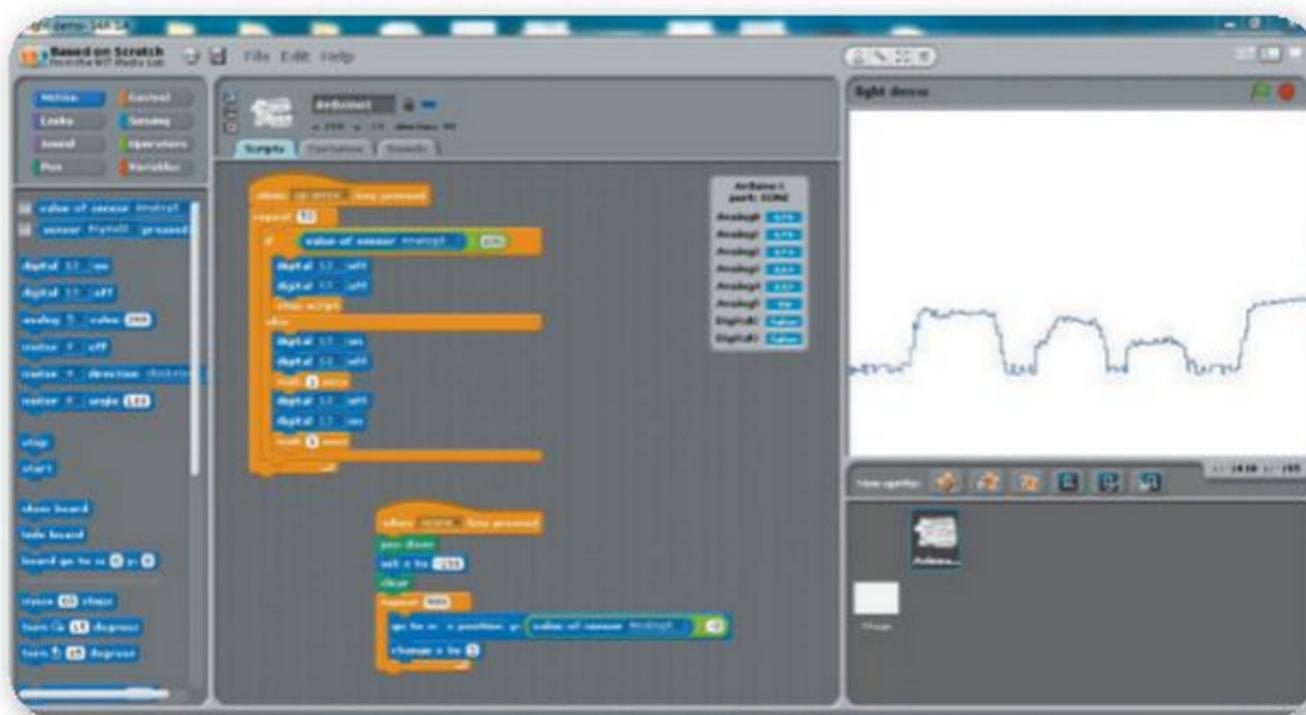


Figura 19. Scratch Arduino es un entorno de desarrollo gráfico, en donde se pueden utilizar elementos visuales para realizar procesos que se puedan cargar en la plataforma Arduino.

Modkit Micro : es similar a Scratch, con la ventaja de que puede utilizarse en más arquitecturas, además de las placas Arduino, y cuenta con la posibilidad de funcionar también online dentro del navegador.

Minibloq : al igual que Modkit Micro, es un entorno listo para ser usado en otras arquitecturas aparte de Arduino.

Ardublock: es un entorno de desarrollo visual para Arduino, programado en Java y multiplataforma.

Physical Etoys

Physical Etoys es una extensión libre y gratuita de Etoys, que es un lenguaje de programación orientado a objetos utilizado en el ámbito educativo. Fue desarrollado por Gonzalo Zavala, Ricardo Morán, Sebastián Blanco y Matías Teragni en la Universidad Abierta Interamericana.

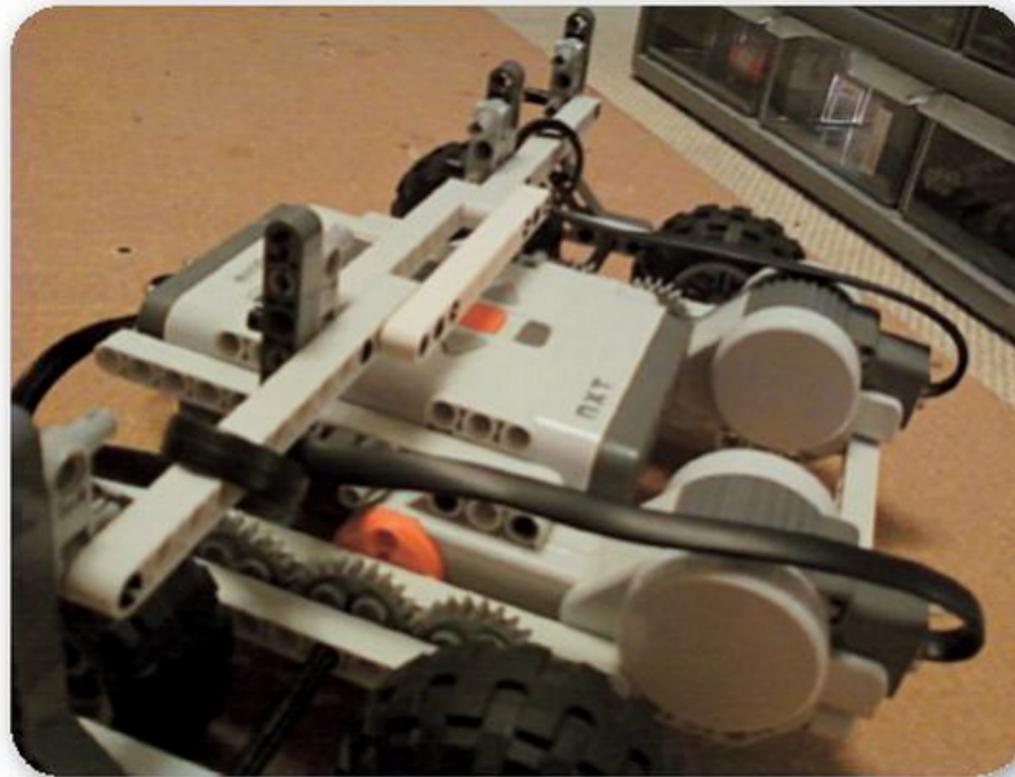


Figura 20. Aplicación de un Wiimote que utiliza Physical Etoys, para el control de un robot Lego NXT.

Al ser Physical Etoys una extensión, contiene la mayoría de las características de Etoys. Es una herramienta de programación visual que conecta el mundo virtual de las computadoras con el mundo real; esto quiere decir que se pueden programar distintos elementos físicos

para que realicen diferentes tareas, de acuerdo a cómo es programado. Un ejemplo conocido de esto serían los robots.

Como se explicó anteriormente, no es necesario tener experiencia previa en programación para utilizar Physical Etoys, ya que por medio de elementos visuales es posible realizar una estructura que represente un procedimiento o una serie de pasos que posteriormente se ejecutará en el elemento físico que contiene un microcontrolador.

Su filosofía es ayudar a los chicos a modelar y programar el mundo real para aprender más acerca de él. Permite que varios dispositivos electrónicos (Lego NXT, placas Arduino, Kinect, Wiimote, entre otros) sean programados de una manera sencilla y que interactúen entre sí gracias a su sistema de bloques, para brindar de esta manera un acercamiento al mundo de la programación y de la electrónica, con posibilidades de desarrollo de proyectos propios.

Physical Etoys hace que el aprendizaje de robótica sea sencillo y divertido en los distintos niveles de educación, que abarcan preescolar, primaria y secundaria. De esta manera, los niños pueden empezar a acercarse a la programación y adentrarse poco a poco para realizar proyectos cada vez más complejos.

A continuación, mencionamos algunas características por las que Physical Etoys constituye una experiencia amena y divertida para iniciarse en la programación:

Facilidad de aprendizaje : Physical Etoys reduce la complejidad que presenta el problema de la abstracción de código de los

PHYSICAL ETOYS
ES UNA EXTENSIÓN Y
CONTIENE CASI TODAS
LAS CARACTERÍSTICAS
DE ETOYS



ARDUINO COMO HARDWARE LIBRE



El hardware libre tiene similitudes con el software libre, ya que permite que los usuarios estudien su funcionamiento, lo modifiquen, lo mejoren o le quiten características. Para esto, se necesita que la comunidad tenga acceso a los diagramas y esquemáticos del hardware. Por eso, dentro de los ficheros viene incluida toda la información necesaria para realizar mejoras.

PHYSICAL ETOYS HACE
QUE EL APRENDIZAJE
DE ROBÓTICA
SEA SENCILLO
Y DIVERTIDO

lenguajes de programación, permite que el niño experimente con objetos virtuales muy parecidos a los reales y viceversa. También propone que, una vez que se entiende el concepto y se tienen las nociones de cómo funcionan todos los dispositivos con los que se prueba, se pueda profundizar el conocimiento para que cada vez se realicen proyectos de mayor complejidad y que añadan características más avanzadas.



Consonante con la transversalidad de

la tecnología: la robótica está naturalmente muy relacionada con diversas disciplinas científicas, como lógica, matemáticas, física, inteligencia artificial, biología, medicina y nanotecnología. Pero también es posible, gracias a la versatilidad que presentan los kits educativos de robótica, desarrollar dispositivos vinculados con las disciplinas más diversas, como música, artes plásticas, educación física, historia, geografía, literatura y otros. Physical Etoys hereda de Etoys todo el potencial educativo para que el niño pueda aplicar la computación en cualquier disciplina, y representar conceptos y fenómenos sin tener que esperar una materia de computación o tecnología específica. Además, se divide en módulos independientes que interactúan entre sí y cada uno representa un kit de robótica específico.

Aprendizaje con material concreto programable : trabajar con material concreto también contribuye a entrenar la motricidad fina y las capacidades cognitivas vinculadas a la experimentación física, el ensayo y el error, el manejo y la comprensión de las infinitas variables de valores continuos que nos presenta el mundo real. El trabajar con material concreto, además, permite afrontar desafíos



HERRAMIENTA INO



INO no es un entorno propiamente dicho, sino un programa ejecutable que compila y carga sketches Arduino. Estos sketches tienen que haber sido escritos y guardados previamente mediante un editor de textos. También incorpora una opción similar a **Serial Monitor** cuyos detalles de configuración se pueden especificar como parámetros del comando o bien en un archivo de configuración específico.

diversos que necesariamente deben encararse en equipo. El armado, la programación, el ordenamiento y cuidado de los materiales, la escritura del documento de trabajo exigen la participación coordinada de un equipo, con roles diferenciados pero integrados entre sí, que da sentido a un verdadero esquema de trabajo en grupo.

Accesibilidad : Physical Etoys es un software gratuito y de código abierto, por lo que se pueden realizar proyectos con una amplia gama de posibilidades, todo con la ventaja de que no hay que pagar por software especializado para obtener características que permitan la programación de los dispositivos electrónicos.

Los usos didácticos de Physical Etoys son muy variados, y uno de los más conocidos es el de los dispositivos electrónicos propietarios Lego. Estos bloques cuentan con una variedad de sensores y motores para llevar a cabo tareas específicas que los niños realizan en un entorno gráfico. Los niños pueden aprender a utilizar la lógica para realizar sus propios programas con tan solo arrastrar los elementos que realizarán determinada acción; de esta manera, pueden empezar a entender cómo es que deben estructurar un procedimiento para que su robot realice la tarea que quieren que lleve a cabo.

Existen también otros tipos de dispositivos Lego que integran más elementos y otros sensores para que se puedan llevar a cabo otras tareas, como, por ejemplo, los sensores de ultrasonido. De esta manera, se pueden realizar proyectos un tanto más complejos que con los bloques básicos.

Con la ayuda del entorno Physical Etoys, las posibilidades para el aprendizaje de la programación y un acercamiento a la electrónica son una realidad, ya que permite que todo aquel que carezca de conocimientos o experiencia previa en programación pueda empezar de una manera fácil, sencilla y amena, a interesarse en este mundo,



LEGO MINDSTORMS



Se trata de un juego de robótica para niños fabricado por **Legó**, que contiene elementos básicos de las teorías robóticas, como la unión de piezas y la programación de acciones, en forma interactiva. Este robot fue comercializado por primera vez en 1998. Puede ser usado para construir un modelo de sistema integrado con partes electromecánicas controladas por computadora.

y posteriormente a preparar y desarrollar proyectos más robustos, y, mejor aún, que se puedan llevar a cabo para que la tecnología avance y nos siga sorprendiendo.

Periféricos y referencias

Dentro del mundo de Arduino, existen placas que se diferencian de las que vimos anteriormente, que son un complemento de las placas originales y poseen funcionalidades que no vienen incluidas en el diseño original, ya que dependen del proyecto que se vaya a realizar.

Las placas de complemento son conocidas como shield, que significa “escudo” en inglés. Son circuitos impresos que van colocados encima de la placa Arduino original, de ahí proviene su nombre.

La conexión de los shields se realiza mediante una serie de pines incluidos en la placa, por lo que no es necesario llevar a cabo ningún otro tipo de conexión adicional mediante cables.

Como mencionamos anteriormente, la finalidad de estas placas es aumentar las capacidades de la placa Arduino, para que se puedan realizar distintas funciones. Todo esto de una manera más compacta, fiable y estable, ya que los shields son compatibles con el entorno Arduino y no es necesario realizar ningún tipo de conexión adicional, se acoplan perfectamente mediante los pines incluidos.

Los shields también pueden apilarse –siempre y cuando el shield inferior lo permita, y para esto debe contar con los pines para realizar dicha conexión– y ampliar aún más las capacidades de la placa Arduino original.



LEGO MINDSTORMS NXT

El bloque **NXT** es una versión mejorada de **Legó Mindstorms RCX**, que se considera la predecesora y precursora de los bloques programables de Legó. Legó vende la generación **NXT** en dos versiones: **Retail Version** y **Education Base Set**. El microcontrolador que posee es un **ARM7** de 32 bits que incluye 256 kB de memoria flash y 64 kB de RAM externa.

Los shields cuentan con una característica para compartir las líneas de conexión GND, 5V, RESET y AREF con las líneas de la placa Arduino y, además, tienen sus propios pines de entrada/salida para comunicarse. Por esta razón, si se les quisiera dar algún otro uso, no sería posible.

También hay que tomar en cuenta la alimentación de los shields, ya que la placa Arduino consume alrededor de 500 mA, por lo que la corriente restante es escasa para ellos. En ocasiones, podemos encontrar shields que consumen hasta 300 mA, una corriente considerable que no se alcanzaría a cubrir con la que deja libre la placa Arduino. Los que comúnmente gastan más energía son los que tienen pantallas LCD o conectividad Wi-Fi, entre otras características. Dentro de los shields oficiales, podemos encontrar los siguientes:

- **Ethernet Shield:** tiene la capacidad de brindar a la placa Arduino Uno la posibilidad de conectarse a una red TCP/IP. Si bien existe la placa Arduino Ethernet, que posee la misma funcionalidad, Ethernet Shield puede acoplarse a la placa Arduino Uno y ampliar sus capacidades. Como mencionamos anteriormente, cuenta con pines que encajan perfectamente en la placa Arduino y, por eso, utilizaremos las salidas y entradas ofrecidas por los pines hembra de este shield. De ser necesario, de acuerdo con los requerimientos del proyecto, es posible conectar un segundo shield encima de este.



Figura 21. El **Ethernet Shield** de Arduino permite la conexión de la placa a una red TCP/IP para obtener o enviar datos por medio de un cable Ethernet.

- **Wireless SD Shield** : permite que la placa Arduino Uno pueda comunicarse de manera inalámbrica con un módulo XBee (el módulo es aparte) o algún otro módulo que permita esta característica. Gracias a esto, se puede lograr un enlace con un dispositivo XBee a una distancia de hasta 100 metros en interiores y de hasta 300 metros en exteriores. De igual manera que sucede con los shields oficiales, la conexión es perfecta con la placa Arduino o sobre otro shield, así como también es posible montar otro shield. Además de las funcionalidades para añadir comunicación inalámbrica a la placa Arduino Uno, el shield también posee un zócalo para memorias de tipo micro SD, que se pueden utilizar para almacenar información, utilizando la librería de programación SD que viene en el lenguaje Arduino.
- **Wireless Proto Shield** : es exactamente igual al anterior, con la diferencia de que carece del zócalo para memorias micro SD.
- **WiFi Shield**: está pensado para dotar a la placa Arduino Uno de la capacidad de conectarse a una red TCP/IP de manera inalámbrica por medio del estándar Wi-Fi. Incorpora el chip HDG104 del fabricante H&D Wireless, que tiene una antena integrada para permitir la conexión a las redes Wi-Fi de tipo 802.11b y 802.11g. Las redes a las que se puede tener acceso por medio de este shield pueden ser tanto abiertas como protegidas mediante encriptación del tipo WEP o WPA2-Personal. Para realizar la gestión de este shield, es necesario utilizar la librería WIFI que viene por defecto en el lenguaje Arduino.
- **Motor Shield** : brinda a la placa Arduino la capacidad de control de motores, ya sea de hasta dos motores de CD. En este caso, permite el control de la velocidad y el sentido de giro, o también de un motor paso a paso. También es posible realizar las mediciones de las capacidades de los motores controlados.



SHIELDS NO OFICIALES



Existen shields no oficiales, diseñados y contruidos por la comunidad de Arduino. Según los requerimientos del proyecto, decidiremos qué tipo de shield vamos a utilizar. En **Internet Shieldlist** (www.shieldlist.org), ha y una lista de todos los shields existentes, ordenados por fabricante, con sus características, conexiones y la información necesaria para entender cómo funcionan.

- **Proto Shield** : permite realizar un nuevo circuito, ya que es fácil de diseñar e implementar. Cuenta con un área de trabajo para soldar los componentes necesarios para realizar un proyecto de shield propio.



Figura 22. WiFi Shield brinda la posibilidad de que la placa se conecte a una red TCP/IP de manera inalámbrica, para la transmisión y recepción de información.

Además de todos los shields anteriormente comentados, existen otros que no son oficiales, realizados por la comunidad de Arduino para dotar de características adicionales a las placas. La cantidad



MÓDULOS XBEE

Los módulos XBee dotan a la placa Arduino de comunicación inalámbrica de otros dispositivos y permiten compartirla con otros, que también tengan el módulo XBee. Brindan dos formas amigables de comunicación: la transmisión en serie transparente (modo AT) y el modo API, que posee muchas ventajas. Pueden configurarse desde la PC utilizando el programa X-CTU, o bien desde un microcontrolador.

WIFI SHIELD CONECTA
UNA PLACA ARDUINO
UNO A UNA RED
TCP/IP DE MANERA
INALÁMBRICA



va en aumento, ya que cada integrante de la comunidad puede hacer su aporte.

Para realizar un shield propio, lo recomendable es utilizar como base los ficheros de Eagle (se encuentran en la página oficial de Arduino) para realizar un esquemático del shield y llevarlo a cabo.

Debemos tomar en cuenta que los pines 7 y 8 no son el estándar de 0.1". Además, existen otros sitios web de donde podemos obtener

recursos para realizar un placa shield propia que sea compatible con el entorno de Arduino.

Existe una gran ventaja y es que Arduino fue concebido como hardware abierto, por lo que es posible realizar desde una placa shield para Arduino hasta la placa completa, ya que todo el material se encuentra disponible en la página oficial y en otras páginas de la comunidad.

Referencias y recursos

No importa si carecemos de experiencia previa en el manejo de las placas Arduino: la página oficial ofrece mucha información sobre los primeros pasos, una introducción al lenguaje de programación, así como también las características técnicas de cada una de las placas que se pueden conseguir ensambladas y listas para su compra.

También encontramos toda la información relacionada con las librerías que maneja el lenguaje de programación de Arduino, junto con los archivos necesarios y los pasos a seguir para la instalación del entorno de desarrollo en cada uno de los sistemas operativos más utilizados.

Además, contamos con las instrucciones necesarias para la realización de una placa Arduino, construida con materiales propios. Esto es posible gracias a los ficheros incluidos.

Además, la página oficial de Arduino cuenta con un enlace a una página llamada Playground, en la que se puede encontrar información sobre las contribuciones que la comunidad Arduino realiza. Estas contribuciones abarcan desde las nuevas librerías para utilizar en el

lenguaje de programación de Arduino hasta placas shields ensambladas junto con sus ficheros para realizar una copia idéntica por medio de programas de diseño de circuitos.

The screenshot shows the Arduino Playground website. The header is teal with the Arduino logo on the left and a search bar on the right. Below the header is a navigation menu with links for Home, Buy, Download, Products, Learning, Reference, Support, and Blog. The main content area is titled "The Arduino Playground" and contains a welcome message: "Welcome to the Arduino Playground, a wiki where all the users of Arduino can contribute and benefit from their collective research." It also includes a description of the site as a place to post and share code, circuit diagrams, tutorials, and DIY instructions. A "Playground Content Tree" is listed, including links for Manuals and Curriculum, Arduino StackExchange, Board Setup and Configuration, Development Tools, Arduino on other Atmel Chips, Interfacing With Hardware (with sub-links for Output, Input, User Interface, Storage, Communication, Power supplies, and General), Interfacing with Software, User Code Library (with sub-links for Snippets and Sketches, Libraries, and Tutorials), Suggestions & Bugs, Electronics Technique, Sources for Electronic Parts, and Related Hardware and Initiatives. The page also mentions "Arduino Playground is a work in progress" and encourages users to read the "Participate" section.

Figura 23. La sección **Playground** del sitio de Arduino cuenta con muchísimos recursos para el entorno.



RESUMEN

En este capítulo conocimos cómo nació la plataforma de desarrollo Arduino, cuáles son sus principales características y las innumerables aplicaciones que tiene en los campos de la electrónica, de la automatización y de la telemetría. Además, accedimos a los conceptos básicos de Arduino, el entorno de desarrollo IDE, otros IDEs alternativos y los Physical Etoys.

Actividades

TEST DE AUTOEVALUACIÓN

- 1 ¿Cuáles son las principales características de la placa **Arduino** ?
- 2 ¿Cómo son el funcionamiento y el entorno de desarrollo general de todos los **modelos existentes** ?
- 3 ¿Qué son y para qué sirven las **librerías** ?
- 4 ¿Qué características tienen los lenguajes de **alto, medio y bajo nivel**?
- 5 ¿Cuáles son los **lenguajes de programación** que se pueden utilizar?
- 6 ¿Qué significan las siglas **IDE**?
- 7 ¿Cuáles son los **IDEs alternativos** que podemos identificar?
- 8 ¿Cuáles son las ventajas de **Physical Etoys** ?
- 9 ¿Qué son y para qué sirven las **placas de complementos** o **shields**?
- 10 ¿Cuáles son los **microcontroladores** más usados en Arduino?



PROFESOR EN LÍNEA



Si tiene alguna consulta técnica relacionada con el contenido, puede contactarse con nuestros expertos: profesor@redusers.com



Proyecto: sistema de telemetría con Arduino

Presentamos un proyecto acorde a los conocimientos presentados en los capítulos anteriores, para que el lector pueda contar con más herramientas e ideas para dejar volar su imaginación y poder crear a partir de sus propios saberes e intereses.

▼ Telemetría con Arduino 100	¿Qué vamos a usar? 120
Telemetría 102	Armado y explicación
Telemetría con Arduino..... 103	del código de Arduino 125
Aplicaciones de los sistemas	Explicación del código final 129
de telemetría 106	
Magnitudes físicas medibles	▼ Limitaciones y otras
con Arduino..... 110	alternativas 136
	Alternativas con otras plataformas ... 140
▼ Implementar el proyecto:	Conclusiones 144
formas de comunicación	
y transferencia 115	▼ Resumen..... 145
Componentes del sistema	
de telemetría 119	▼ Actividades..... 146



Telemetría con Arduino

En este capítulo, trataremos de desarrollar un proyecto uniendo los conceptos de Arduino y de electrónica, y enfocándonos principalmente en una aplicación particular de la telemetría.

Comenzamos con una definición etimológica de la palabra **telemetría**, que viene de las palabras griegas $\tau\eta\lambda\epsilon$ (tele), que quiere decir “a distancia”, y $\mu\epsilon\tau\rho\omicron\nu$ (metron), que quiere decir “medida”; en conjunto, “medir a distancia”.

En otras palabras: la telemetría es la **medición** de cualquier magnitud física y el posterior **envío** de esa información a través de un canal de **comunicación**. Básicamente, en todo sistema de telemetría se distinguen al menos tres partes: la magnitud física por medir, un canal de comunicación y una unidad de control.



Figura 1. Esquema básico de comunicación en un sistema de telemetría con punto medio (servidor).

En este capítulo, veremos cuáles son y cómo se pueden medir las distintas **magnitudes físicas** como origen de información para nuestro sistema de telemetría. Si bien más adelante mostraremos un proyecto en particular, la idea es que el lector tenga de manera general un pantallazo de todas las variantes que puede implementar con Arduino para crear y diseñar su propio sistema de telemetría, ya sea como pasatiempo o para una aplicación profesional. La elección de la plataforma Arduino para esta ocasión no es arbitraria, sino que se basa en el hecho de que al ser **hardware libre**, existe una gran comunidad detrás, y esto permite rápidamente obtener nuevos conocimientos ayudándonos de la experiencia compartida por otras personas.

Además, esta herramienta es **abierta y flexible** tecnológicamente; algo necesario y útil en lo que respecta a la telemetría.

Tenemos un abanico de posibilidades de diseño muy amplio por dos motivos. En primer lugar, porque son muchas las magnitudes físicas que se pueden medir, como, por ejemplo: temperatura, presión, posición, velocidad, distancia, sonido o voz, luz, campo magnético, etcétera. En segundo lugar, este conjunto se combina con otra buena cantidad de medios de transmisión, que dependerán de las distintas **tecnologías** por implementar. Entre ellas, podemos nombrar Wi-Fi, Bluetooth, radiofrecuencia, NFC, Ethernet, XBee, etcétera.

En particular, daremos una explicación paso a paso acerca de cómo transmitir a través de la red de telefonía móvil **GSM** una posición medida con **GPS**, un proyecto interesante por la unión de ambas tecnologías. Aunque Arduino tiene bastantes ventajas frente a otras plataformas, posee algunas limitaciones en ciertos aspectos. Para esas ocasiones, daremos algunas variantes y explicaremos también otras plataformas que conviven hoy en día, con sus ventajas y desventajas.

Al final del capítulo, podremos leer una conclusión de este proyecto y algunas palabras de cierre.

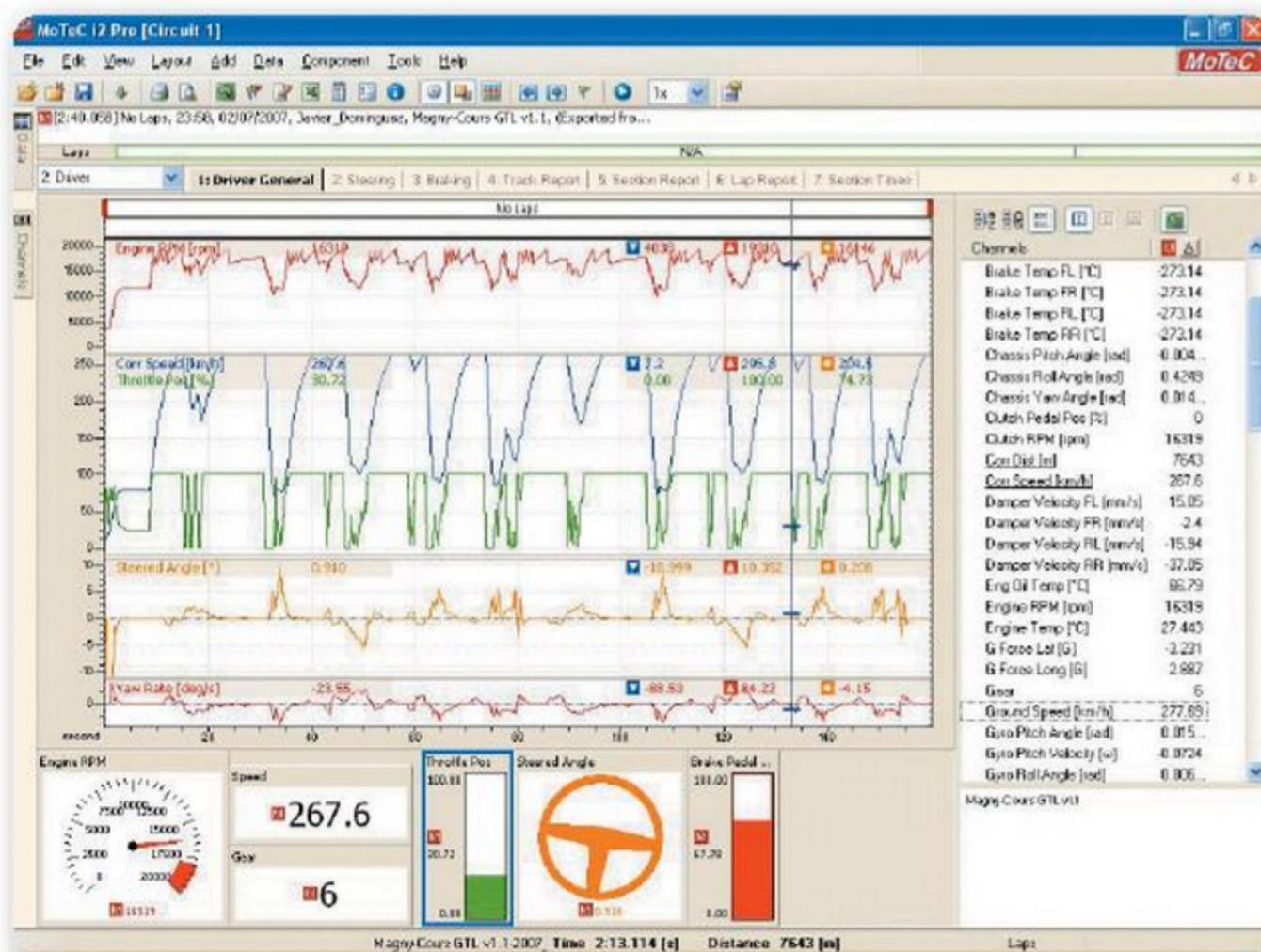


Figura 2. Software de telemetría aplicado a motos GP.

Telemetría

Las **magnitudes físicas** medibles son muchas, entre las que podemos nombrar: temperatura, presión, posición y velocidades relativas, aceleración en distintos ejes, humedad. Para medirlas, es posible utilizar cualquier sensor o transductor. Más adelante, hablaremos particularmente de los distintos módulos que podemos usar con Arduino y cómo logran transformar esa información física en una variable para controlar en nuestro programa.

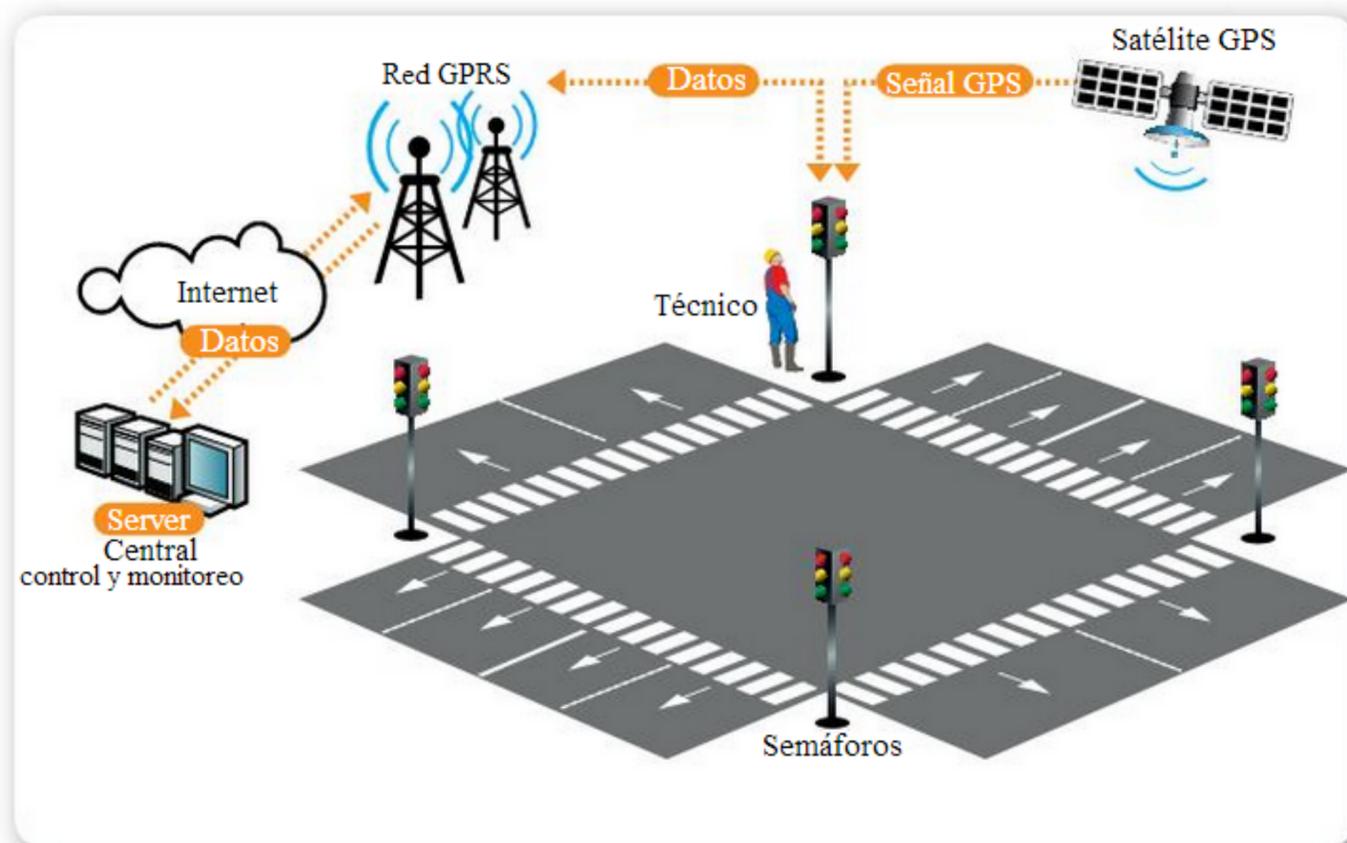


Figura 3. Diagrama de un sistema de telemetría aplicado a un sistema de control de tránsito.

El **canal** donde se transmite la información también es muy variado y dependerá de la aplicación particular que queramos desarrollar, pero podemos clasificarlo en dos grupos:



PRIMEROS PASOS

Los primeros circuitos de transmisión de datos datan de 1845. En 1874, ingenieros franceses construyeron un **sistema de sensores de tiempo y nieve profunda** sobre el Mont Blanc. En 1901, se patentó el **selsyn**, un circuito para el envío de información de rotación sincronizada a distancia. En 1906, un conjunto de estaciones sísmicas fueron construidas en el Observatorio Pulkovo de Rusia.

Cableado: los sistemas de telemetría a través de cable fueron los primeros en aparecer. Se usan cuando no se necesita transferir información en grandes distancias o cuando se requiere una alta velocidad y fidelidad de transmisión. La necesidad de unión del sistema de control y la magnitud medible mediante un cable o varios puede resultar accesoria, en principio, si se desea, por ejemplo, medir la posición de un auto y transmitirla. No obstante, puede ser útil si se necesita medir con estabilidad la rotura de vidrios para una alarma.

Inalámbrico : en la actualidad, los sistemas de telemetría modernos se inclinan más hacia la transmisión inalámbrica, por varios motivos. En primer lugar, hay aplicaciones particulares en las que resulta totalmente impracticable usar cables para transmitir la información, como en el ejemplo de la medición de la posición de un auto o avión que nombramos antes. Aquí, la necesidad de transmisión a través de **ondas electromagnéticas** resulta obvia. Pero también, por otro lado, gracias a los avances tecnológicos se ha logrado alcanzar muy buena estabilidad y velocidad en la transferencia de datos inalámbricos a bajo costo. Además, con la facilidad de implementación multiplataforma, hoy no resulta difícil desarrollar sistemas donde se usen tecnologías como Wi-Fi, GPS o 3G para un fin específico y que se puedan monitorear desde un teléfono móvil, una tableta o computadora personal.

La **unidad de control** es la encargada de recibir la información transmitida y realizar las acciones pertinentes. Dependiendo de la aplicación que se comande en el sistema de telemetría, puede que se necesite, en algunos casos, medir alguna magnitud y mostrarla en un display o pantalla, aunque en aplicaciones un poco más sofisticadas los datos no solo se muestran, sino que pueden ser utilizados para controlar algún dispositivo externo, haciendo que la etapa de telemetría sea el comienzo de otras etapas subsiguientes. Incluso puede resultar necesario que la unidad de control vuelva a emitir datos al origen y se genere una comunicación ida y vuelta (comunicación dúplex).

Telemetría con Arduino

Este proyecto apunta a entender y crear un sistema de telemetría con la plataforma Arduino. Básicamente, realizaremos las tres etapas que describimos a continuación.

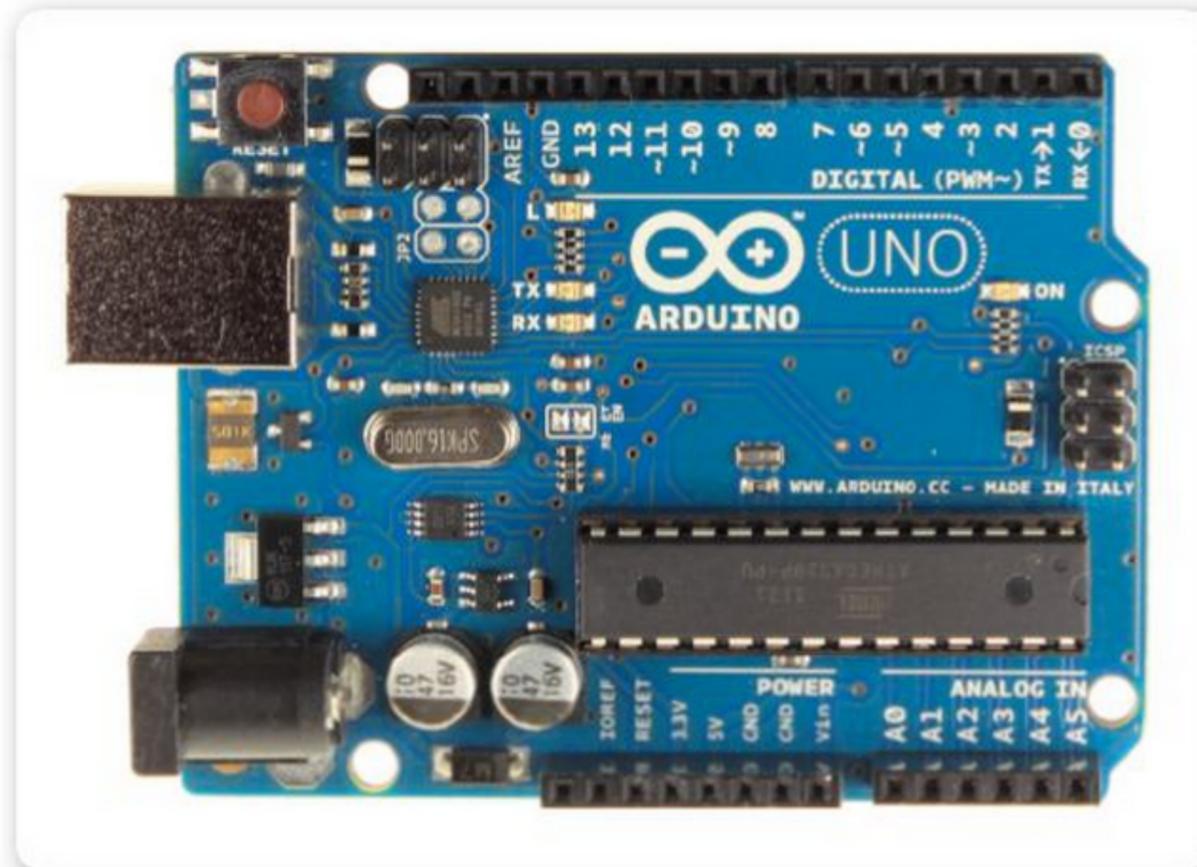


Figura 4. En esta imagen vemos una placa Arduino UNO Rev. 3.

En la primera etapa, la **medición de la magnitud física** se realizará a través de algún **transductor**, como aclaramos antes, y esta información se introducirá a la Arduino a través de algunos de los puertos o pines de entrada. Los modos en los que se pueden ingresar los datos son:

Puertos analógicos: estos puertos conversores analógico-digitales son empleados por la Arduino para transformar variaciones de tensión entre $\langle 0 \text{ y } 5 \text{ V} \rangle$ en un registro de 10 bits, es decir, un número que tomará valores entre $\langle 0 \text{ y } 2^{10}-1 = 1023 \rangle$.

Puertos digitales: algunos módulos incorporan los datos a la Arduino mediante un **conjunto de bits** que deberán ser leídos según la codificación especificada por el fabricante.

Algunos de los shields más avanzados, como el de GPS, ya vienen estandarizados para usar los protocolos de comunicación que incorpora la Arduino (UART, SPI, I2C).

LA PLATAFORMA
ARDUINO ES IDEAL
PARA LA TELEMETRÍA
POR SU FLEXIBILIDAD
TECNOLÓGICA

La segunda etapa será la del procesamiento de esa información numérica.

Puede ser que se necesite recopilar varias muestras y calcular promedios o combinarlas con otras variables que

representen datos de otras magnitudes, etcétera. Se trabaja con las distintas variables que entraron a la Arduino y se crea un programa que realice alguna aplicación particular.

La última etapa es la del **envío de la información procesada**. Para esto, la Arduino se comunicará con algún otro dispositivo que sea capaz de transmitir la información con la tecnología adecuada. La **tecnología de salida** utilizada dependerá de la aplicación; por ejemplo, Wi-Fi tiene buena velocidad pero la distancia no deberá ser mucha, en cambio, si se transfiere mediante GPRS, la velocidad es mucho menor pero se puede llegar a prácticamente todo el país. La comunicación entre la Arduino y este dispositivo se realizará mediante UART, SPI o I2C, como más adelante detallaremos.

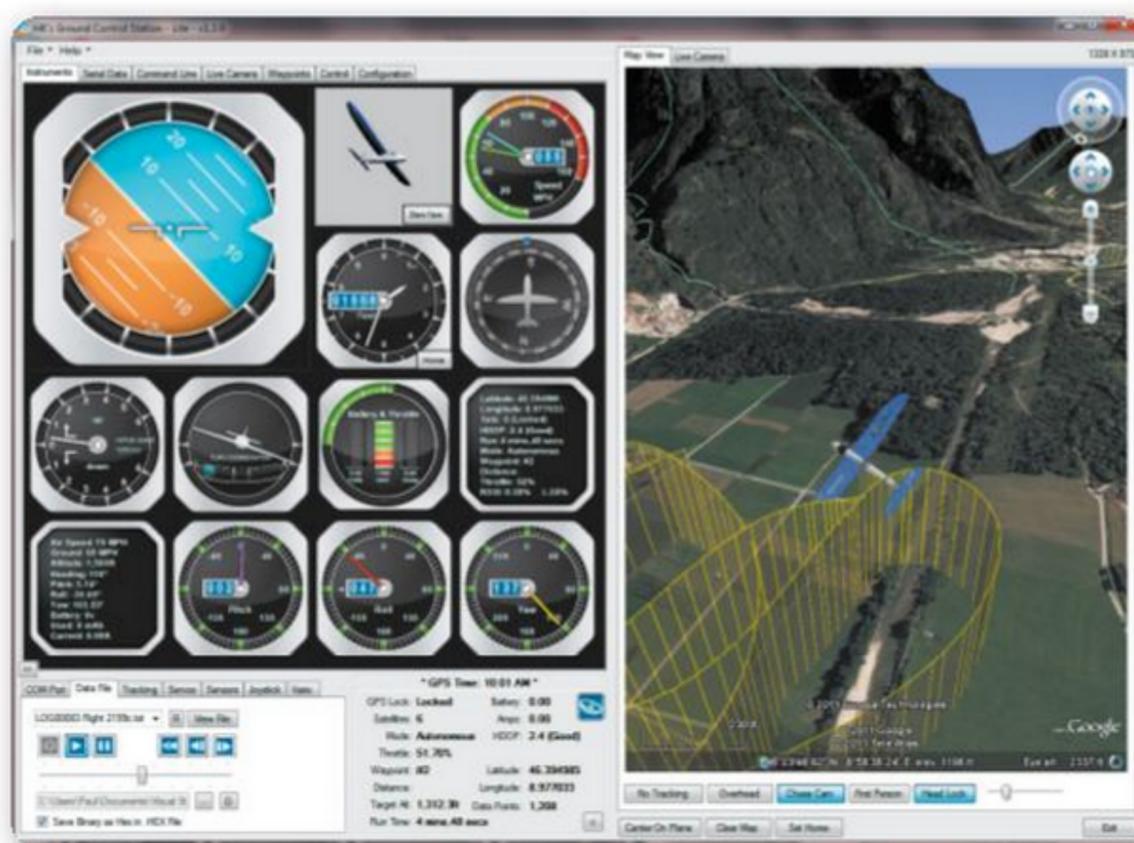


Figura 5. Software utilizado para monitorizar y simular telemetría en un avión RC.



DISTINTOS TIPOS DE ARDUINO

Existen distintos tipos de Arduino (**boards**) que incorporan gran variedad de tamaño y cantidad de pines. **Arduino MEGA 2560** incorpora mayor cantidad de puertos de entrada/salida y de comunicación, que pueden ser muy útiles cuando se necesita usar varios al mismo tiempo. Se debe tener cuidado, porque al tener una distribución de pines diferente, **algunos shields pueden ser incompatibles**.

Aplicaciones de los sistemas de telemetría

Las aplicaciones que utilizan los beneficios de la telemetría son muy diversas y nos rodean frecuentemente en nuestro **ambiente cotidiano**. Por nombrar un caso, es probable que el lector tenga o conozca a alguien que posea alarma en la casa; en ese caso, su sistema de monitoreo es un claro ejemplo de telemetría.



Figura 6. Uno de los kits de alarma que podemos encontrar en el mercado. Este modelo posee particularmente la función de reportar su estado vía GSM.

Los sistemas de **alarmas** tradicionales incluyen detectores de movimiento, sensores para las roturas de vidrio, entre otros, cuyo funcionamiento detallaremos en la siguiente sección.

La empresa que se encarga del monitoreo de la seguridad de una casa recibe la lectura de todos los sensores de manera constante y, cuando alguno se dispara, más allá del funcionamiento de la propia alarma (se dispara la sirena, emite o no avisos personalizados, etcétera), obtiene esta información mediante la línea telefónica, o bien, a través de un módulo GSM (que utilizaremos en nuestro proyecto, dado que le permite una autonomía superior al sistema). Entonces, ahí tenemos un ejemplo claro de la telemetría. Dependiendo de la complejidad del sistema, a través de una conexión de datos, como

la GPRS que nos proporciona el módulo de GSM, podríamos recibir las lecturas de los sensores de nuestra casa en nuestro smartphone.

Otra aplicación típica, siguiendo con las alarmas, la encontramos en los **vehículos**, donde el sistema nos informa lecturas similares a las de una casa, pero, además, nos proporciona la ubicación del vehículo mediante un **módulo GPS** (que también utilizaremos en nuestro proyecto) para conocerla en caso de robo o para algún distraído que olvidó donde lo estacionó.

Saliendo un poco de las aplicaciones más cotidianas, en el **ámbito científico** e industrial también nos encontramos con un uso intensivo de esta herramienta.

LA TELEMETRÍA
SE ENCUENTRA
PRESENTE EN
NUESTRAS CASAS Y EN
EL PLANETA MARTE



Figura 7. En esta imagen observamos un sistema industrial de adquisición de datos utilizado para realizar ensayos dinámicos y estáticos.

Algunos habrán escuchado hablar de la misión espacial **Curiosity**, en la que se aterrizó, sobre la superficie de Marte, un vehículo explorador destinado a enviar información sobre este planeta. El aterrizaje se logró

mediante muestras de suelo, imágenes y decenas de sensores específicos para la tarea, sumados a otros sensores, por ejemplo, para asegurar que la trayectoria de viaje sea la correcta o que el descenso sea a la velocidad adecuada. Estos son claros modelos de telemetría.

En este tipo de aplicaciones, el uso de la telemetría es indispensable, ya que las distancias son enormes y la capacidad de estar físicamente para corroborar los datos es, en algunos casos, imposible y, en otros, limitada por los elevados costos de realización. No solo hablamos de misiones de exploración espacial; también encontramos ejemplos de telemetría en los numerosos **satélites** que orbitan por nuestro planeta y nos permiten estar comunicados.



Figura 8. Observamos al **Curiosity** junto a otros rovers menores. Notemos su tamaño en comparación con los científicos.

Para mencionar más ejemplos que dejen esto en evidencia, podemos proponer los **aviones**, un sistema de transporte convencional para esta época, que se nutre de muchas lecturas de medición electrónica y las transmite a su base de control, la cual le permite, además, en determinados momentos, adoptar un **vuelo automático**, controlado por computadora a partir de estas lecturas. Profundizando un poco más en esto, existen también los llamados **drones**, vehículos aéreos no tripulados, que se manejan remotamente o de forma autónoma mediante una programación de vuelo previa.

En esta programación es imposible pensar en un modelo ideal donde las variables –por ejemplo, climáticas– sean constantes o presuponer que todos los demás sistemas (propulsión, dirección, etcétera) van a funcionar perfectamente. Por eso, debemos incluir en el algoritmo el accionar a partir de alguna falla o ante una determinada variación en el ambiente donde se desarrolle.

Para esto, es vital la medición electrónica permanente de todas las magnitudes que le permiten tanto a la computadora como al piloto remoto actuar en base a ello. De no existir la posibilidad de obtener estos valores en tiempo real en la base de control (o donde sea que se reporten), este tipo de prácticas sería imposible.

TODO PROCESO
AUTOMATIZADO
UTILIZA LA
TELEMETRÍA PARA
REPORTAR SU ESTADO



Figura 9. El Altair UAV es un drone adaptado por la NASA para ser utilizado como herramienta para la investigación.

Como último ejemplo donde la telemetría no toma un rol tan protagónico, sino que más bien es utilizada como una herramienta de soporte sin restar importancia, tenemos la **Fórmula Uno**.

Los vehículos que compiten en esta categoría poseen más de doscientos sensores, los cuales reportan al equipo del conductor

miles de lecturas por segundo que se traducen en valores tangibles, como el rendimiento del motor, el desgaste de los frenos, la adherencia de los neumáticos, y otros más puntuales, por ejemplo, si el acelerador está presionado o no y, si es así, cuán presionado está. Este sistema dispone hoy de una alta complejidad y le permite al equipo visualizar todos los detalles, tanto numéricamente como mediante gráficos, para luego tomar decisiones y, de esta manera, asistir al conductor. Tanto es así, que hoy se pueden controlar todos los parámetros del vehículo sin moverse de los boxes.



Figura 10. El equipo de F1 estudiando los datos obtenidos mediante telemetría. Se pueden observar los gráficos de la PC de uno de los operadores.

Magnitudes físicas medibles con Arduino

Conociendo entonces ya más sobre la telemetría y sus aplicaciones, vamos a hablar sobre cuáles son las magnitudes que podemos medir con Arduino y cómo se implementan en esta plataforma.

Arduino es, como sabemos, una plataforma de desarrollo ampliamente difundida y utilizada; gracias a esto, la implementación

en el sistema de hardware para medir la magnitud que deseemos se realiza fácilmente mediante módulos o shields (estos últimos son más completos, complejos y caros). Los shields o módulos pueden fabricarse manualmente utilizando nuestros propios diseños –tarea en la que adquiriremos nuevos conocimientos–, pero comprarlos simplifica mucho la tarea y los tiempos de avance en proyectos particulares.

Básicamente, podemos introducir las mediciones de los distintos tipos de magnitudes físicas que necesitamos medir de dos maneras distintas:

La forma más económica y manual es **crear el circuito nosotros mismos** y, mediante un transductor que cambie la resistencia o la tensión de salida con alguna variación de la magnitud física por medir, incorporarlo a la Arduino mediante los puertos de **conversión analógica-digital (ADC)**

Diseñar un circuito de esta manera permite tener la libertad de poder **elegir y adaptar el transductor** que deseemos y lograr el equilibrio precio/calidad que nos resulte adecuado.

La segunda opción es, en lugar de comprar y diseñar un circuito para usar con el ADC de Arduino, **utilizar los módulos y shields comerciales**. El beneficio de usar estos módulos es que, además de no ser necesario diseñar el circuito, son de fácil implementación. Se conectan a la Arduino, a los puertos ADC, a los puertos digitales o a los pines de comunicación serie. Suelen venir acompañados por **librerías y hojas de datos** que nos dan la información de conexión e implementación necesaria.

COMPRAR CHIPS
Y DISEÑAR CADA
SHIELD ES MÁS
ECONÓMICO, PERO
MUY TEDIOSO



TELEMETRÍA CON FINES MILITARES



Los misiles guiados poseen sensores según el tipo de objetivo. Existen los **pasivos**, en los cuales el misil busca su objetivo; los **semiactivos**, donde el sistema de localización del misil se dirige hacia una marca emitida; los **activos**, en los que el misil selecciona su objetivo; y los **combinados**, donde la computadora de la plataforma de lanzamiento calcula las coordenadas según la información de los sensores del misil.



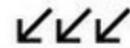
Figura 11. En esta imagen se ve el módulo de temperatura y humedad **DHT22**. Notemos que, a pesar de que aparecen cuatro pines, solo tres están conectados.

A continuación, conoceremos algunos de estos módulos y shields comerciales:

Temperatura: empezamos con una magnitud clásica. Medir la temperatura es fácil, a priori. Pero, cuanto mayor exactitud busquemos, más complejo será el sistema, aunque esto se da en toda magnitud en general. Con solo incluir un integrado –el LM35– al sistema podremos, de un modo simple, medir temperatura analógicamente mediante `analogRead`, por lo que, en este caso, no sería necesario utilizar un shield. Sin embargo, existe un módulo que, además de temperatura, mide humedad, el **DHT22**. Con solo tres pines (GND, VDD y Data), este módulo le transmite digitalmente a la Arduino, a partir del pin Data, los



TENSIONES DE REFERENCIA



Debemos tener en cuenta que Arduino está diseñada para medir tensiones entre 0 y 5 V; por lo tanto, el circuito que vamos a diseñar debería estar formado por algún divisor de tensión que modifique la tensión que disminuya en una resistencia entre el puerto ADC y GND (tierra) en ese rango específico. Si no se puede trabajar en ese rango, otra opción es utilizar la función `AnalogReference` que, mediante código, permite ajustar otra referencia de tensión.

valores correspondientes de cada magnitud y, además, por ser digital, permite definir otros parámetros (como un estado de bajo consumo) utilizando las librerías DHT existentes en el compilador de Arduino.

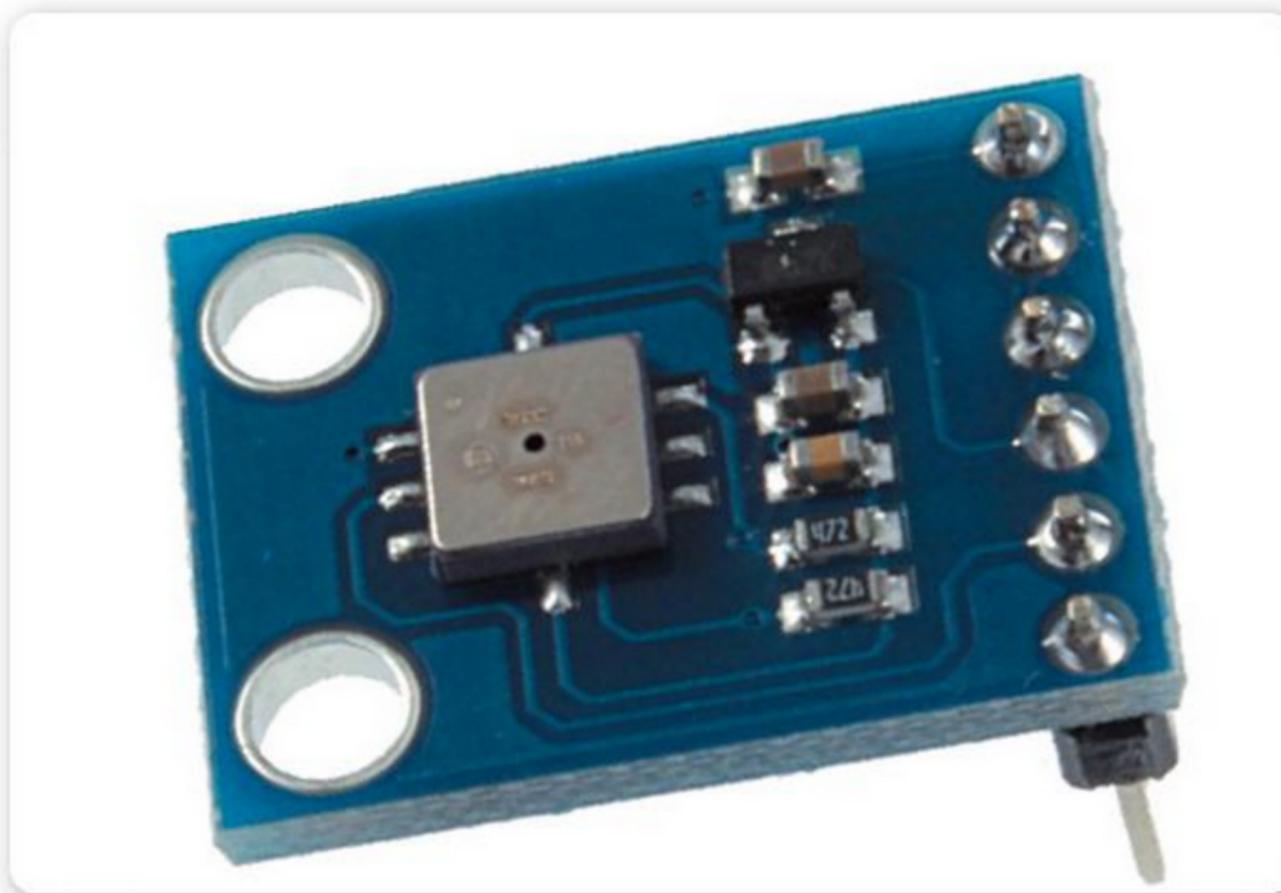


Figura 12. Este es uno de los módulos que podemos encontrar que trabajan con el **BMP085** (integrado metálico con un orificio en el centro).

Presión : para añadir una magnitud ambiental, podemos utilizar uno de los módulos basados en el **BMP085**. Este es un sensor digital de bajo consumo y alta precisión que, a través de una comunicación I2C (con la librería `wire`), nos envía las lecturas de esta magnitud y, además, la temperatura –si así lo queremos–, ya que para lograr medir la presión necesita este valor y tiene un termómetro embebido. Este módulo también posee sus librerías en el IDE de Arduino con el mismo nombre del integrado: `bmp085`.

Detector de movimiento: corriéndonos un poco de las magnitudes físicas, ahora evaluamos los **PIR** (*Passive Infrared Sensor*) o sensores infrarrojos pasivos. Estos se centran en un sensor infrarrojo que detecta las variaciones en esta frecuencia cuando un cuerpo radiante (personas, por ejemplo) entra en su rango. Cuando esto sucede, establece el pin Out en uno lógico (HIGH) y, de esta manera, reporta su estado a la Arduino, que actuará o no según el algoritmo.

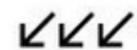


Figura 13. Pueden verse los dos piezoeléctricos del **HC-SR04** , junto con el cristal de 4 MHz que determina la frecuencia del pulso sonoro.

Ultrasonido: para medir distancias relativamente grandes, la tecnología que generalmente se usa es el ultrasonido. Arduino cuenta con varios módulos que utilizan esta técnica, uno de ellos es el HC-SR04. Mediante dos piezoeléctricos, uno que emite el pulso sonoro y otro que escucha, se calcula el tiempo en que el sonido emitido vuelve a ser detectado y, de esta manera (conociendo la velocidad del sonido que es aproximadamente 343 m/s), calcula la distancia. Este módulo cuenta con cuatro pines, de los cuales dos son para su alimentación (VCC y GND) y otros dos para el control de los piezoeléctricos: Trig es la entrada para la señal por reproducir y Echo es la salida por donde vuelve a detectarse una vez que rebotó en algún obstáculo.



SHIELD GPS



Un shield muy interesante para la telemetría es la medición de una posición mediante **GPS**. La mayoría incorpora un socket de tarjeta mini SD para almacenar los datos. Existen varios fabricantes de este tipo de shields. Aquí usaremos un chip Fastrax UP501 con soporte de WAAS y EGNOS, con salida de datos en forma NMEA. Todos estos chips se comunican a la Arduino mediante **UART**.



Implementar el proyecto: formas de comunicación y transferencia

Los módulos más económicos para el **envío y recepción de datos** son los RF de entre 315 y 433.92 MHz. La frecuencia es ajustable, consumen muy poca corriente y se pueden alimentar con la tensión de la Arduino.

Básicamente, toman la señal desde el puerto serie de la Arduino (ya sea UART o SPI) y emplean **modulación ASK** (*Amplitud Shift Keying*, modulación por desplazamiento de amplitud), en donde se representan los datos digitales como variaciones de amplitud de una onda portadora en función de los datos por enviar desde el emisor hasta el receptor. La Arduino que tiene conectado el módulo receptor, demodula la onda y transfiere los datos nuevamente a través del puerto serie.

La velocidad que manejan es baja (4 kbits/s) por la baja frecuencia de la onda portadora. Dependiendo de la tensión de alimentación, se puede llegar a cubrir **distancias de entre 50 y 200 metros**, lo que, por el bajo costo de los módulos, puede ser una variante muy interesante en cualquier sistema de telemetría.

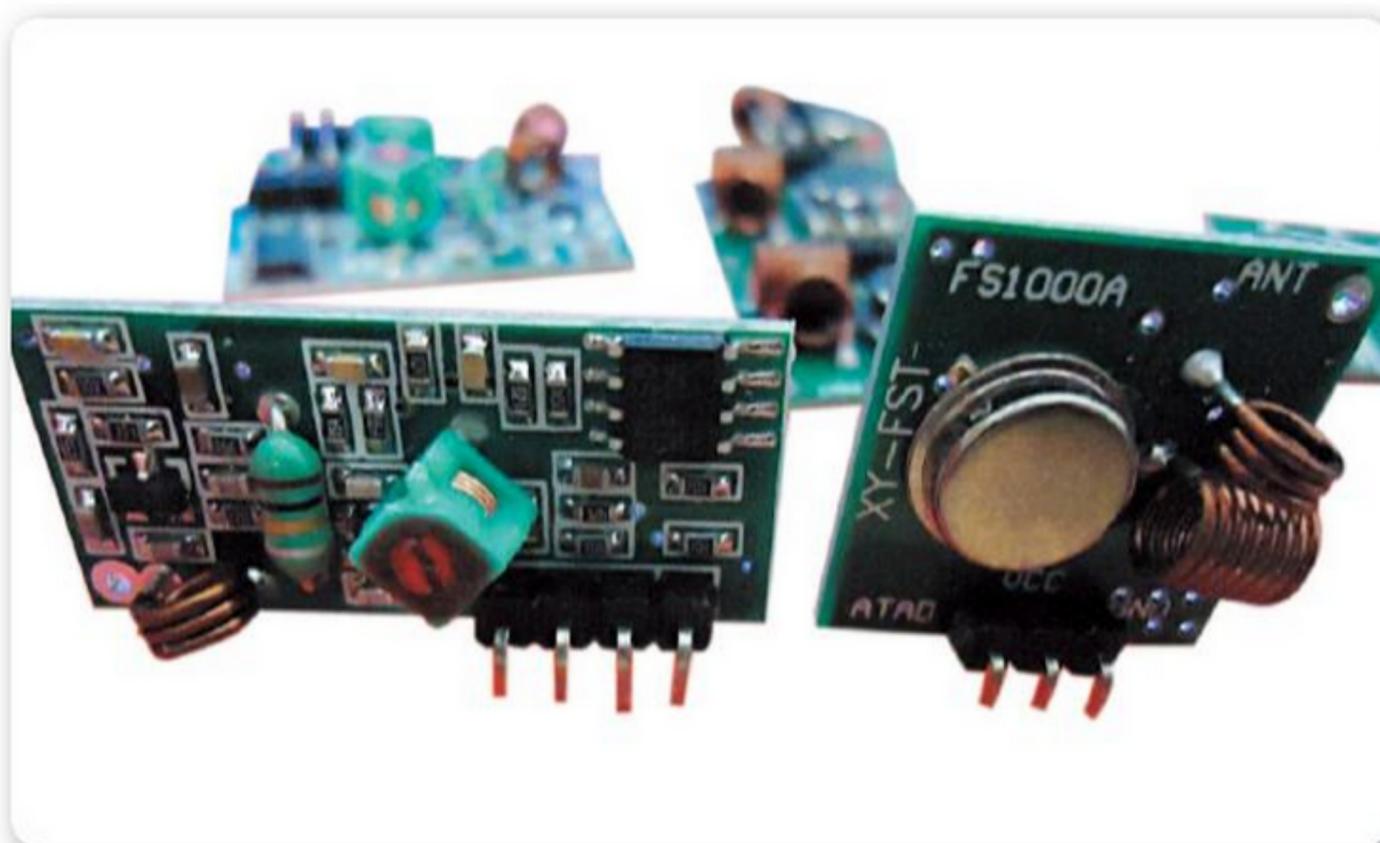


Figura 14. Módulos de RF para Arduino:
receptor XY-RF-5V (izquierda) y emisor XY-FST (derecha).

La desventaja de estos módulos es que se necesita una Arduino en el emisor y otra en el receptor. Además, no tienen la compatibilidad ni la flexibilidad de otras tecnologías, como Bluetooth, que permiten, con cualquier teléfono móvil, enviar o recibir información directamente desde la Arduino.

Wi-Fi es una de las tecnologías de transferencia inalámbrica más utilizada hoy en día. Gracias a los módulos y shields de Wi-Fi, se puede incorporar esta tecnología en la Arduino y lograr interactuar fácilmente con dispositivos compatibles.

El **shield WiFi** de Arduino permite la conexión a internet mediante la librería WiFi y leer o escribir datos en una **tarjeta SD**, también con una librería especial para este fin. Se alimenta con 5 V y se comunica con la Arduino a través del bus SPI, ubicado en los pines digitales 11, 12 y 13 en la Arduino UNO.

Se puede conectar a una red abierta o encriptada mediante WEP o WPA2. Otro de los usos de este shield es conectarnos a un **Web server** utilizando la Arduino como WebClient o crear directamente nosotros mismos un Web server en la Arduino.

El hecho de poder guardar datos en la tarjeta SD permite mayor flexibilidad, ya que muchas veces la memoria de Arduino puede ser limitante.

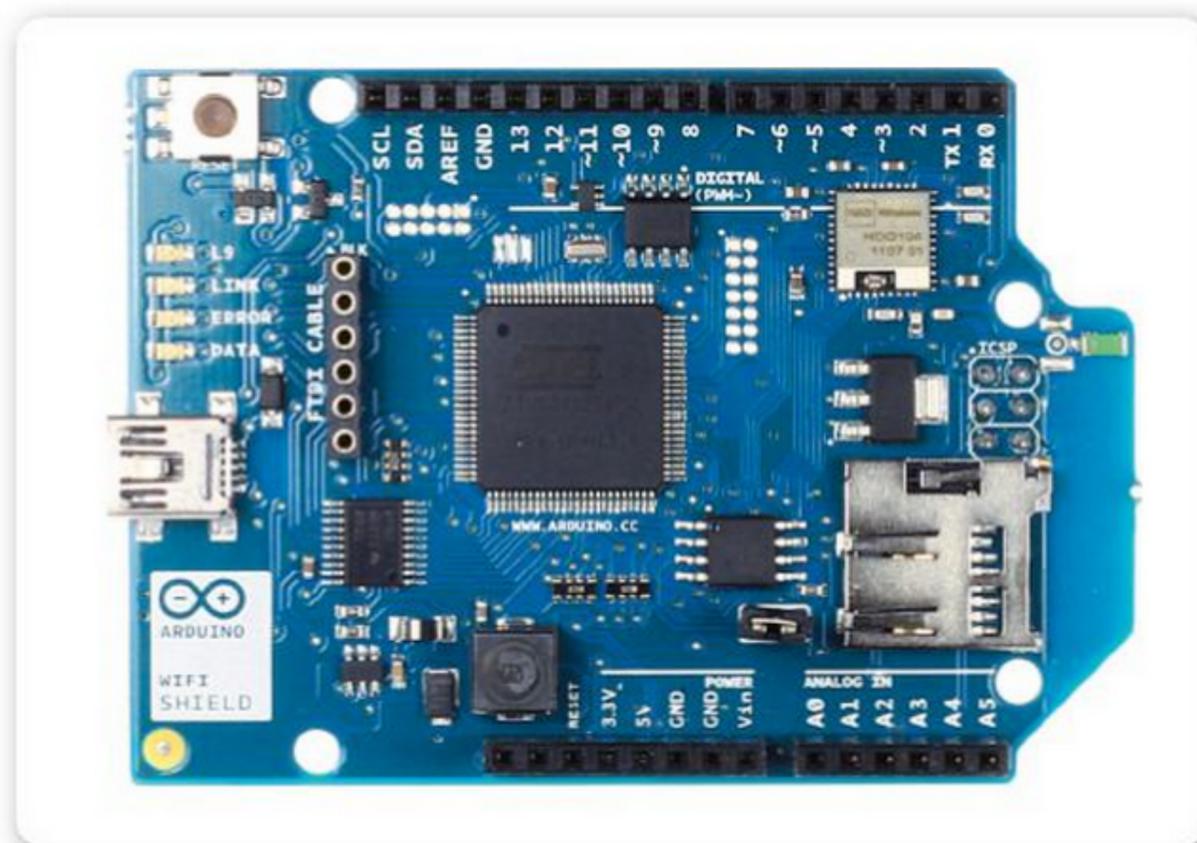


Figura 15. Imagen del shield WiFi oficial de Arduino. Por lo general, incorpora un módulo SD para leer/escribir datos.

Una de las principales ventajas del uso de Wi-Fi en un sistema de telemetría es la posibilidad de crear una red o Web server y monitorizar los datos enviados. La desventaja es la corta distancia que se puede cubrir con esta tecnología, que, en el mejor de los casos, a campo abierto, sin paredes, puede llegar a 100 metros.

ZigBee es un conjunto de protocolos de comunicación, basado en el estándar WPAN que opera en la banda de los 2.4 GHz. Con fuertes ventajas para el uso en domótica, el sistema se destaca por su bajo consumo y un mayor alcance, en comparación, por ejemplo, con Bluetooth. La tasa de transferencia es algo baja –250 kbits/s–, por lo que no resulta suficiente para aplicaciones multimedia o para la transferencia de grandes archivos. Sin embargo, para transferencias menores, como valores o estados de sensores, resulta más que suficiente.

CADA SISTEMA TIENE
SUS BENEFICIOS
Y DESVENTAJAS
EN DETERMINADOS
CAMPOS

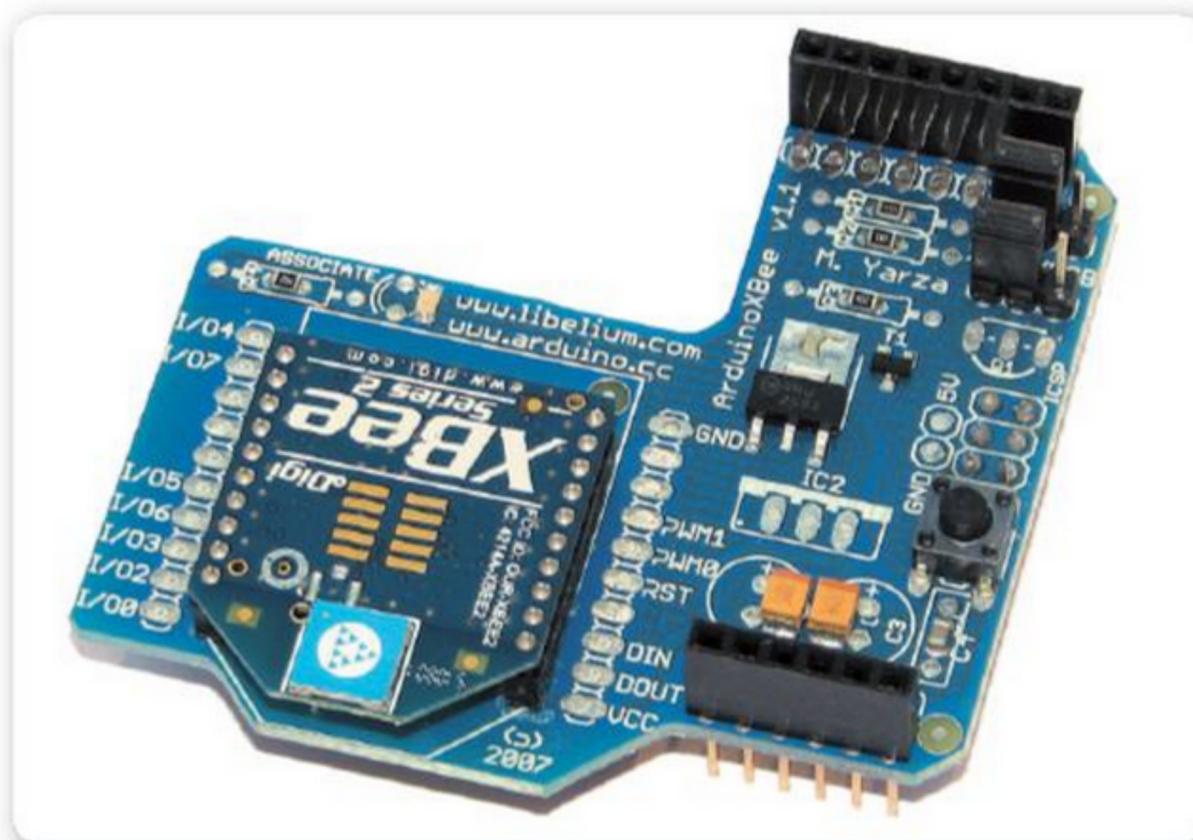


Figura 16. Este es el shield completo para Arduino que nos proporcionará conectividad ZigBee. Se compone de dos placas: la inferior es la interfaz de comunicación y la pequeña, la que implementa el protocolo.

Otro gran punto a favor de esta tecnología son sus **topologías de red**: en estrella, en árbol y en malla. Esta última es la más destacada, soporta hasta 65535 nodos distribuidos en subredes de 255 nodos, lo

LOS PROTOCOLOS SE
IMPLEMENTAN EN
ARDUINO GRACIAS A
LOS SHIELDS
Y MÓDULOS

que le da la robustez necesaria para interconectar sistemas de alta complejidad.

Por último, el **costo de implementación** de estos nodos es muy inferior al de sus competidores, ya que requiere solamente el 10 % del hardware necesario para cada nodo, comparado con Bluetooth o Wi-Fi, y llega hasta la irrisoria cifra del 2% para los nodos más sencillos.

Otro protocolo de comunicación muy utilizado hoy en día es el **Bluetooth**, diseñado específicamente en sus orígenes para lograr una

comunicación inalámbrica de **corto alcance**, de un **ancho de banda muy bueno** (entre 1 y 24 Mbits/s, dependiendo de la versión), con **bajo consumo y bajo costo**. Debido a su simplicidad, generalmente utilizamos módulos en Arduino para obtener este método de transmisión, ya que, si bien existen shields, aportan otras características adicionales como pines de expansión, que a veces no justifican la diferencia de precio.

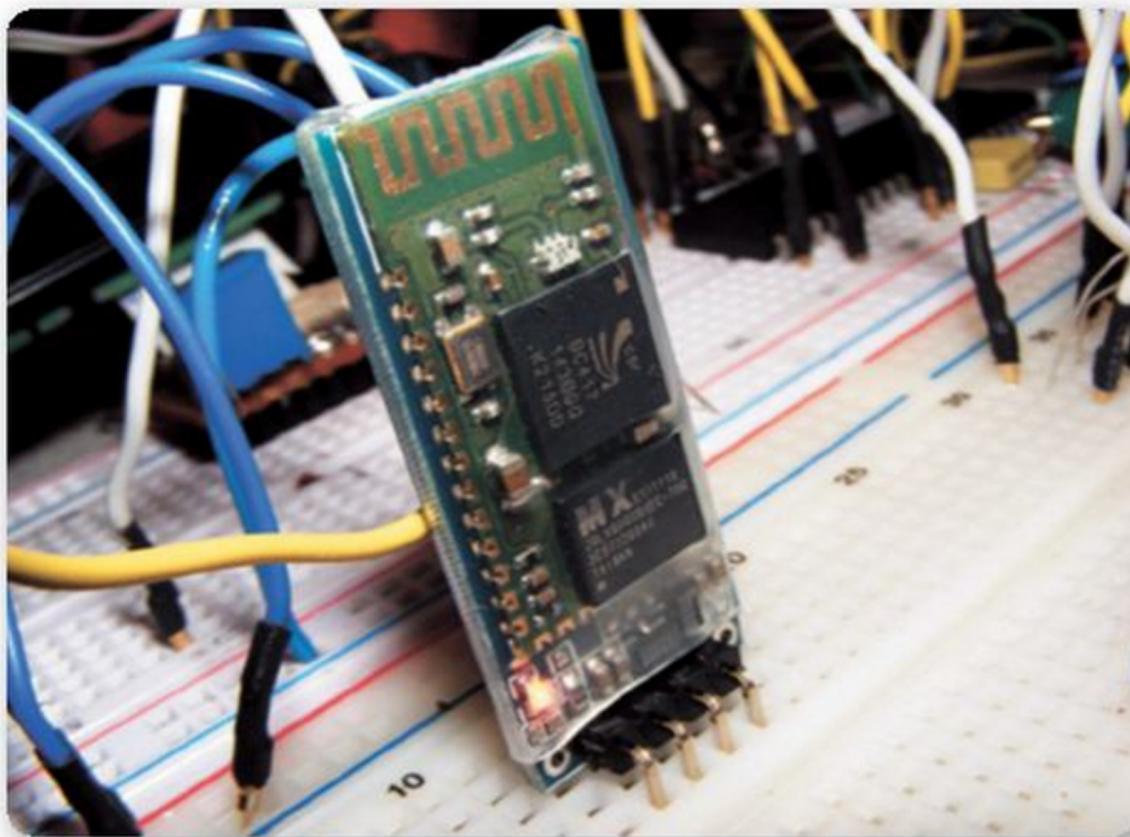


Figura 17. Así se ve el módulo **Bluetooth JY-MCU** conectado a una protoboard. Existe una versión con dos pines más: STATE (reporta el estado) y KEY (encriptación de la señal).

Un módulo muy utilizado es el **JY-MCU**, el cual es la opción económica, pero que, a su vez, ofrece una excelente relación

precio/calidad. Este módulo dispone de cuatro pines: GND, VCC, Tx y Rx, ya que se comunica con la Arduino mediante el protocolo UART. El integrado trabaja con los valores de tensión lógicos propios de TTL, por lo que hay que utilizar un divisor de tensión en el RX (para transformar los 5 V de la Arduino en los 2.4-3.3 V para el estado high).

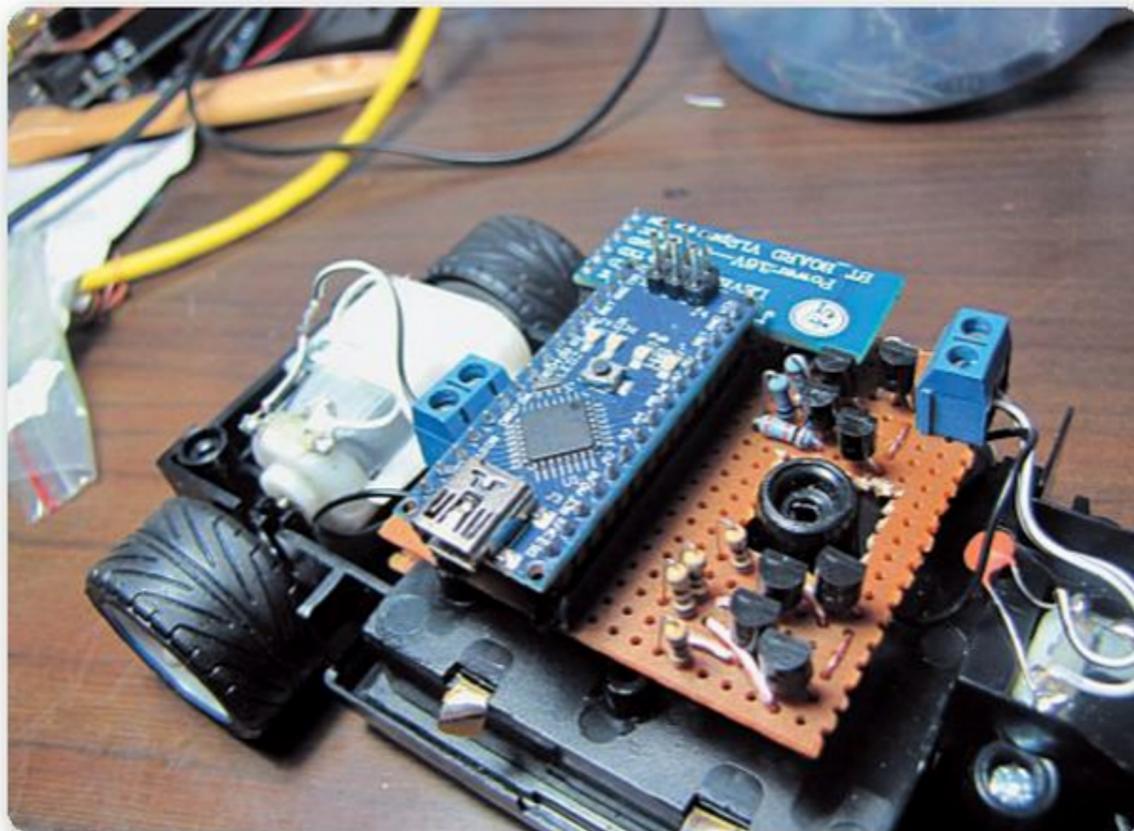


Figura 18. Coche montado con placa Arduino controlado con RC mediante un teléfono móvil y Bluetooth.

En cuanto a las aplicaciones, son muy variadas. Gracias a que este protocolo está presente en gran cantidad de **dispositivos electrónicos cotidianos**, nos permite una interacción muy completa que va desde la transmisión de datos hasta el control inalámbrico, ya sea mediante periféricos u otras plataformas móviles, como **smartphones**.

Componentes del sistema de telemetría

Como habíamos comentado antes, nuestra idea en este proyecto será **transmitir una posición medida por GPS a través de la red de telefonía móvil GSM**. Usaremos un módulo GSM para enviar un mensaje de texto a un número determinado, cada un cierto intervalo de

tiempo previamente establecido. En el contenido del mensaje, colocaremos las coordenadas de latitud y longitud obtenidas desde un shield GPS, seguido de un link para visualizarlo en **Google Maps** .

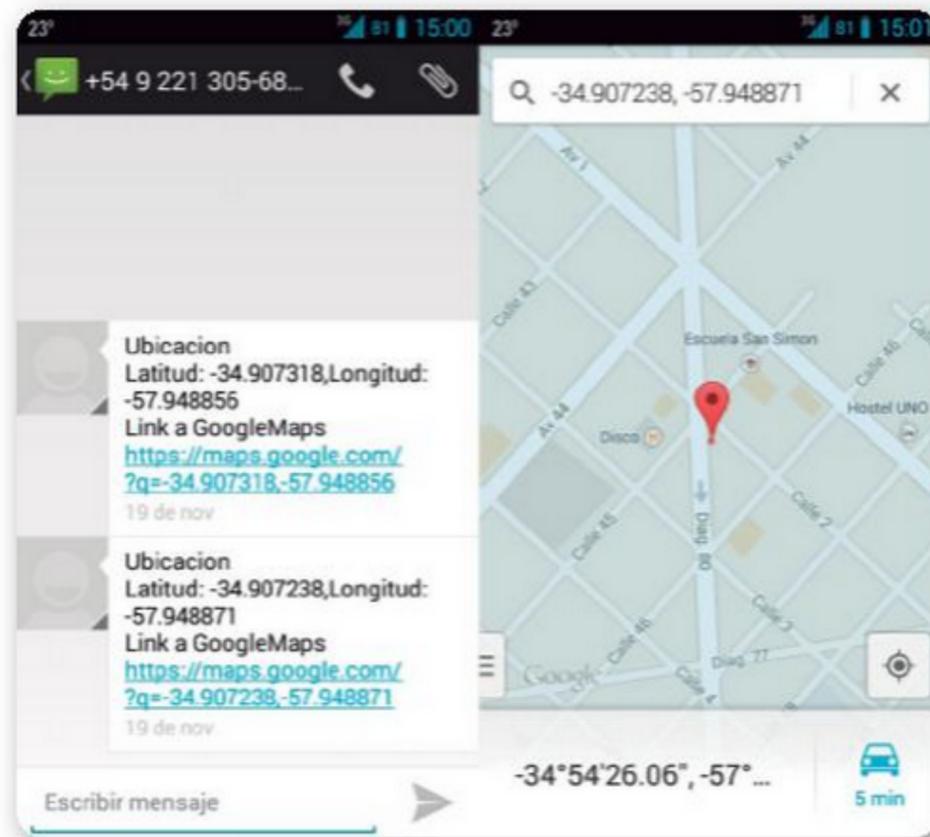


Figura 19. Un smartphone que prueba la recepción del mensaje. Así debería verse nuestro sistema de telemetría.

¿Qué vamos a usar?

Placa Arduino : utilizaremos la **Arduino UNO**, revisión 3, la más común de las placas de Arduino. La elegimos porque es la que mayor difusión tiene en internet y, además de ser bastante económica, es también de las más accesibles en Latinoamérica. Se pueden usar otras placas para esta aplicación, pero se debe tener en cuenta la diferencia de ubicación de los pines, en especial los del puerto serie (Tx y Rx).



NEAR FIELD COMMUNICATION

Este protocolo no está pensado para utilizarse con telemetría, dado su corto alcance (20 cm). Sin embargo, merece su mención ya que para determinadas aplicaciones resulta sumamente útil. De muy bajo consumo, posee dos métodos de comunicación: activa y pasiva. En el primer modo, ambos dispositivos emiten sus propios campos magnéticos. En el modo pasivo, solo un dispositivo genera este campo.

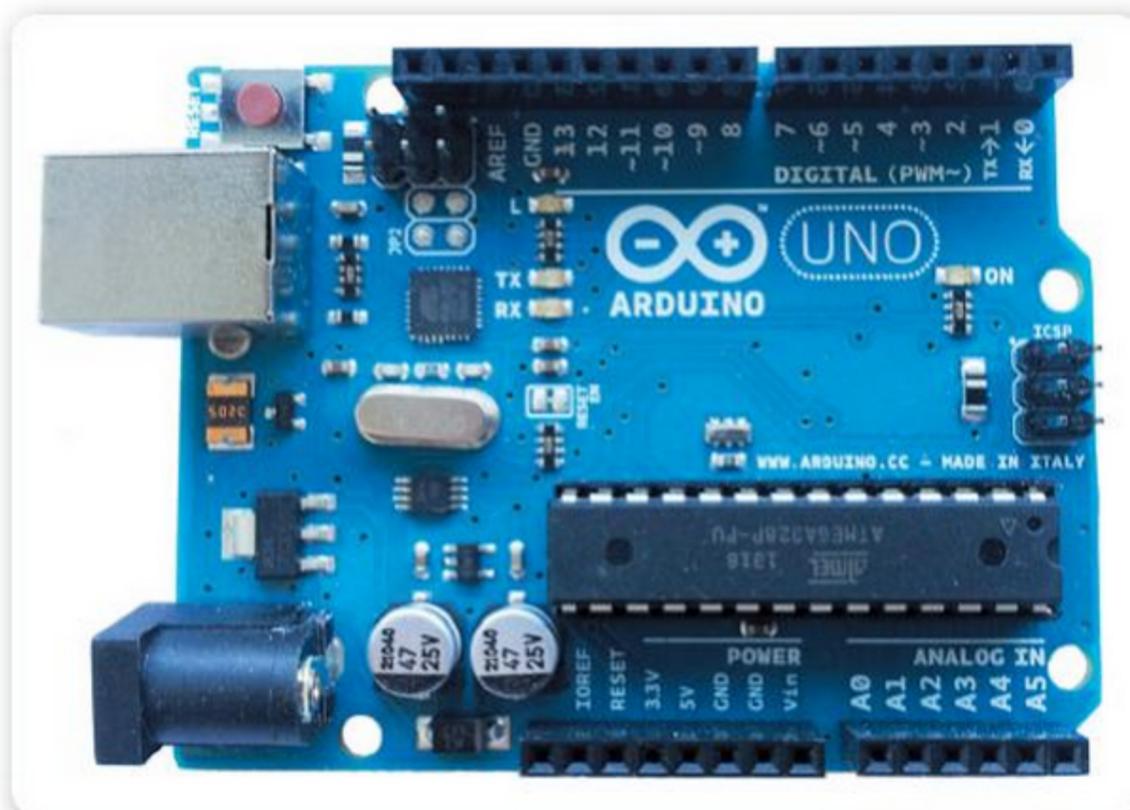


Figura 20. La **Arduino UNO** lista para acoplarse con los dos shields adicionales que requiere este proyecto.

Módulo/shield GPS: como nuestro proyecto será la transmisión de la posición medida con GPS, necesitaremos un módulo o shield que realice esa medición. Existen distintos fabricantes de este tipo de shields, que variarán de acuerdo con la región donde nos encontremos. Hemos elegido uno en particular, pero es probable que la explicación que demos y los comandos que usemos sean totalmente extrapolables a otros shields, si se aplican correctamente.

Para este proyecto hemos conseguido un shield GPS desarrollado por ElecFreaks, llamado **GPS Shield-Fastrax UP501**, que, como su nombre lo indica, está basado en un chip **UP501** de la empresa Fastrax.

Sus principales características son:

- **Bajo consumo:** 75 mW típico operando a 3 V, se eleva a unos 115 mW en la etapa de búsqueda de satélites.
- **Tensión de alimentación:** 3.0-4.2 V. En nuestro caso, el shield posee un regulador de tensión que eleva el límite hasta los 5 V, permitiéndonos alimentarlo directamente desde la Arduino.
- **Muy alta sensibilidad:** -148 dBm para el inicio de adquisición de señal, -158 dBm para la readquisición de señal (en caso de haberse perdido la comunicación) y -165 dBm para la navegación una vez establecida la conexión.

- **Tamaño reducido** : en un área de 22.0 x 22.0 x 8.0 mm, el UP501 integra todo lo necesario, incluyendo una antena embebida (intenna) de 18.4 x 18.4 x 4.2 mm.
- **Velocidad de actualización configurable** : por defecto, 1 Hz, pero se puede llevar hasta los 10 Hz vía comandos NMEA.
- **Batería opcional de backup (UP501B)** : utilizada para los RTC y los bloques de RAM no volátiles. De no ser incluida, necesitamos alimentar con una tensión de backup VDD_B (ya implementado en el shield), de entre 2.0-4.2 V, pero con muy baja potencia (15 μ W típicos a 3 V).
- **Puerto serial** : utiliza los niveles de tensión CMOS, aunque gracias al shield adapta los niveles de tensión de la señal UART para que sean compatibles con la Arduino. También existe una variante (UP501R) que incluye un adaptador de tensión embebido en el chip, que lo hace compatible con el protocolo RS-232.
- Compatible con **WAAS/EGNOS/MSAS** : estos sistemas se nutren de estaciones terrestres a partir de las cuales corrigen los errores en las señales de GPS, mejorando la precisión en la ubicación desde los 10-12 m (resolución propia del sistema GPS) hasta errores inferiores a los 2 m.
- Comunicación mediante el protocolo **NMEA** : por defecto, configurado a 9600 bits/s.

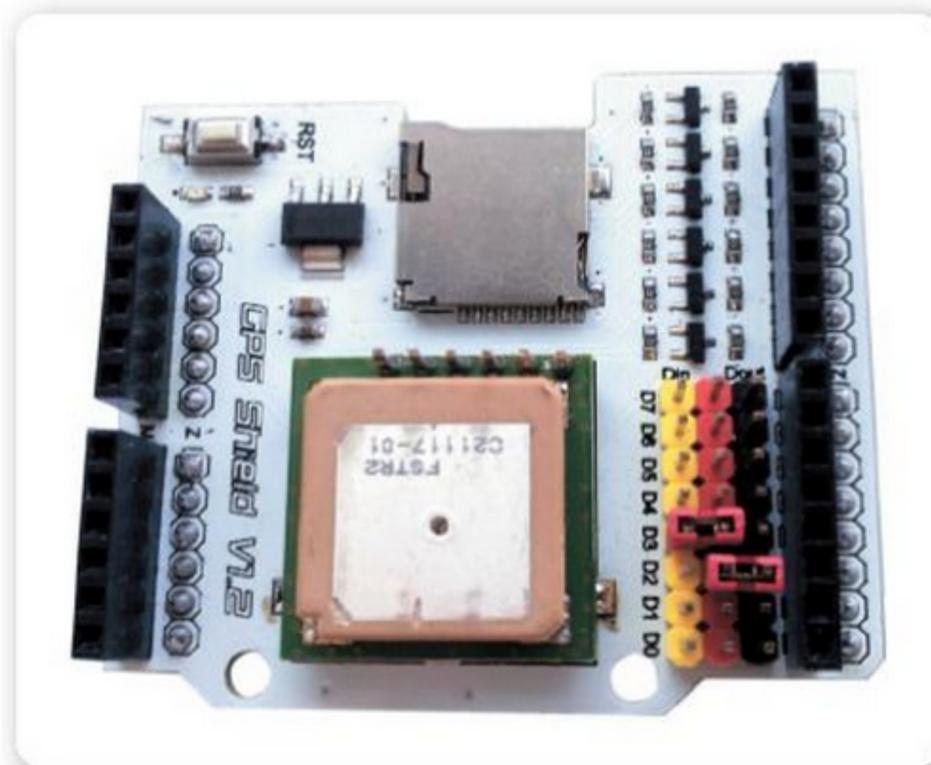


Figura 21. Esta es nuestra versión del shield que utiliza el **Fastrax UP501**. Se observa el gran encapsulado frente al lector de tarjetas micro SD.

Por último, podemos detallar los tres modos de funcionamiento:

Modo búsqueda/navegación: aun habiendo encontrado suficientes satélites como para otorgar el posicionamiento del dispositivo, el sistema sigue buscando otros satélites adicionales, los cuales va almacenando para luego, en caso de necesitarlos, poder utilizarlos.

Modo navegación/bajo consumo : una vez que obtuvo los datos suficientes, el dispositivo deja de buscar y continúa solamente con el modo de navegación, por lo que reduce su consumo significativamente.

Modo backup : cuando se interrumpe la fuente de energía principal (VDD), el chip entra en modo backup si la alimentación destinada a este objetivo se encuentra disponible, ya sea mediante una batería u otra conexión aparte. Con esta alimentación, mantiene el RTC (reloj de tiempo real), la configuración que hayamos realizado en él y la RAM donde almacenó el registro de los satélites encontrados.

Módulo/shield GSM : el módulo que utilizaremos para la explicación de este proyecto será uno fabricado también por la empresa ElecFreaks, llamado **GPRS/GSM Module-EFCom Pro**, basado en el chip SIM900 de la empresa SIMCOM, un integrado con las siguientes características:

- Rango de la tensión de alimentación entre 3.2 y 4.8 V (se puede monitorizar mediante comandos AT).
- Bajo consumo: en modo Sleep, solo 1.5 mA.
- Puede trabajar en las **cuatro bandas de frecuencia** GSM (850/ 900/ 1800/ 1900 MHz), realiza una **búsqueda automática** de estas.
- Potencia de transmisión de 2 W en clase 4 a 850 y 900 MHz y 1 W en clase 1 a 1800 y 1900 MHz.
- Conectividad **GPRS** multislot clase 10 u 8, con velocidad de bajada máxima de 85.6 kbits/s y de subida de hasta 42.8 kbits/s. Soporta protocolo **TCP/IP**.
- Soporte para SMS, CSD, USSD, FAX.
- Temperatura de operación de -40 °C a +85 °C.
- Interfaz de audio con cancelación de eco y supresión de ruidos.

LOS CHIPS DE GPS
TIENEN VARIOS MODOS
DE FUNCIONAMIENTO
PARA OPTIMIZAR
LA CALIDAD



- Puerto conversor analógico-digital (ADC).
- Interfaz para **display LCD** y matriz de teclas (**keypad**).
- Interfaz **SIM** para tarjetas de 1.8 y 3 V.
- Comunicación serie mediante **UART**, **SPI** o **I2C**.
- 12 puertos de entrada/salida para propósitos generales (**GPIO**)
- 2 puertos para generar funciones moduladas por ancho de pulso (**PWM**).
- Interfaz de **antena independiente** (se puede cambiar para mejorar la señal).
- Puerto **Serie** controlado con **comandos AT**, velocidades desde 1200 a 115200 bits/s.
- Puerto **Debug** para depurar y actualizar firmware.
- **Agenda de contactos** y reloj de tiempo real.

Los comandos que se pueden pasar para configurar este chip SIM900 son varios; se puede descargar el **SIM900_AT Command Manual** , donde se encuentra la lista completa. Sin embargo, más adelante explicaremos los más comunes.

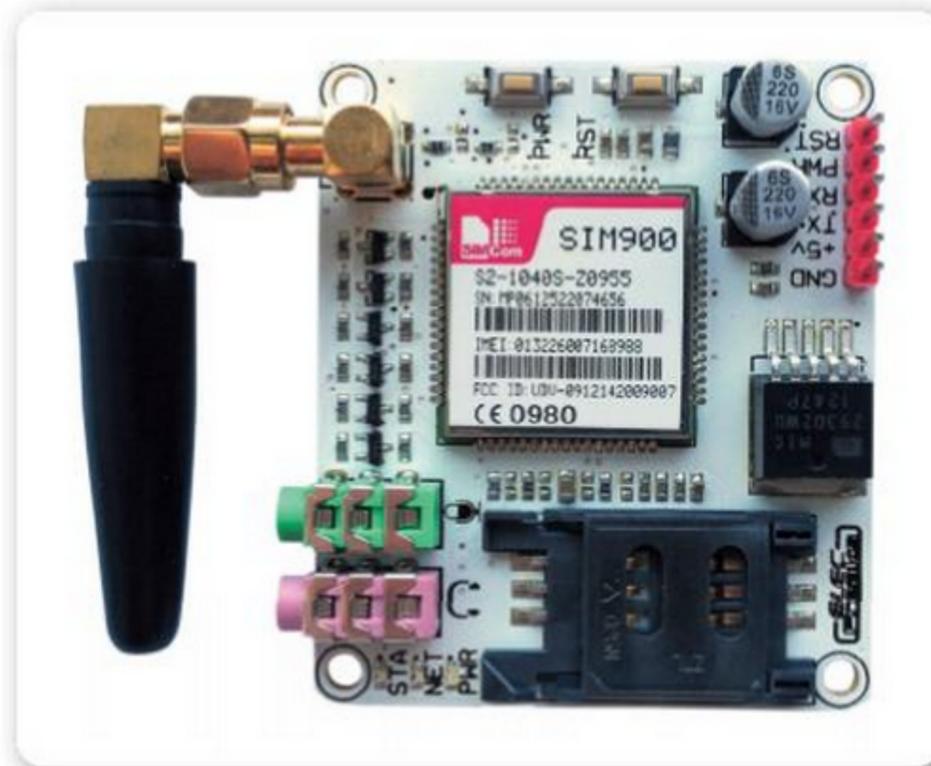


Figura 22. Módulo **GPRS/GSM Module-EFCom Pro** de **ElecFreaks** utilizado en este proyecto. Se puede observar la antena, el chip SIM900, la interfaz de audio y el slot para tarjeta SIM.

Con el objetivo de incentivar la **creatividad** , nombramos una lista de funciones que podrían realizarse con este módulo GSM y otros agregados:

En lugar de enviar la posición cada un cierto tiempo establecido, se puede enviar un mensaje de texto solo cuando el módulo GSM recibe un mensaje desde algún número, con algún texto en particular. Esto sería menos costoso si no se necesita recibir constantemente la posición o si no se poseen números gratis con el servicio de red.

Se puede usar para controlar sistemas a distancia en lugar de monitorizarlos; por ejemplo, para que se active cierto componente cuando se reciba una llamada o SMS de cierto número.

Al poder implementar conexiones de internet mediante GPRS, se podría realizar, por ejemplo, un seguimiento de la posición y verla de manera continua en un mapa de Google. También se podría agregar una cámara en la Arduino y utilizar la conexión para enviar imágenes además de la posición. Combinando GSM/GPRS, cámara, audio y GPS, se puede lograr un interesante sistema de seguridad para un auto, por ejemplo.

LAS FUNCIONES
QUE REALIZA
GSM/GPRS CON
OTRAS TECNOLOGÍAS
SON MUY AMPLIAS



Armado y explicación del código de Arduino

Como detallamos previamente, el shield GPS no necesita alimentación externa, por lo que procedemos directamente a montarlo sobre la Arduino. Tenemos que prestar atención: con la **Arduino UNO** es fácil, ya que por la disposición de sus pines hay una sola forma de conectarlo, pero si tenemos la **MEGA**, debemos **chequear que los pines** cercanos al botón de Reset (RST) queden conectados a las salidas de 5 V y GND, coincidiendo además con los TX0 y RX0 en el vértice opuesto.



SIM900

En este proyecto solo mostraremos brevemente el uso de este **módulo GSM** y una introducción al **chip SIM900**. Este último es bastante más poderoso y permite realizar más funciones. El manual de referencia de comandos AT del SIM900 contiene más de doscientas páginas, y el manual de diseño de hardware, otras cincuenta, que incluyen detalles técnicos de hardware y cuestiones de fabricación.



Una vez acoplado, solo nos falta definir en qué puertos vamos a establecer la comunicación serial con la Arduino. Para esto, el shield dispone de tres líneas de patillas, diferenciadas en tres colores. La línea roja está puenteada a cada pin, y mediante los dos jumpers seleccionamos los pines que nos parezca conveniente para el Din y el Dout (3 y 4, respectivamente, en nuestro caso). Conviene recordar esto, ya que luego tendremos que definirlos en el código, como Tx y Rx.

Conectamos la Arduino a la PC, abrimos el IDE y le cargamos el código que aparece en la imagen. Con este código y la aplicación **GPSTracker** de SkyTraq, podremos ver rápidamente el shield en acción, entregando nuestra posición en tiempo real, con una precisión sorprendente. Si se usan otros pines para el Din y Dout, deberemos modificar txPin y rxPin.

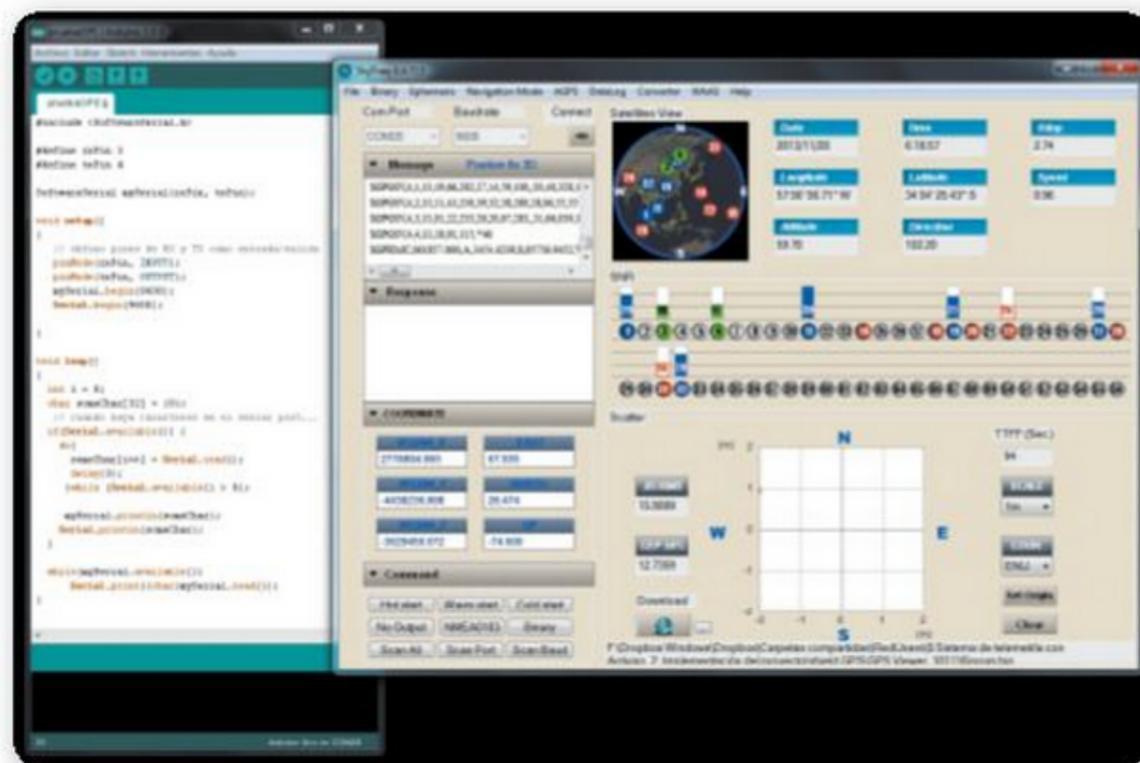


Figura 23. Acá observamos el código de prueba inicial, junto con el software que gráficamente nos muestra los resultados obtenidos en tiempo real.

Para probar el **módulo GSM**, recomendamos seguir los siguientes pasos:

- 1) Instalar la tarjeta SIM en el módulo GSM (no se requiere un plan de datos a menos que se piense usar la conexión GPRS).
- 2) Conectar los pines GSM_RX y GSM_TX del módulo a los pines TX (D1) Y RX (D0) de la Arduino, respectivamente, para comunicarnos con la plataforma mediante UART TTL (5 V).

- 3) Dar alimentación al módulo mediante la conexión de los pines 5 V y GND. Aún no debemos encender el módulo.
- 4) Conectar la Arduino a la PC usando un cable USB.
- 5) Subir el sketch de la imagen a la Arduino
- 6) Encender el Monitor Serial de la Arduino u otro software para leer y escribir en el puerto serie.
- 7) Ahora sí, ya podemos prender el módulo GSM presionando durante alrededor de 2 segundos el botón PWR y esperar que tome señal (Led NET parpadea cada 3 segundos).

EL SHIELD GPS
NO NECESITA
ALIMENTACIÓN
EXTERNA Y SE MONTA
SOBRE LA ARDUINO



Si todo está correcto, debería verse en el Monitor Serial algo similar a esto:

RDY

+CFUN: 1

+CPIN: READY

Call Ready

- 8) Al enviar mediante el Monitor Serial la palabra `AT`, deberíamos recibir `OK`.

- 9) Para enviar un mensaje de prueba, ingresamos los siguientes comandos mediante el Monitor Serial:

`AT+CMGF=1` (responde `OK`);

`AT+CMGS=""<número de teléfono a enviar el mensaje>"`

`<Texto de Prueba>`

`Control + Z` (para indicar fin del mensaje. Si no, también se puede enviar el carácter hexadecimal `01xa`).

- 10) El módem responde `OK` y deberíamos recibir el mensaje en nuestro teléfono móvil.



BÚSQUEDA DE INFORMACIÓN



Para encontrar las **características** de cada shield, debemos buscar en internet el fabricante y el modelo correcto de cada uno. Una vez hallados, los manuales suelen ser extensos, pero es necesario leerlos para saber cómo funcionan. Por ejemplo, un error común es colocar **tensiones** distintas de alimentación por no leer correctamente el manual del fabricante.

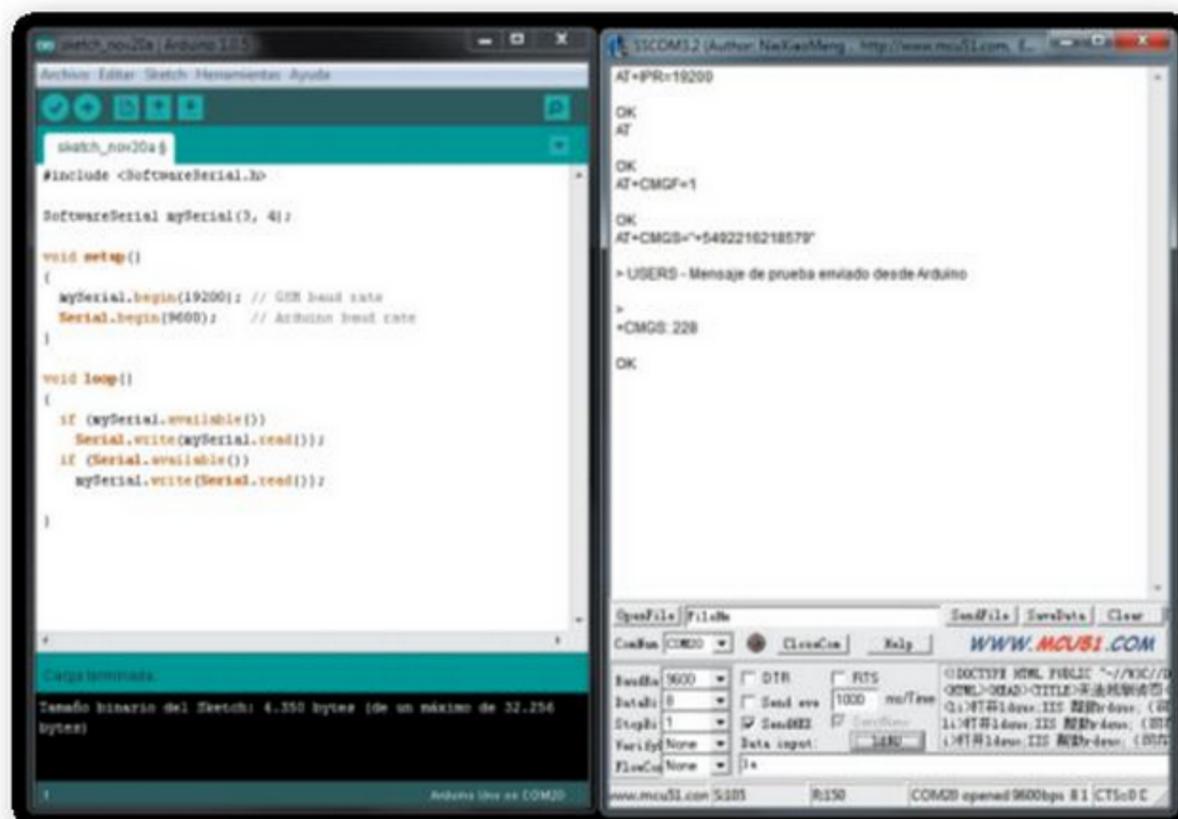


Figura 24. Imagen con el código de Arduino y el software serial utilizado para testear el módulo GSM.

Después de haber teasteado los shields o módulos, llega el momento de unirlos. Como la placa Arduino UNO solo posee un puerto serie físico (ubicado en los pines digitales 0 y 1), será necesario emular un puerto serie virtual en los pines 3 y 4, a través de la librería `SoftwareSerial` (incluida en el IDE).

Si prestó atención, el lector habrá percibido que en los códigos de testeo de los párrafos anteriores se utilizó esta librería para probar que el shield GPS captaba satélites y que el módulo GSM podía recibir comandos AT para enviar mensajes.

El conexionado que usaremos entonces será el shield GPS, configurado para usar los pines 3 y 4 (puertos serie virtuales), y el módulo GSM utilizará el puerto serie nativo por hardware de la Arduino.

Si usáramos una placa **Arduino MEGA**, no sería necesario usar la librería `SoftwareSerial` para emular puertos serie virtuales, ya que de fábrica incorpora por hardware cuatro puertos serie.

Al usar el puerto serie físico del UNO con el módulo GSM, surge la consecuencia directa de que, para subir los sketches, se debe primero siempre desenchufar el módulo GSM, ya que, de lo contrario, generaría conflictos con el puerto serie que el microcontrolador ATMEGA 328 utiliza para conectarse a través de USB a la PC.

Explicación del código final

La primera parte del código corresponde a la inclusión de **librerías** y declaración de variables. Para facilitarnos el uso del shield GPS, vamos a utilizar una librería llamada TinyGPSplus. Esta es la que introducimos mediante el código:

```
#include <TinyGPS++.h>
```

Esta librería nos permitirá crear un objeto (`gps`, en nuestro caso), cuya clase ya está definida y, además, posee las funciones necesarias para los datos que envía el shield.

Al utilizar un software como **GPSViewer**, este se encarga de la interpretación de los datos. Nosotros lo haremos en nuestro código de una manera muy sencilla, gracias a TinyGPSplus, en los siguientes pasos.

Creamos el objeto de clase TinyGPSPlus llamado `gps`:

```
TinyGPSPlus gps;
```

Otra librería que debemos incluir, debido a lo explicado anteriormente sobre la necesidad de los puertos serie virtuales, es la librería `SoftwareSerial`:

```
#include <SoftwareSerial.h>
```

Incluida esta librería, creamos el puerto serie virtual para el GPS con la sentencia:

```
SoftwareSerial GpsSerial(3, 4);
```

Como ya sabemos, dentro del `void setup()` incluimos la configuración inicial del programa, ya que este se ejecuta por única vez.

Aquí solamente tendremos que configurar la velocidad a la que cada módulo transferirá mediante su conexión serial con la Arduino. Ambas partes (GSM-Arduino y GPS-Arduino) deben estar configuradas a la

misma velocidad: esto es muy importante, ya que de ello depende el “entendimiento mutuo” entre ellas.

Por defecto, el shield GPS viene configurado a 9600 baudios/s. Esta velocidad es más que suficiente para nuestro caso y, además, hay que tener en cuenta que una mayor velocidad demanda mayor consumo, por lo que solamente lo elevaremos cuando sea necesario. Mediante la siguiente línea de código, inicializamos el puerto serie virtual:

```
GpsSerial.begin(9600);
```

El módulo GSM, en cambio, viene configurado por defecto a 19200 baudios/s. Tampoco vamos a cambiarlo ya que no es necesario, aunque –si se quisiese– bastaría con ejecutar algunos comandos AT para hacerlo. Ya que lo conectamos al puerto serie físico de la Arduino, lo inicializamos a la velocidad correspondiente mediante:

```
Serial.begin(19200);
```

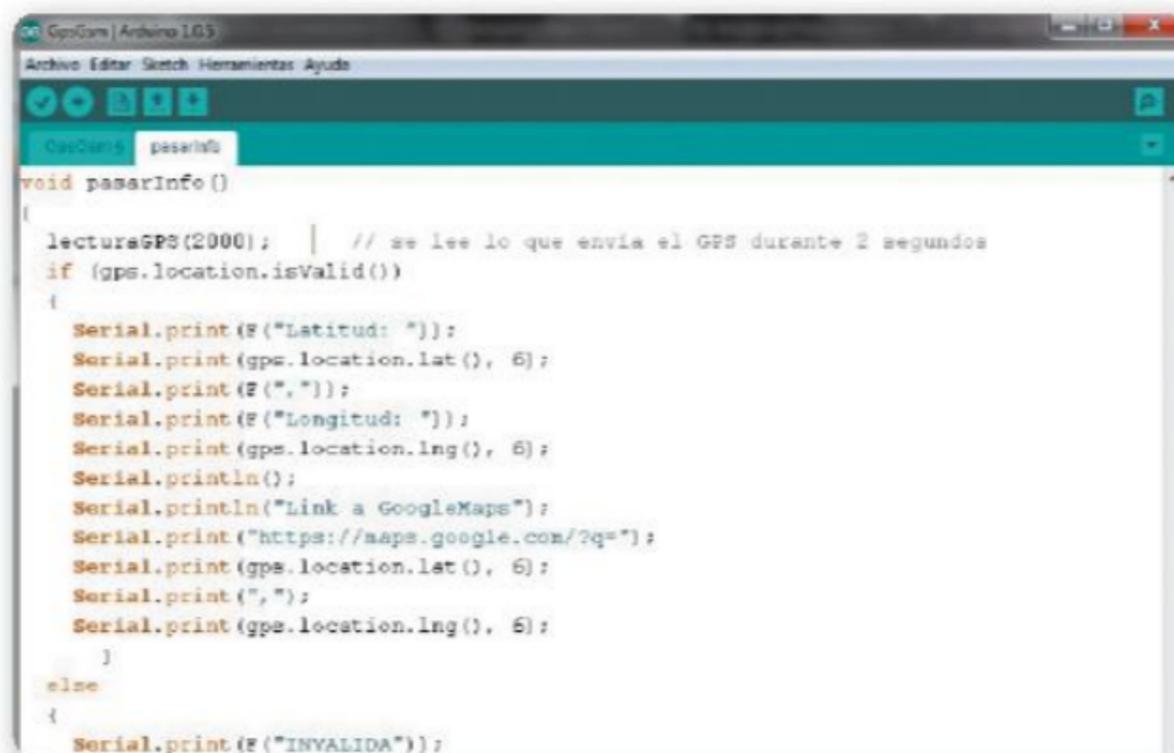
La siguiente parte del código corresponde al contenido del `void loop()`, que se repetirá cíclicamente en la Arduino.

En el comienzo, entregamos desde el puerto serie al módulo GSM el comando `AT+CMGF=1`, que se configurará para enviar luego el mensaje en modo texto.

```
Serial.print("AT+CMGF=1\r");  
delay(1000);  
Serial.print("AT+CMGS=\"+549XXXXXXXXXX\"\r");  
delay(1000);
```

Los delays entre los comandos son necesarios para la correcta interpretación por parte del módulo.

Luego, se imprime el comando `AT+CMGS=\"+549XXXXXXXXXX\"` para indicarle que vamos a enviar un SMS a ese número. En este comando, debemos reemplazar las `x` por el número telefónico al cual queremos enviarle el mensaje, en formato E.123.



```

void pasarInfo ()
{
  lecturaGPS(2000); // se lee lo que envía el GPS durante 2 segundos
  if (gps.location.isValid())
  {
    Serial.print(F("Latitud: "));
    Serial.print(gps.location.lat(), 6);
    Serial.print(F(", "));
    Serial.print(F("Longitud: "));
    Serial.print(gps.location.lng(), 6);
    Serial.println();
    Serial.println("Link a GoogleMaps");
    Serial.print("https://maps.google.com/?q=");
    Serial.print(gps.location.lat(), 6);
    Serial.print(", ");
    Serial.print(gps.location.lng(), 6);
  }
  else
  {
    Serial.print(F("INVALIDA"));
  }
}

```

Figura 25. Función `pasarInfo()`. En esta función se imprimen en puerto serie los datos del GPS.

Ahora el módulo quedará “escuchando” lo que le enviemos, para luego incluirlo en el mensaje.

Lo primero que imprimimos por puerto serie será una línea que diga Ubicación. Luego se llama a la función `pasarInfo()`. Con esta función (ubicada en una pestaña nueva a modo de orden), le pasamos al GSM la porción de datos que nos interesa, la cual obtuvimos del GPS y que quedó almacenada en el objeto `gps`. Además, en el final de `pasarInfo()`, generamos el enlace de Google Maps que nos dará una representación gráfica de nuestras coordenadas, si disponemos de un smartphone.

```

Serial.print(F("Ubicacion "));
pasarInfo();
Serial.print((char)26);

```

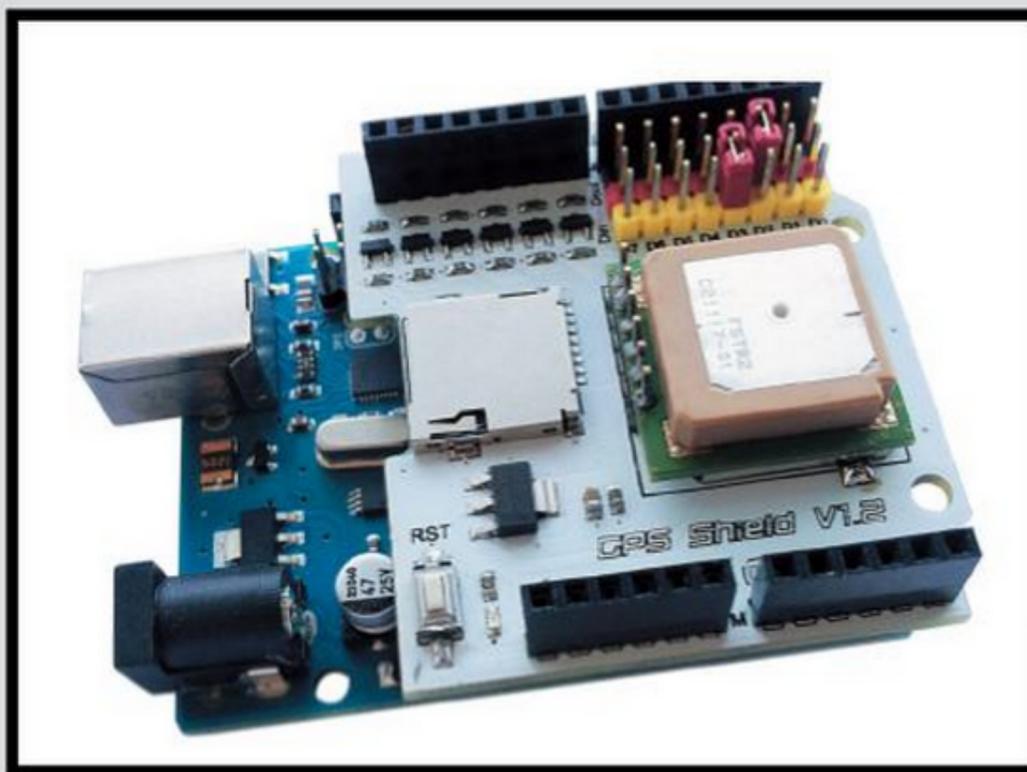
En caso de que el GPS no tuviera señal –por ejemplo, al encenderlo–, la sentencia `gps.location.isValid()` retorna `false`, por lo que `pasarInfo()` devuelve “INVÁLIDA”.

En el final del `void loop()`, se coloca un `delay` que será el que marque el tiempo del ciclo. Nosotros hemos colocado 180 segundos (3 minutos) como intervalo para que envíe mensajes de texto con la posición, pero se puede modificar a gusto.

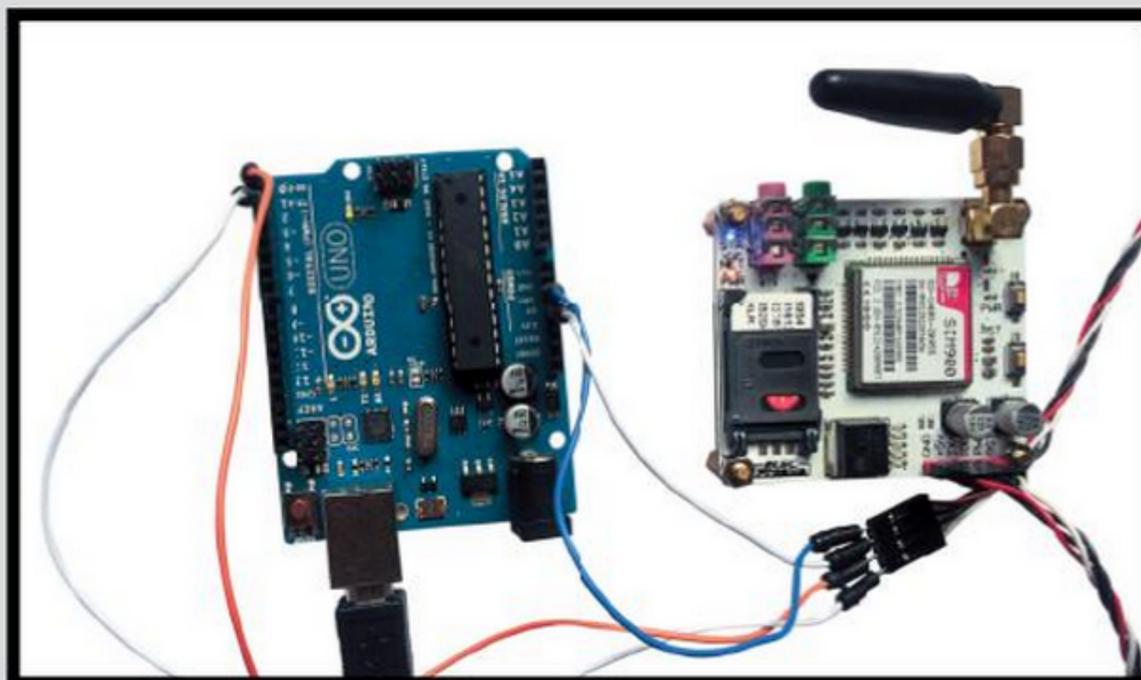
PAP. TELEMETRÍA CON ARDUINO GPS+GSM



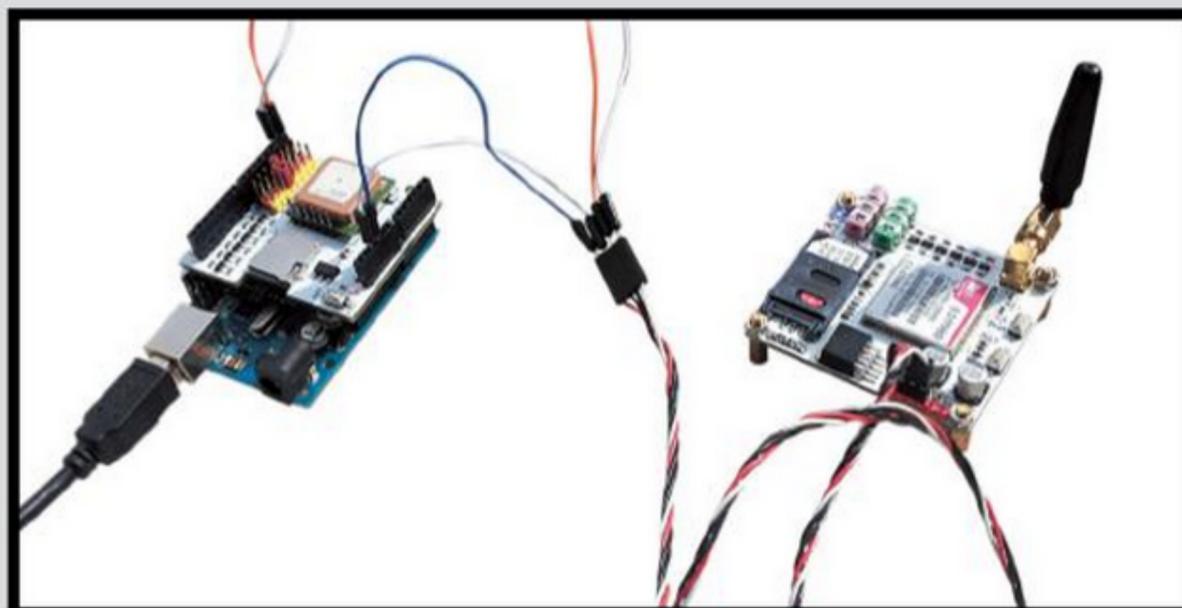
- 01** Conecte el shield GPS a la Arduino prestando atención a las consideraciones sobre la orientación previamente explicadas. Verifique que capte satélites.



- 02** Desconecte el shield GPS y envíese un mensaje de texto mediante puerto serie para probar el módulo GSM.



03 Luego de probar por separado el GPS y el GSM, una los shields y comience a cargar el siguiente código.



04 Así debería quedar la primera parte del código. Reemplace RXpin y TXpin si se usaron otros pines. Instale la librería TinyGPSplus en ~/Arduino/libraries/.

```
GpsGsm | Arduino IDE
Archivo Editar Sketch Herramientas Ayuda
GpsGsm$
#include <SoftwareSerial.h> //Incluimos las librerías que vamos a necesitar.
#include <TinyGPS++.h>

TinyGPSPlus gpe; // Creamos el objeto "gpe".

SoftwareSerial GpsSerial(3, 4); // Creamos el puerto serie virtual para el GPS.

void setup()

  Serial.begin(19200); // Seteamos la conexión con el módulo GSM a 19200 Bdps.
  GpsSerial.begin(9600); // Lo mismo con el GPS pero a 9600Bdps.

void loop()

  Serial.print("AT+CMGF=1\r"); // Indicamos que vamos a mandar texto.
  delay(1000);
  Serial.print("AT+CMGS=\"+54922216218579\r\"); // Reemplazar las X por el número.
  // Luego de este comando lo que se
  // imprime en el puerto serie será
  // lo que se enviará en el mensaje.
  delay(1000);
```

05 Así se ve la otra parte del código, con la función `pasarInfo()` que imprime en puerto serie los datos del GPS.

```
GpioSim [Arduino 10.5]
Archivo Editar Sketch Herramientas Ayuda

pasarInfo

void pasarInfo ()

lecturaGPS(2000); // se lee lo que envia el GPS durante 2 segundos
if (gps.location.isValid())
{
  Serial.print(F("Latitud: "));
  Serial.print(gps.location.lat(), 6);
  Serial.print(F(", "));
  Serial.print(F("Longitud: "));
  Serial.print(gps.location.lng(), 6);
  Serial.println();
  Serial.println("Link a GoogleMaps");
  Serial.print("https://maps.google.com/?q=");
  Serial.print(gps.location.lat(), 6);
  Serial.print(", ");
  Serial.print(gps.location.lng(), 6);
}
else
{
  Serial.print(F("INVALIDA"));
}
```

06 Por último, si todo se realizó correctamente, recibirá un mensaje de texto como el de esta figura.



Comandos AT más comunes

El conjunto de **comandos AT** se convirtió en estándar abierto universal de comandos para configurar y parametrizar módems. Los caracteres AT provienen de la palabra **atención** y, al preceder a todos los comandos, surge la denominación “comandos AT”.

La sintaxis básica es “AT<x><n>” o “AT&<x><n>”, donde “<x>” es el comando y “<n>” son los argumentos del comando. Para pasar parámetros, se usan comandos con este formato “ATS<n>=<m>”, donde “<n>” es el índice del registro para configurar, y “<m>”, el valor para asignarle. “<m>” es opcional: si no se especifica, se asignará el valor por defecto.

Podemos encontrar básicamente los siguientes cuatro tipos:

Comandos de verificación: “AT+<x>=?” →. Retorna una lista de parámetros configurados que corresponde a comandos de escritura o procesos internos.

Comandos de lectura: “AT+<x>?” →. Retorna el valor actual del parámetro o los parámetros.

Comandos de escritura: “AT+<x>=<...>” →. Este comando configura parámetros que pueden ser definidos por el usuario.

Comandos de ejecución: “AT+<x>” →. Estos leen parámetros no modificables afectados por procesos internos en el módulo GSM.

Algunos de estos comandos los hemos utilizado en nuestro código. Si bien la lista es bastante extensa, aquí daremos los principales para voz/SMS y su explicación:

“AT”: comando de prueba.

“AT&V”: detalla la configuración actual.

“AT+COPS?”: devuelve el operador de red actual.

“ATA”: atiende una llamada entrante.

“ATDxxxxxxxx”: realiza una llamada al número “xxxxxxxx”.

“ATH”: cuelga la llamada en curso.

“AT+CMFG=1”: configura el sistema de SMS en modo texto, contrario al modo PDU (protocol data unit).

“AT+CMGS=”xxxxxxxx”

<Ctrl+Z>: envía un mensaje al número xxxxxxxxxxx.

“AT+CMGR=1”: lee el mensaje de texto que acaba de llegar.

LA PRIMERA
PARTE DEL CÓDIGO
CORRESPONDE A
LIBRERÍAS
Y VARIABLES



Limitaciones y otras alternativas

Como se ha comentado antes, la idea de este proyecto era **unir conceptos previamente adquiridos** de manera aislada y proponer una aplicación particular para juntarlos. Elegimos los elementos particulares siguiendo un esquema de **aprendizaje**, pensando en un primer proyecto de este tipo para el lector. Si al finalizarlo, se desea ir más lejos y encarar este plan de actividades (o algo similar) más profesionalmente, puede que le sean útiles las siguientes líneas.

En cuestiones de costo, hay que tener en cuenta que comprar los shields adicionales para la Arduino, por lo general, no suele ser económico. Si bien la placa Arduino UNO no posee un costo elevado para la gran cantidad de funciones que ofrece, en particular el shield GPS y el de GSM/GPRS tienen valores más altos, y esto puede ser una limitación en más de un caso. Una solución posible puede ser usar los módulos Bluetooth o RF, que son más económicos, pero tendremos las limitaciones propias de estas tecnologías, detalladas anteriormente.

Otra cuestión que debe tenerse en cuenta es el **tamaño** y las dimensiones que ocupará la Arduino con todos sus agregados, que puede ser un problema en algunas aplicaciones particulares donde se necesite portabilidad y practicidad. Si pensamos colocar un sistema de telemetría antirrobo para ver la posición de un auto, esto no es un problema porque hay espacio de sobra. Pero si se piensa en un micrófono espía, la cosa cambia. Vinculado con esto, está el tema del **consumo de potencia**; dijimos que, en este caso, el módulo GSM necesita una fuente auxiliar para proveer los picos de 2 A requeridos en ciertos modos de operación. De nuevo, si lo montamos en un auto,



CUIDADO CON LAS LIBRERÍAS



El uso de múltiples librerías agranda mucho el código y deja poco espacio para el programa. En nuestro ejemplo, usamos una placa Arduino UNO con 32 kB de **memoria flash** dentro del microcontrolador ATmega238 y el código ocupó un poco más de 10 kB. Frente a estos problemas, hay que reducir el código quitando librerías y programar manualmente o conseguir otra placa con más memoria.

no tendremos problemas de energía; pero si necesitamos algo pequeño, tendrá **pilas o baterías** y tendremos que reconsiderarlo.

Otra de las limitaciones que nos surgieron y que explicamos un poco más arriba es la cantidad de **puertos serie UART** de hardware que incorpora la placa Arduino UNO. Al tener solo uno, tuvimos la necesidad de crear un puerto serie virtual mediante la librería SoftwareSerial. El uso de esta librería tiene ciertas limitaciones propias; la más importante es la **velocidad** soportada, de hasta 9600 bd/s. Este límite de velocidad puede ser un problema, por ejemplo, si queremos usar una conexión de datos GPRS; en ese caso, deberíamos pensar en alguna placa alternativa con más puertos UART físicos, como la Arduino MEGA, la ADK o la DUE.

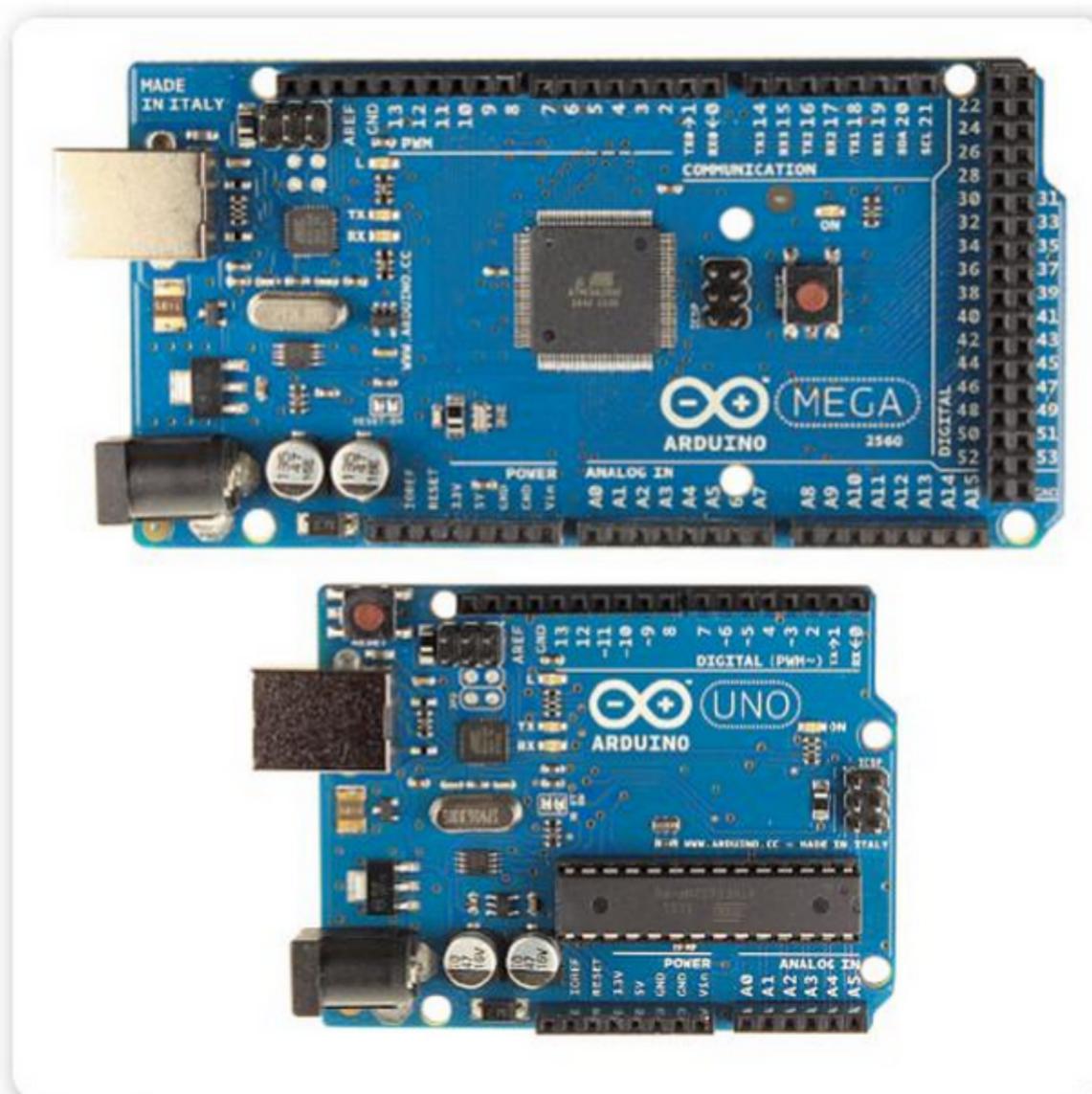


Figura 26. Comparación entre Arduino UNO y Arduino MEGA. Se nota a simple vista que la MEGA nos ofrece más pines de E/S, analógicos y de comunicación serie.

En una de las ideas anteriores, nombramos la posibilidad de tomar **imágenes** con una **cámara** y transmitir las por internet. Si bien este proyecto en teoría puede realizarse con el módulo GPRS,

la velocidad de conexión que este maneja (alrededor de 40 kbits/s de subida) será un problema. Para más velocidad, se pueden considerar los shields con tecnología **WCDMA (3G)** o **HSDPA (3.5G)**, aunque sus costos son elevados.

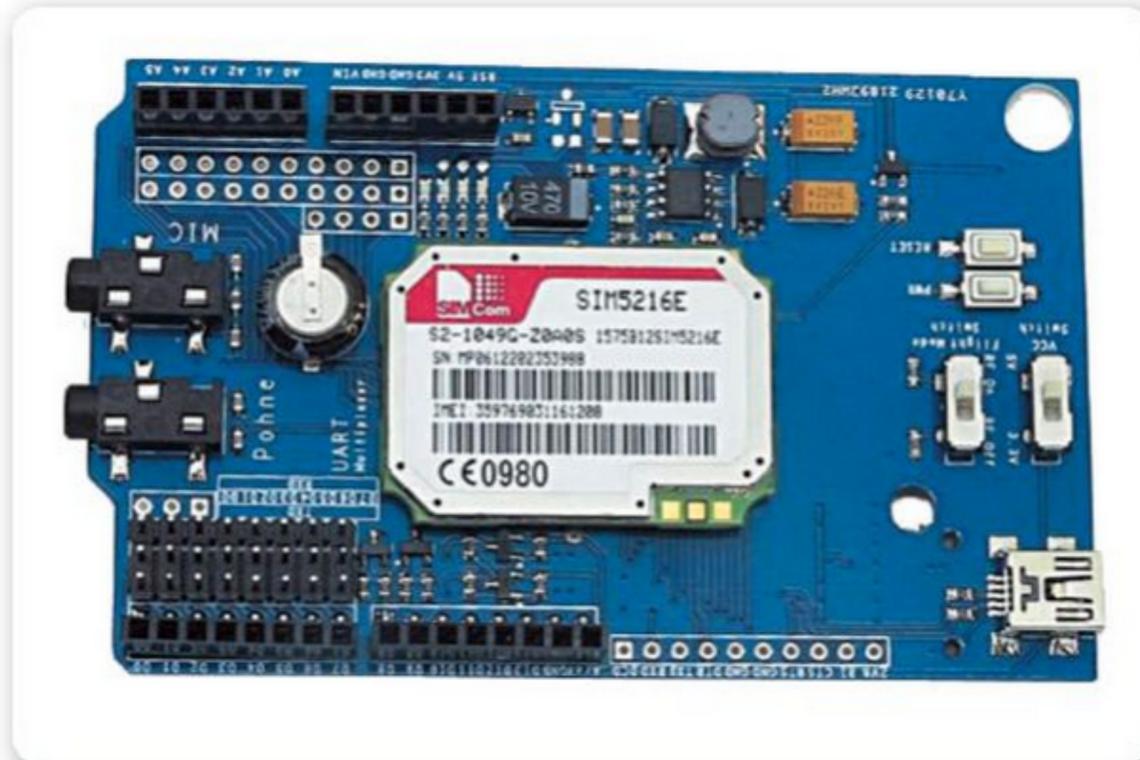


Figura 27. El shield 3G para Arduino nos permite conectarnos a internet con un enlace de alta velocidad.

El shield GPS que usamos en este proyecto no tiene la posibilidad de agregar una **antena externa**. Para poder captar satélites rápidamente es casi vital tener visión directa con el cielo, y esto implica otra restricción para tener en cuenta. Para esto, es necesario colocar la Arduino con el shield GPS en la ventana para poder tomar más de cuatro satélites y medir correctamente la posición. El módulo GSM sí cuenta con la posibilidad de antena externa (incluida aunque reemplazable), como se veía en las imágenes y se detalló en la sección anterior.



INTERFAZ GRÁFICA

Un agregado interesante son los displays LCD o **pantallas táctiles** para crear un mini GPS para el auto que marque su velocidad. Existen dos tipos de pantallas táctiles: las resistivas, que miden la resistencia mecánica al presionar el panel, y las capacitivas, que, con soporte **multitouch**, miden la distorsión de campo electromagnético provocado por el dedo conductor.

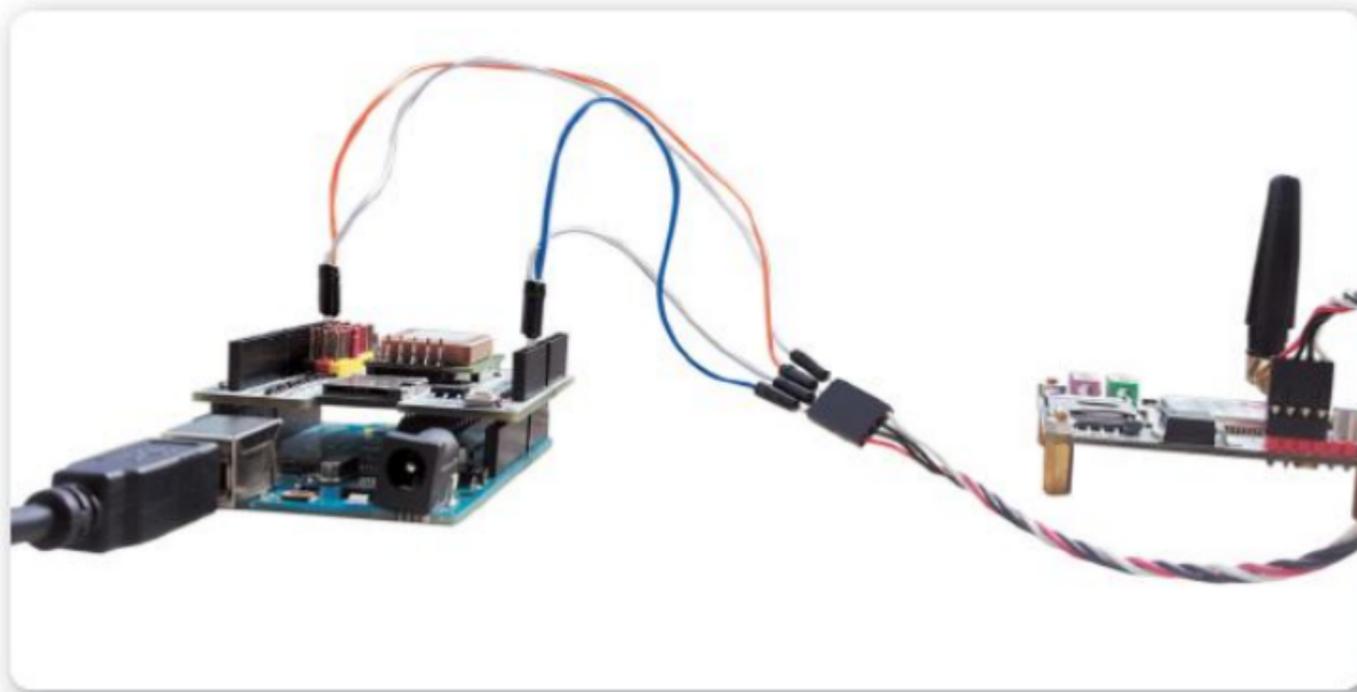


Figura 28. Arduino + shield GPS + módulo GSM para obtener la vista al cielo y, de esta manera, captar satélites más rápidamente.

Una práctica interesante es diseñar y crear nosotros mismos los circuitos integrados de lo que queremos armar. La Arduino UNO nos brinda la posibilidad de quitar el microcontrolador ATmega328 luego de programarlo. Se puede diseñar una placa que contenga solo lo necesario para ese trabajo específico y, por ejemplo, en este proyecto de telemetría, si se toma este camino, se puede pensar en chips como el **SIM908**, que incorpora las tecnologías GPS y GSM/GPRS juntas, y además de reducir el tamaño, obtendremos un decremento en el costo final.



Figura 29. El **SIM908** fabricado por la empresa **SIMCom** incorpora a la vez GPS y GSM/GPRS en un solo chip.

Una desventaja de Arduino frente a otras plataformas es la posibilidad de correr un **sistema operativo**, lo que para algunas aplicaciones puede resultar muy útil. Lo detallaremos en la sección siguiente.

Alternativas con otras plataformas

Cuando la complejidad del sistema por implementar supera las limitaciones de la Arduino, disponemos de alternativas de filosofía similar pero con un hardware más potente que nos ofrece una mayor capacidad de cómputo, procesamiento gráfico por hardware (GPU), mayor cantidad y velocidad de memoria RAM, mayor conectividad y la posibilidad de instalar un **sistema operativo**.



Figura 30. Esta es una de las placas de **Raspberry Pi**: el modelo B, que viene con 512 MB de RAM, dos USB y puerto ethernet integrado.

La **Raspberry Pi** es una de ellas. Dispone básicamente de una CPU ARMv6 @ 700 MHz, desde 128 hasta 512 MB de memoria SDRAM (según la versión) y un GPU VideoCore IV de Broadcom @ 250 MHz. En materia de **conectividad**, posee salida HDMI, RCA (video compuesto), DSI (para panel LCD), jack 3.5 mm (salida de audio), 1 o 2 puertos USB (según la versión), lector de tarjetas SD, puerto ethernet RJ-45 (según la versión) y varios pines de propósito general

(símil Arduino). Como vemos, ya desde un principio es no solo más potente, sino también más completa, ya que varias de las cosas que vienen por defecto demandarían el uso de shields o módulos en la Arduino para incorporarlas a nuestro sistema.

Gracias a este hardware, esta mini PC nos permitiría continuar desarrollando nuestro proyecto adicionándole otras cosas, como, por ejemplo, montar un **Web server** e incluir la API de Google Maps para que muestre su posición en tiempo real. También podríamos sumar una cámara para realizar **streaming de video** (aunque para eso no será suficiente el ancho de banda de subida que nos proporciona el módulo GPRS, como aclaramos antes; sin embargo, en este caso el límite es el módulo y no la placa).

Podríamos también guardar las posiciones en la memoria SD, a modo de **data logger** (esto también sería posible en nuestro caso, con la Arduino, gracias a que el shield GPS dispone del lector de tarjetas) y aprovechar su capacidad de cómputo para generar gráficos o estadísticas o calcular distancias contra una ubicación por determinar por el usuario.

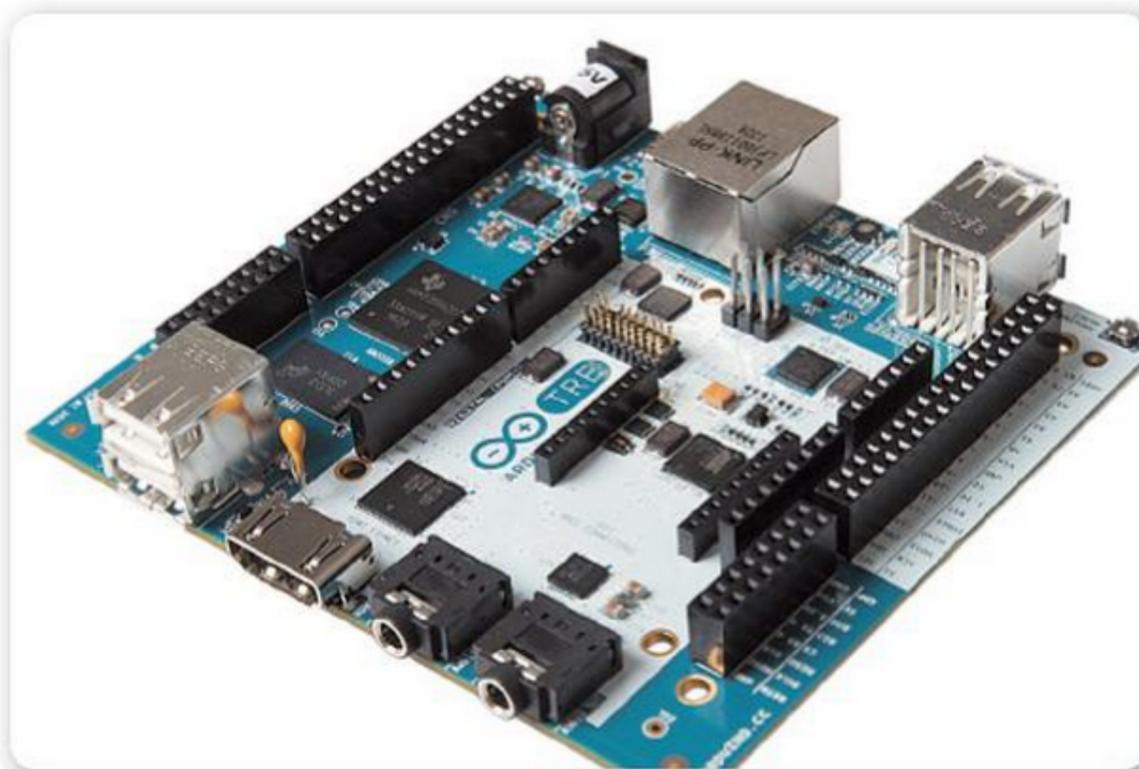


Figura 31. El futuro **Arduino TRE** . Desarrollado en conjunto por el equipo de Arduino y el de BeagleBoard, promete ser una plataforma muy interesante y completa.

Vale la pena destacar que esta plataforma es capaz de correr varios sistemas operativos, sobre todo, distintas distribuciones de **Linux** adaptadas específicamente para su hardware. Esto nos eleva aún más

las posibilidades, como establecer **conexiones SSH** con el dispositivo y poder controlarlo remotamente, o también –por qué no– diseñarlo como un sistema independiente, con interfaz gráfica y algún periférico de entrada. Podría transformarse fácilmente en la computadora de un auto (que no lo tenga, claro está) y utilizar el GPS como sistema de navegación, que a su vez puede seguir reportando su ubicación como en la idea original.

Como vemos, el abanico de posibilidades se agiganta gracias a este nuevo hardware que, en este caso, nos permite seguir construyendo sobre la idea original.

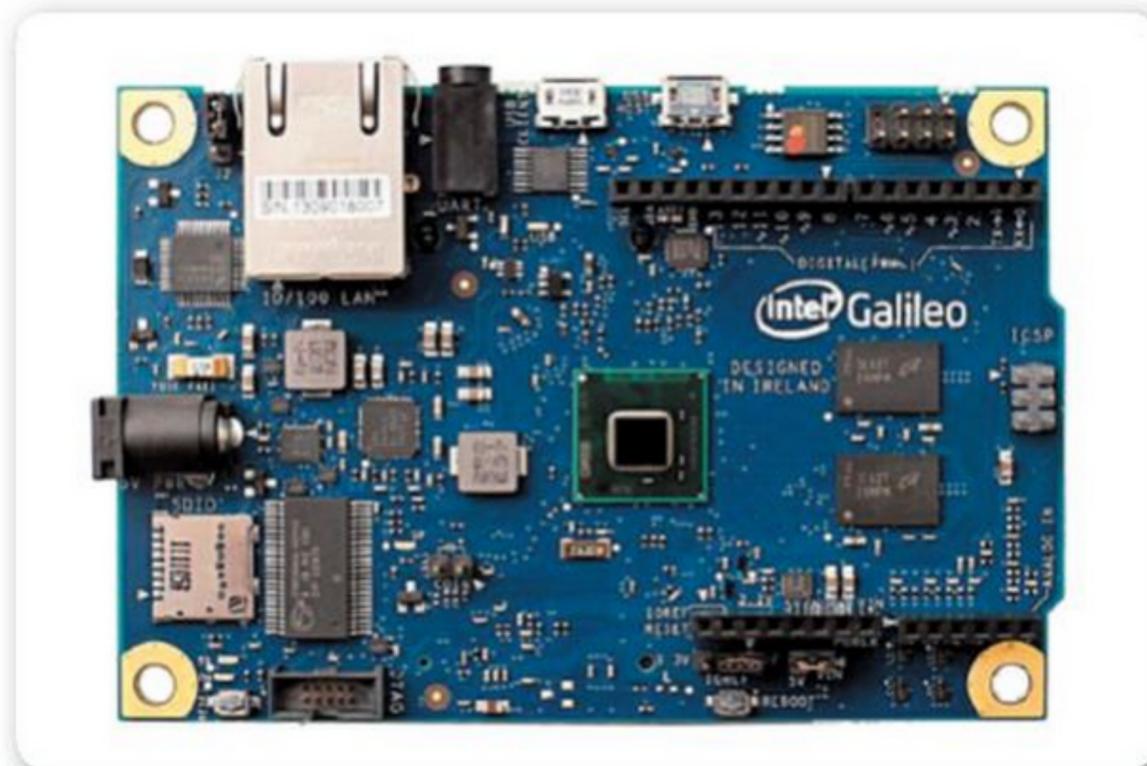


Figura 32. **Arduino Galileo** , la nueva placa Arduino con arquitectura Intel, cuyo lanzamiento acerca la arquitectura x86 al hardware libre.

La Raspberry PI no es única en su tipo, existen varias plataformas similares. Entre ellas, Arduino posee **Due** de inferior categoría, si hablamos de potencia, comparada, por ejemplo, con la Raspberry PI, pero que también utiliza un chip **ARM CortexM3 @ 84Mhz**, aunque las demás categorías se acercan más a la Arduino UNO que a la PI.

Sin embargo, Arduino pronto sacará al mercado dos nuevas versiones:

Arduino Galileo : utiliza un micro x86 de **Intel**, el Quark X1000 @ 400 MHz, 256 MB de RAM, conectividad y accesorios muy similares a la Raspberry, aunque agrega un conector mini PCI-E, el cual le añade interés. También tiene la gran ventaja de ser compatible con todas las aplicaciones y sistemas operativos hechos para x86.

Arduino TRE : integra el microprocesador ATmega32u4 (el mismo que utiliza Arduino Leonardo) para manejar todos los shields propios de la Arduino y, a su vez, ARM Sitara @1 GHz, con 512 MB de RAM y todas las conectividades y accesorios de la Raspberry sumadas a las de Arduino. O sea, esta plataforma será capaz de levantar un Linux, hacer todo lo que hemos descrito y, además, contará con un subsistema ya integrado del cual se encarga el ATmega, que le adiciona todas las herramientas que posee actualmente Arduino.

La última que mencionaremos es la **BeagleBone Black** . Desarrollada por BeagleBoard, posee el mismo ARM Sitara @ 1 GHz que la futura Arduino TRE (de hecho, TRE está siendo desarrollada por ambos equipos), 512 MB de RAM, de GPU un PowerVR SGX 530, conectividad y accesorios similares a sus pares, a los que agrega 2 GB de memoria flash integrada y dos coprocesadores ARMv7 @ 200 MHz, que utiliza para controlar los GPIO (E/S de propósito general) que incorpora.

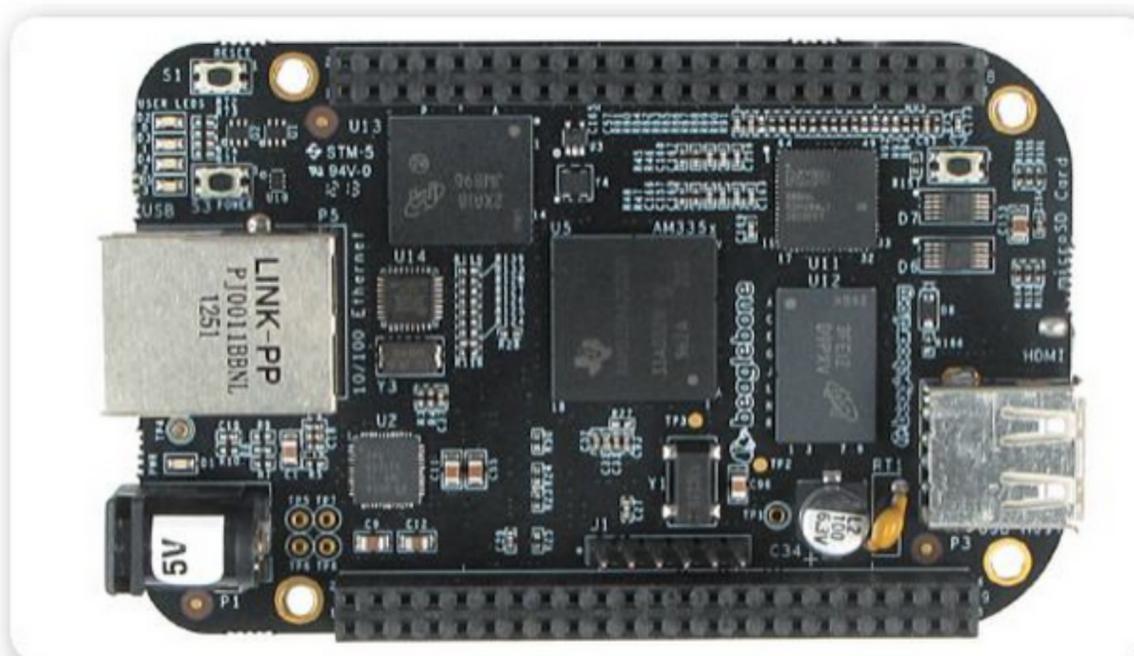


Figura 33. La **BeagleBone Black**. Un producto con muy buena relación precio/potencia y cuya comunidad ha ido creciendo.



LINUX Y EL HARDWARE LIBRE

Linux está disponible, en general, para todas las arquitecturas, y estas plataformas de desarrollo no escapan a ello. Todas poseen varias distros de **Linux** diferentes, que han sido optimizadas para funcionar específicamente con el hardware propio de todas ellas. Incluso, en algunas de ellas, pueden instalarse versiones de PC de escritorio o también de teléfonos móviles, como Android.

Conclusiones

El propósito de este capítulo fue enseñar y estimular la **creatividad** del lector, no solo explicando paso a paso un proyecto particular para implementar, sino también haciendo hincapié, a

EXISTEN
ALTERNATIVAS
PARA CUANDO LA
COMPLEJIDAD SUPERA
LA DE ARDUINO

cada momento, en las distintas variedades y **modificaciones** que se pueden hacer para adaptar nuestra enseñanza a los fines del lector. Como ya vimos, la **telemetría** es un concepto amplio para medir a distancia y, en la actualidad, hay un sinfín de aplicaciones en las que esto es necesario. Si bien en este caso elegimos, como magnitud por transmitir, la posición obtenida mediante **GPS**, son muchas las alternativas que incluso se pueden agrupar para formar un proyecto cada vez mayor.

Combinar distintos proyectos armados por varias personas en uno solo es un desafío que no se debe dejar pasar y que, incluso, si se crea el adecuado **equilibrio entre costo, calidad y tiempo**, puede significar un futuro sistema que se puede comercializar y **obtener ingresos** directos de nuestro esfuerzo y conocimiento.

Dependiendo de la aplicación particular, vamos a tener que tomar decisiones y elegir las distintas tecnologías a usar, como así también la **plataforma de desarrollo**, ya que, como aclaramos antes, Arduino a veces puede ser limitada y necesitar “ayuda externa”. A modo de aprendizaje, se pueden ir probando todas las plataformas y sacar conclusiones propias sobre cada una. Lo que hay que evitar es creer que los microcontroladores PIC (hoy menos usados), la plataforma de hardware libre Arduino, la Raspberry PI, la BeagleBone o cualquier nuevo chip que salga pueden realizar todas las tareas, y que no nos



SHIELDS Y MÓDULOS

Todos los modelos poseen sus shields y módulos propios (“capas”, en el caso del **BeagleBone**) como la Arduino. En general, los módulos pueden utilizarse en cualquier plataforma; solo se debe considerar el voltaje de alimentación, 3.3 V o 5 V. Incluso los **shields** específicos para Arduino se pueden utilizar en la Raspberry PI, añadiendo unas placas que se encargan de adaptarlos.

perdamos de abrir nuestra mente a nuevas plataformas por el miedo a la diversificación. Este es un punto importante, sobre todo pensando en el comercio, ya que no existe ninguna “plataforma ideal”, sino que **cada solución será particular** para algún problema dado y, por lo tanto, deberemos tener la capacidad de resolverlo con la mejor eficiencia, analizando todas las soluciones posibles.

Lo bueno de usar Arduino y microcontroladores que podemos programar mediante software es que luego es posible reutilizar nuestro conocimiento adquirido e, incluso, compartirlo con otras personas. Hoy en día, Arduino es lo que es gracias a su **filosofía de hardware libre** y a una gran comunidad dispuesta a ayudarnos cuando se nos presenta algún tipo de problema.



RESUMEN



En este capítulo, presentamos un proyecto que concentra muchos de los conocimientos adquiridos en los capítulos anteriores: elegimos un sistema de telemetría, pero no lo construimos a partir de circuitos armados desde cero, sino de una plataforma Arduino. Incluimos también la explicación de los módulos necesarios para implementar el sistema de telemetría, medir la posición con GPS y transmitirla a través de mensajes de texto. Las aplicaciones de un sistema de telemetría son múltiples y pueden encontrarse tanto en el entorno hogareño como industrial, por lo que quizás algún interesado en el tema decida continuar construyendo luego modelos más complejos y ampliados para comercializar o, simplemente, para seguir aprendiendo.

Actividades

TEST DE AUTOEVALUACIÓN

- 1 ¿Cuál es la definición y el origen etimológico de la palabra **telemetría**?
- 2 ¿Qué **magnitudes** pueden ser medibles por un sistema y por qué **canales** puede transmitirse dicha información obtenida?
- 3 Identifique algún ejemplo o aplicación de un **sistema de telemetría** descrito en este capítulo.
- 4 ¿Cuáles son los dos modos en que podemos **introducir las mediciones** de las magnitudes físicas utilizando Arduino?
- 5 ¿Cuáles son las ventajas y limitaciones que puede identificar al utilizar **módulos** y **shields** comerciales?
- 6 ¿Cuáles son los módulos más económicos para el envío y recepción de datos?
- 7 Enumere las principales características de la **Arduino UNO**, elegida para llevar a cabo este proyecto.
- 8 Enumere las principales características de la **GPS Shield-Fastrax UP501**, elegida para llevar a cabo este proyecto.
- 9 ¿Cuáles son las recomendaciones sugeridas en este capítulo para probar el **módulo GSM**?
- 10 ¿Qué plataforma se sugiere utilizar cuando las complejidades del sistema superan los alcances de la Arduino?



PROFESOR EN LÍNEA



Si tiene alguna consulta técnica relacionada con el contenido, puede contactarse con nuestros expertos: profesor@redusers.com



Raspberry Pi

Desde la salida de la Raspberry Pi, se ha generado una proliferación de usuarios alrededor del mundo que buscan experimentar con este dispositivo en distintos ámbitos.

En este capítulo, conoceremos las características del hardware y todo lo relacionado con el software para mejorar su rendimiento, además de su interacción con Arduino.

▼ Raspberry Pi148	Consideraciones sobre Python 162
Características del hardware 149	
Rendimiento 153	▼ Salida - entrada GPIO.....162
	WiringPi..... 166
▼ Software necesario.....154	▼ Ethernet y Raspberry Pi.....168
Raspbian..... 155	▼ Tecnologías VNC y SSH.....169
▼ Carga del sistema operativo..156	▼ Resumen.....171
▼ Overclock157	▼ Actividades.....172
▼ Integración de Raspberry y Arduino.....158	



Raspberry Pi

En la actualidad se han desarrollado muchas placas y dispositivos destinados al aprendizaje y al desarrollo. Así como Arduino, Raspberry Pi puede ubicarse en este segmento de la electrónica.

A pesar de esto, Arduino y Raspberry Pi poseen diferencias. Con ambas se pueden controlar entradas y salidas, pero Arduino es apta para resolver circuitos de automatización, control y robótica. Además, Raspberry Pi es más versátil, pues se presenta como un motherboard de pequeño tamaño.

Es interesante destacar que Raspberry Pi se asocia al concepto de **open source**, lo que nos garantiza trabajar con software legal y contar con información útil, dispuesta de forma gratuita en internet.

Se han generado, al igual que alrededor de Arduino, comunidades virtuales que intercambian y comparten proyectos, que pueden ser el punto de partida para nuestro trabajo.



Figura 1. La Raspberry Pi, una poderosa herramienta para desarrollar aplicaciones.

Pero ¿qué podemos hacer con una Raspberry Pi? Aunque la respuesta podría ser muy extensa, en síntesis, podemos decir que nos permitirá integrar multimedia con el control de elementos externos, a través de sus entradas y salidas, y nos posibilitará, además, comunicar datos.

Al igual que en Arduino, existen placas prefabricadas para apoyar nuestros fines, pero quizás lo mejor sea desarrollar hardware y software, es decir, complementar los roles de técnico electrónico y de programador para obtener los mejores resultados.

Figura 2. En esta imagen, vemos el logo que caracteriza a la Raspberry Pi.



Características del hardware

En el mercado existen dos modelos de placas Raspberry Pi, denominados **A** y **B**. Cabe aclarar que la diferencia de precio entre ambas es muy poco significativa, aunque sí lo son sus prestaciones, siendo más potente el modelo **B**.

El modelo **B** integra un **SoC (System On Chip)**, es decir, un chip que contiene varios módulos: CPU **ARM1176JZF-S** que corre a 700 MHz y GPU **Broadcom VideoCore IV**.

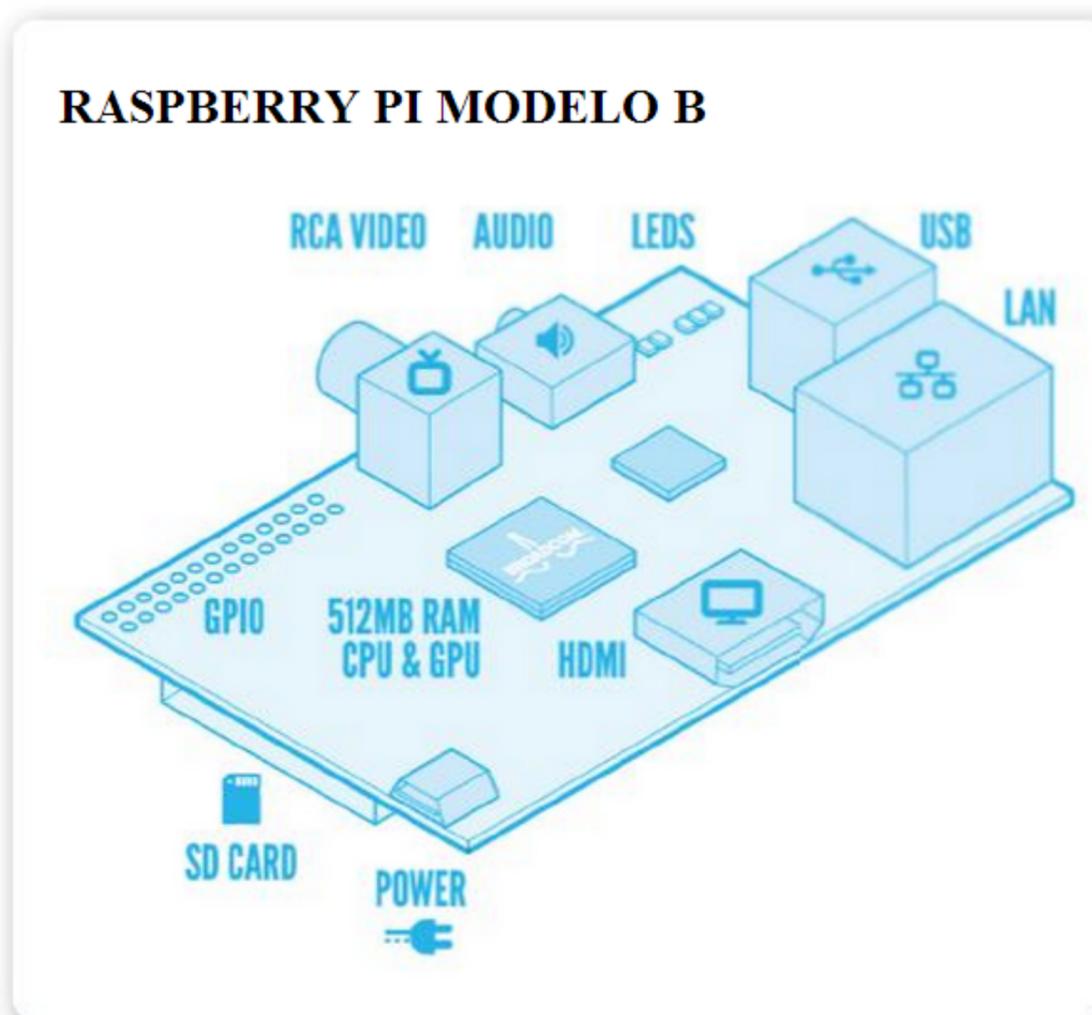


Figura 3. Esquema de la placa Raspberry Pi modelo **B**.

El SoC forma parte de una tendencia actual de diseño que optimiza recursos, permitiendo obtener buenos rendimientos en un espacio reducido. Además, las cantidades de memoria RAM (del tipo SDRAM) llegan hasta los 512 MB.

Esta placa puede ofrecernos salida de audio y video por HDMI, aunque el video también puede salir por RCA, para los sistemas PAL y NTSC. En cuanto al audio, presenta una salida para un conector de 3.5 mm.

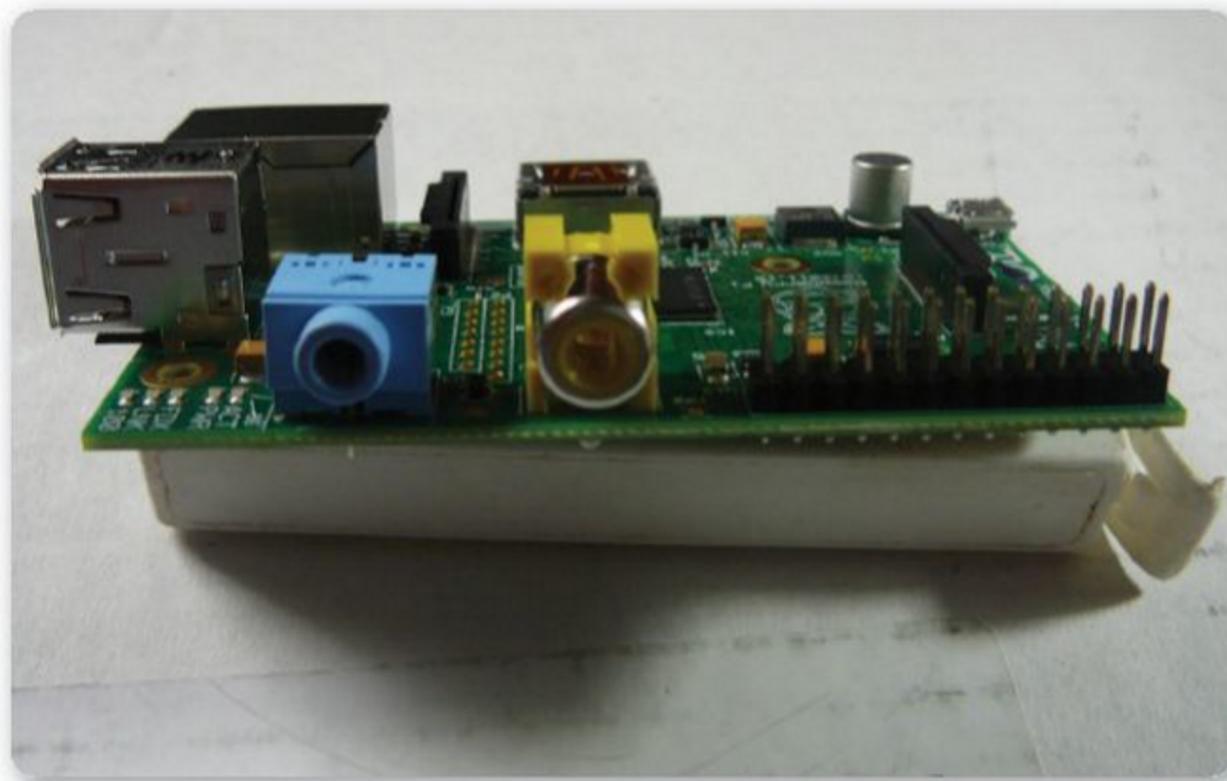


Figura 4. Las conexiones de audio y video permitirán realizar aplicaciones multimedia.

Además, posee dos puertos USB para realizar la conexión de periféricos; de esta forma, su operación es similar a la de una computadora personal.

Al conectarle un monitor mediante HDMI, la interfaz gráfica no es distinta de cualquier PC, ya que veremos una pantalla similar a un escritorio de una distribución GNU/Linux.



SOFTWARE



Es importante tener en cuenta que, en la actualidad, Raspberry Pi se desempeña de manera muy eficiente utilizando software diseñado bajo los sistemas GNU/Linux. Por ejemplo, trabaja muy bien con Raspbian, una aplicación derivada de la distribución Debian.

Es importante considerar que para almacenar su sistema operativo utiliza una memoria de tipo SD. Por otra parte, posee un conector RJ-45, mediante el cual podremos realizar la conexión a una red de tipo Ethernet. Si bien no posee un adaptador Wi-Fi, podemos usar los puertos USB para conectar una placa Wireless externa.

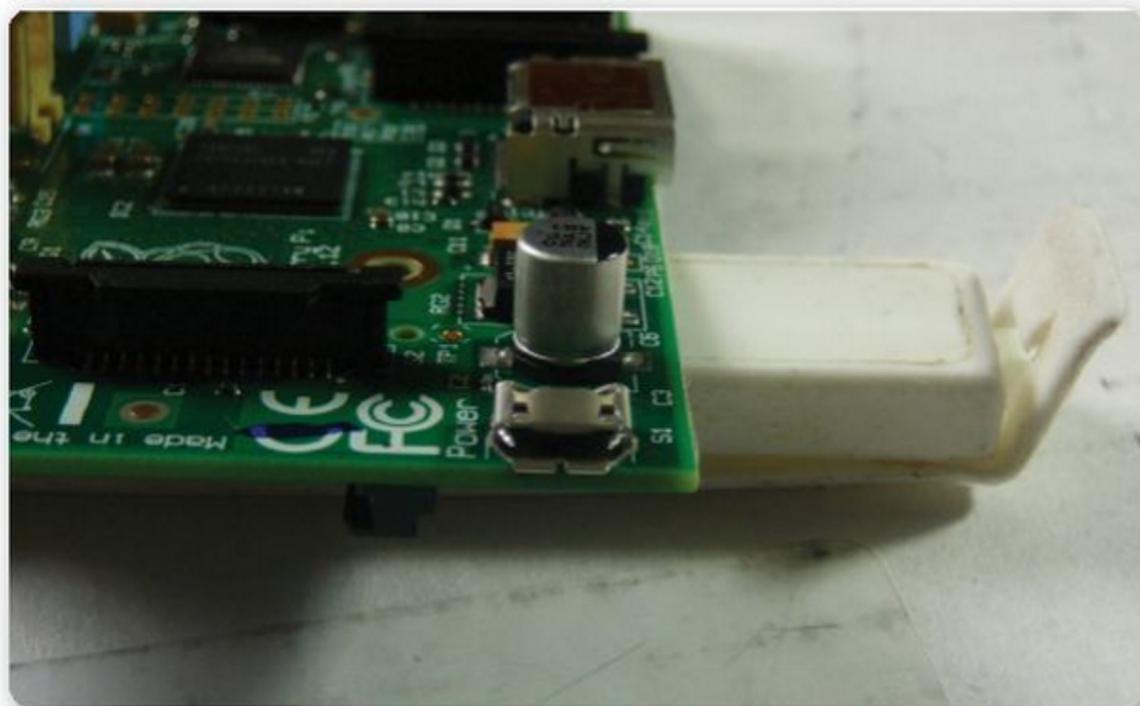


Figura 5. Se utiliza un conector mini USB para poder alimentarla.

Generalmente buscaremos conectar y controlar las entradas y salidas de la placa Raspberry Pi; para ello, posee **GPIO** (entradas y salidas de propósito general), **SPI** (bus para comunicación serial) y comunicación por **UART** (transmisor - receptor asincrónico universal), todo lo que le permite comunicarse con el exterior.

Debemos considerar que la potencia de estos pines es reducida y se deben adaptar eléctricamente a partir de circuitos, por ejemplo, aplicando **optoacopladores** y **transistores MosFet** o relés, entre otros elementos.



CONEXIÓN A LA RED



Ethernet se presenta como un estándar de transmisión (IEEE 802.3) de datos para redes de área local. Se diferencian en uno de los campos de la trama de datos. Sin embargo, las tramas Ethernet e IEEE 802.3 pueden coexistir en la misma red. Mediante el uso de este estándar es que la placa Raspberry Pi es capaz de conectarse a la Web.

Todo lo que hemos mencionado se encuentra en una superficie menor que 50 cm² y con una potencia de 3.5 Watts, con una alimentación de 5 V a través de una entrada Micro USB.

Debemos controlar que la fuente de energía que utilizamos proporcione la corriente necesaria (de, aproximadamente, 700 mA); de lo contrario, la placa se reiniciará por falta de energía.

En muchos casos, siempre que tengamos la alimentación eléctrica, podemos utilizar un cargador de teléfono o de otros dispositivos, como un GPS.



Figura 6. La Raspberry Pi en un gabinete que la protege y le permite cómodamente conectar todo lo necesario.

Consumo eléctrico

Luego de analizar el hardware de la Raspberry Pi, una pregunta importante que surge es: ¿qué relación existe entre el rendimiento de esta placa y su consumo eléctrico?

Al ver esta pequeña pero potente placa, un problema a tener en cuenta será el calor generado. Además, si queremos obtener un mayor rendimiento, tendremos que aumentar la velocidad del procesador, mediante el overclock.

Ahora bien, esa mayor frecuencia generará inevitablemente un mayor consumo eléctrico (que debe ser contemplado en la fuente que usaremos) y también aumentará la temperatura. Para esto

último, debemos considerar soluciones de refrigeración; por ejemplo, un kit denominado **Raspberry Heat Skin Kit**, constituido por pequeños disipadores para el SoC, el GPU y el regulador de tensión. Sin este dispositivo, la temperatura de trabajo podría pasar de los 60 °C, lo que sería peligroso para el hardware.

A pesar de esto, en esta placa el **overclocking** es necesario, pues si necesitamos usarla como una PC, se necesita mayor velocidad que la implementada de fábrica.

Por esta razón, para llevar adelante el overclocking, también se realiza un **overvolting**, que consiste en otorgarle mayor alimentación eléctrica. Por este motivo, es aconsejable tener una fuente que pueda entregar una corriente de hasta 1 Amper.

LA PLACA, LA FUENTE
Y LA MEMORIA SD
SON ELEMENTOS
QUE MEJORAN EL
RENDIMIENTO



Rendimiento

Para obtener un mejor rendimiento de la Raspberry Pi, tendremos que combinar tres elementos: la placa, la fuente de alimentación y la memoria SD. Ya mencionamos que la placa B posee un mejor rendimiento y sabemos que necesitamos contar con una fuente de alimentación que satisfaga nuestras necesidades. Solo nos queda referirnos a la memoria SD.

Al seleccionar una memoria SD, se toma como criterio su capacidad, pero también es importante la velocidad de acceso que posee.

Un detalle importante es que las tarjetas SD miden su velocidad de escritura de datos en base a la velocidad máxima. Las de 40x escriben a 6 MB por segundo; las de 80x, a 12 MB por segundo; las de 100x a 15 MB por segundo; etcétera.



FUENTE DE ALIMENTACIÓN

La fuente de alimentación debe entregar una corriente mínima de 700 mA, con una tensión de 5 V. En cambio, para aplicar técnicas de **overclock** a la placa, será recomendable contar con unos 300 mA adicionales, pensando también en la vida útil de la fuente de alimentación.



En cambio, el tipo de memoria **SDHC** mide la velocidad mínima de escritura, por lo tanto, una clase 2 escribe a 2 MB por segundo como mínimo y una clase 6 escribirá a 6 MB por segundo.



Figura 7. La memoria SD es uno de los elementos indispensables para obtener un mejor rendimiento.

Software necesario

Cuando tengamos nuestro primer contacto con la Raspberry Pi, veremos que no posee un sistema operativo, por lo que tendremos que instalarlo en la memoria SD.

RASPBERRY PI
CARECE DE SISTEMA
OPERATIVO Y HAY QUE
INSTALARLO EN LA
MEMORIA SD

Todo el software que utilizaremos es libre, ya que se basa en distribuciones GNU/Linux, como Raspbian (una versión de la distribución Debian).

Para instalar el sistema operativo, existen dos caminos: el primero es instalar el software **BerryBoot**, que permite arrancar la placa y luego bajar el sistema operativo.

La segunda opción es cargar en la tarjeta **Noobs**, una aplicación que, al ser descomprimida, nos permitirá elegir entre algunos sistemas

GNU/Linux para instalar.

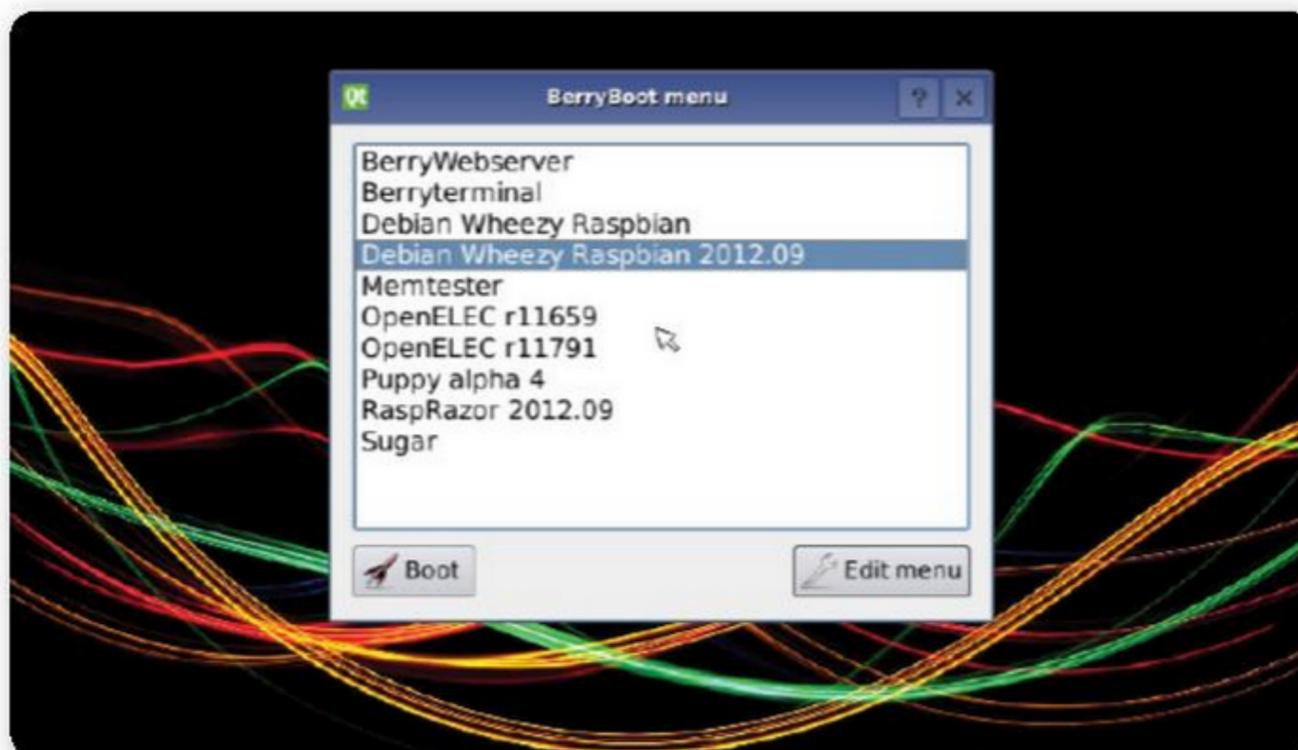


Figura 8. BerryBoot nos asistirá en la tarea de instalar un sistema operativo en la Raspberry Pi.

Raspbian

Pero, ¿qué podemos decir del Raspbian? Se trata de un sistema basado en Debian, que es una distribución GNU/Linux preparada para trabajar con procesadores ARM, los cuales son ideales para ejecutar aplicaciones de baja potencia (por ejemplo, en sistemas de aplicaciones portátiles).

Raspbian fue diseñado para Raspberry Pi, procurando obtener el máximo provecho al hardware disponible en la placa y utilizando, para ello, paquetes de software que se encuentran en la distribución **Debian ARM Wheezy Hard Float**, compilados para trabajar con **procesadores ARM v6**.



OPEN SOURCE

Uno de los creadores de Raspbian, Mike Thompson, pensaba que, al mantener Raspbian similar a Debian, se reducía el riesgo de que solo dos personas trabajen en el proyecto. De esta forma, si los desarrolladores principales dejan de trabajar en cualquier momento, el trabajo quedará en diversos servidores alrededor del mundo. Así, cualquiera con el conocimiento de compilar en Debian puede fácilmente seguir manteniéndolo.

Raspbian es una excelente opción para la placa Raspberry Pi, pues nos permite aprovechar al máximo el hardware, teniendo en cuenta la velocidad de CPU y la memoria RAM disponible. Podemos descargar Raspbian desde www.raspberrypi.org.



Figura 9. En la dirección www.raspbian.org/RaspbianImages encontraremos diversas opciones para descargar e instalar Raspbian.



Carga del sistema operativo

Tanto si realizamos la carga del sistema operativo con **BerryBoot** o con **Noobs**, debemos tener en cuenta algunas consideraciones.

Como sabemos, BerryBoot es un programa que actúa como **instalador y gestor de arranque o boot**, lo que nos permite tener varios sistemas operativos funcionando en Raspberry. Este programa se instala al copiarlo en la SD que utilizaremos con la placa. Gracias a BerryBoot, podremos instalar Raspbian o cualquier otro sistema de forma sencilla.

Entre las ventajas de este método de instalación del sistema operativo, encontramos que no es necesario estar conectados a internet permanentemente.

Desde otra computadora, bajaremos el archivo comprimido del programa; por ejemplo, podemos encontrarlo como `berryboot-20130213.zip` y lo descomprimimos en la memoria SD,

la cual debería estar formateada con el sistema de archivos FAT32. Luego, introducimos la memoria e iniciamos la placa Raspberry Pi. Posteriormente, seguimos las instrucciones presentadas por el asistente de instalación.

Por otra parte, si deseamos utilizar Noobs, debemos considerar que la placa esté conectada a internet. Por esta razón, debemos tener el modelo B, ya que este cuenta con un puerto Ethernet.

Una vez que haya sido descargado el archivo de instalación, ya no será necesario mantener la conexión. Al igual que con BerryBoot, tendremos que descomprimirlo en la tarjeta SD, la que debería contar con una capacidad de, al menos, 4 GB.

En las últimas versiones de Noobs, podremos seleccionar entre los sistemas operativos Raspbian y OpenELEC, entre otras opciones.

Cuando la instalación del sistema haya finalizado, Noobs permanecerá en la memoria SD. De esta forma, cada vez que se realiza el proceso de arranque podremos presionar la tecla SHIFT para cambiar de sistema operativo y acceder a las opciones de configuración del arranque.

CON BERRYBOOT
NO ES NECESARIO
ESTAR CONECTADO A
INTERNET DE FORMA
PERMANENTE



Overclock

La posibilidad de realizar overclock es importante cuando trabajamos con una placa Raspberry Pi. Por esta razón, describiremos el siguiente procedimiento para llevarlo a cabo.

Para comenzar, en la terminal de comandos, escribimos lo siguiente:

```
S sudo raspi-config
```

Así, accederemos a una pantalla que nos presenta una serie de opciones, entre las cuales seleccionamos `overclock`. Gracias a ella, podremos configurar este parámetro.

Una vez que seleccionamos la opción, aparecerá la pantalla en donde elegiremos los parámetros de overclock que se aplicarán. Es importante saber que, para movernos dentro de las opciones presentadas en la pantalla, debemos utilizar las flechas de dirección del teclado.

Para continuar, seleccionamos la opción deseada y presionamos Ok.

Entre las alternativas presentadas, encontramos None, la opción predeterminada, que supone no aplicar overclocking.

Si tenemos una fuente que entregue por lo menos 900 mA, es recomendable seleccionar la configuración Turbo, pues nos llevará a trabajar a una frecuencia de 1000 MhZ.

Si luego de elegir esta opción, la placa no inicia, no deberíamos preocuparnos ya que difícilmente se haya dañado la placa y podremos reconfigurarla.

Para enfrentar esta tarea, tendremos que desconectar la alimentación eléctrica de la placa, esperar algunos segundos y reiniciarla manteniendo presionada la tecla SHIFT.

A continuación, nos encontraremos con el siguiente mensaje:

Ok, checking if shift key held down:Yes. Not switching scaling governor .

Esto significa que no ha considerado la configuración anterior e iniciará con los parámetros predeterminados, es decir, a 700 MhZ.

Luego, debemos repetir el procedimiento anterior y seleccionar otra opción, hasta lograr una configuración exitosa.



Integración de Raspberry y Arduino

En algunas ocasiones puede ser necesario lograr una interacción entre Arduino y Raspberry Pi. Si bien la Raspberry Pi puede cumplir con la función de terminal para realizar la programación y, también, oficiar de interfaz con el usuario, necesita ser acompañada de circuitos electrónicos para desempeñar funciones que están disponibles en una placa Arduino.

Para lograr esta interacción, podemos utilizar un protocolo llamado firmata, que permite comunicar el microcontrolador con una computadora.

Otra opción interesante es mostrar datos capturados mediante el monitor de una computadora. Para ello, en Raspbian existe una versión del protocolo llamado `pyfirmata`, que nos permitirá crear una interfaz para utilizar Python en nuestra Raspberry Pi.

En cuanto al sensado, utilizaremos la **sonda LM35**, que entrega 10 mV por cada grado centígrado de variación. La idea es convertir esa variación en un valor analógico, conectando este sensor a un canal analógico de Arduino. Recordemos que Arduino UNO, por ejemplo, posee un microcontrolador **ATMEGA 328**, y su conversor analógico digital es de 10 bit con un método de conversión por aproximaciones sucesivas.

Otra forma de integración consiste en que desde la Raspberry Pi se pueda manejar el brillo de un led con un control deslizante en la pantalla y, de esta forma, monitorear la temperatura.

En Raspbian ya se encuentra incluido el IDE de Arduino, pudiendo ejecutarlo con el siguiente comando:

```
$ sudo apt-get install arduino
```

Al abrir el IDE por primera vez, tendremos que destinar una carpeta para almacenar los programas realizados en Arduino.

Un problema que puede surgir es que la placa Arduino se conecte a la Raspberry Pi mediante el puerto USB, pero quizás posea menor corriente a la necesaria, ya que la placa no dispone de mucha energía. En estos casos, lo que es aconsejable es alimentar la placa Arduino de

RASPBERRY Y
ARDUINO PUEDEN
COMPLEMENTARSE
E INTERACTUAR
PERFECTAMENTE



OVERCLOCK

Se trata de un anglicismo de uso muy habitual que, literalmente, significa “sobre el reloj”, es decir, aumentar la frecuencia de reloj de la unidad central de procesamiento. La práctica, conocida como **overclocking**, supone alcanzar un mayor rendimiento para un componente electrónico (por encima de las especificaciones del fabricante).



forma externa, con su propio conector, para que, en el momento de cargar un programa o cualquier operación que necesite de comunicar datos, tengamos la energía suficiente.

Para ejecutar el protocolo firmata y su versión para la Raspberry Pi, el pyfirmata, ejecutamos los siguientes comandos:

```
S sudo apt-get install python-serial mercurial
S hg clone https://bitbucket.org/tino/pyfirmata
S cd pyfirmata
S sudo python setup.py install
```

El código del programa desarrollado en Python será el siguiente:

```
import pyfirmata
from Tkinter import *

board = pyfirmata.Arduino('/dev/ttyACM0')
iter8 = pyfirmata.util.Iterator(board)
iter8.start()

pin0 = board.get_pin('a:0:i')
pin3 = board.get_pin('d:3:p')

while pin0.read() is None:
    pass
def get_temp():
    pin0.read() * 5 * 100)
    label.config(text = label_text)
    root.after(500, get_temp)
def set_brightness(x):
    pin3.write(float(x) / 100.0)
    pin3.write(0)
board.exit()
root = Tk()
root.wm_protocol("WM_DELETE_WINDOW",cleanup)
scale = Scale(root,
```

```

        command = set_brightness,
        orient = HORIZONTAL,
        length = 400,
        label = 'Brightness')

scale.pack(anchor = CENTER)
label = Label(root)
label.pack(anchor = 'nw')
root.after(500, get_temp)
root.mainloop()

```

En síntesis, lo que hace este programa en Python es establecer la comunicación con la placa Arduino, es decir, que la Raspberry Pi la reconozca.

Además, como las entradas serán desde Arduino pero usadas por Python, se declaran las entradas de información que se usarán y luego se mostrarán.

También se da un tiempo de espera para que Arduino comience a enviar datos reales, antes de entrar en el programa principal. De esta forma, no se desborda el buffer de la comunicación serial de Raspberry Pi.

Al igual que en otros lenguajes, se definen funciones que permitirán resolver distintas situaciones problemáticas. Por ejemplo, la función `get_temp()` se encarga de tomar la temperatura y pasarla a grados centígrados.

En Arduino, los valores analógicos serán establecidos de 0 a 1023, mientras que, en Raspberry Pi, se hará entre 0,0 y 1,0.

Con la función `set_brightness(x)` se realiza la acción del deslizador, convirtiendo los valores de 0 a 100 para luego escribirlos en el pin 3 de Arduino y así controlar su brillo, a partir de la modulación por ancho de pulso, conocida como PWM.

Para realizar la creación de la interfaz gráfica de usuario o GUI, se utiliza **Tkinter**, que creará la barra deslizable de 400 px, que funcionará como un potenciómetro virtual.

Tkinter es un módulo que ya viene integrado en Python y es utilizado para crear una interfaz gráfica de manera sencilla.

PARA QUE ARDUINO
Y RASPBERRY PI
INTERACTUEN,
PODEMOS USAR EL
PROTOCOLO FIRMATA



Consideraciones sobre Python

Es importante destacar algunos conceptos sobre Python, pues se trata de un buen complemento para poder desarrollar en la placa Raspberry Pi.

Python es un lenguaje de programación interpretado, es decir, que está diseñado para que lo ejecute un intérprete, sin necesidad de un compilador.

Los lenguajes interpretados pueden ser llamados **script**, lo que significa que se presentan como programas sencillos, almacenables en un archivo de texto. Generalmente, estos archivos pueden ser utilizados para combinar componentes diversos o interactuar con el usuario. Para esta última tarea, pueden utilizar los **Shells**, que permitirán crear una interfaz gráfica.

Al tratarse de un lenguaje interpretado, Python ofrece una sintaxis bien simple y clara, fácil de comprender con una sencilla base de programación.

Al instalar el sistema Raspbian, se presentará un icono en el escritorio, llamado IDLE 3. Este icono nos ofrece acceso a un editor para Python.

Gracias a esta herramienta se pueden crear las aplicaciones en este lenguaje.



Salida - entrada GPIO

Sabemos que la placa Raspberry Pi nos permitirá controlar dispositivos periféricos de entrada y salida, a partir de sus GPIO, o entradas y salidas de propósitos generales.



RASPBERRY PI EXPANSION BOARD



Es una placa de expansión para la Raspberry Pi que ha sido pensada para poder explotar todas las posibilidades de la Raspberry Pi a nivel de hardware. La placa incorpora un conector para alimentar la placa y la Raspberry Pi. Por este conector se tiene que proporcionar un mínimo de 7 V y un máximo de 12 V. Esta entrada está conectada a un regulador lm323 que proporciona de 5 V a 3 A.

Una sencilla descripción de estos puertos puede ser conveniente, principalmente para conocer las posibilidades con las que contamos.

La cantidad de estos pines (GPIO) fue modificándose con los sucesivos modelos y seguramente evolucionarán en los próximos lanzamientos, pero podemos decir que contamos con algunas entradas y salidas digitales (salidas tipo PWM).

También podemos encontrar, para la comunicación de datos, un puerto I2C (dos hilos), uno SPI (4 hilos) y una UART.

Es importante mencionar que estos pines destinados a las comunicaciones, si no son utilizados para esos fines, pueden ser usados como entradas o salidas digitales.

Un detalle negativo a tener en cuenta es que no poseen una entrada analógica; es decir que, si queremos sensor, por ejemplo, la luz con una resistencia LDR, tendremos que recurrir a una placa externa como Arduino o a una placa de expansión para la Raspberry Pi, como la **Raspberry Pi Expansion Board**. Su aplicación aumentará considerablemente las capacidades de la placa base, ya que posee las siguientes características:

- Conversor de niveles lógicos bidireccional.
- ADC de 8 canales y 10 bits MCP3008 controlable mediante SPI.
- Relé a 5 V que soporta cargas de hasta 5 A.
- Alimentación externa.
- Regulador LM323 (llega hasta 3 A).
- Led.
- Un sector para realizar pruebas, similar a una placa experimental.
- Pines de comunicación SPI, I2C y UART, disponibles para usar.
- Puede montarse sobre la parte superior de la Raspberry Pi.

Es decir, con este complemento podemos solucionar el problema de los puertos y, principalmente, la carencia de una entrada analógica.

Para utilizar estos puertos, en los sistemas GNU/Linux, se pueden realizar dos métodos diferentes. Un método consiste en la creación de un acceso de tipo de archivo para el periférico en el sistema de archivos. El otro método supone escribir/leer la dirección base de la memoria asignada a la GPIO o módulo en los punteros. Estas ubicaciones de memoria se pueden encontrar en la hoja de datos del **BCM2835**.

El método más sencillo es el primero, y sobre él trabajaremos.

Desde la consola de comandos, podemos acceder a crear estos archivos, las líneas que necesitaremos para realizarlo son las siguientes:

```
echo 17 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio17/direction
echo 1 > /sys/class/gpio/gpio17/value
echo 0 > /sys/class/gpio/gpio17/value
```

Con este código tendremos acceso, en este caso, al pin 17. La primera línea otorga el acceso a este puerto; con la segunda línea, le damos la función de salida, ya que también puede ser utilizado como una entrada.

En las líneas siguientes, se escriben los valores lógicos para encender o apagar la salida, con un 1 y 0 lógicos, respectivamente.

```
echo 17 > /sys/class/gpio/unexport
```

Con este último comando, podemos deshacer el archivo creado para ese puerto.

En algunas ocasiones, puede ser necesario trabajar con algún lenguaje de programación como Python, ya que resultará muy fácil de interpretar y, además, posee una librería para poder operar los GPIO. Para instalar esta librería, podemos escribir el siguiente comando:

```
wget 'http://downloads.sourceforge.net/project/raspberry-gpio-python/RPi.GPIO-0.5.4.tar.gz'
```

Una vez descargado el archivo, tendremos que descomprimirlo. Lo hacemos con el siguiente comando:

```
tar zxvfRPi.GPIO-0.5.4.tar.gz
```

Luego, ingresamos en el archivo que hemos descomprimido:

```
cd RPi.GPIO-0.5.4/
```

Para continuar, procedemos a instalar la librería con los siguientes comandos:

```
sudo apt-get install python-dev
sudo python setup.py install
```

Para realizar un programa en Python, creamos un archivo con un nombre –en este ejemplo, blink–, luego escribimos el siguiente código. Debemos notar que los archivos para este lenguaje se denominarán con la extensión `.py`.

```
Sudo nano blink.py
```

Ahora observemos el código de nuestra aplicación de ejemplo en Python, que es el siguiente:

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.OUT)
GPIO.setup(27, GPIO.OUT)
def blink():
    print "Comienzo"
    iteracion = 0
    while iteracion < 30:
        GPIO.output(17, True) ## Enciendo el 17
        GPIO.output(27, False)
        time.sleep (1)
        iteracion=iteracion+2
    print "final"
    GPIO.cleanup() ## Hago una limpieza de los GPIO
    blink() ## llamada a la funcion blink
```

Esta aplicación importa la librería que ya tenemos instalada, luego se le otorga la funcionalidad a los pines, declarándolos como salida. Posteriormente, se genera una función que realizará el destello, donde se establece que los leds conectados cambian intermitentemente de estado, alternándose. La ejecutamos con la siguiente línea:

```
Ssudo Python blink.py
```

WiringPi

Se trata de una librería para utilizar GPIO (los puertos de entrada y salida de la placa Raspberry Pi), que se encuentra escrita en C para el **BCM2835** de la placa.

Para utilizarla, bastará con incluirla al comienzo de un programa en C. A continuación, veremos un ejemplo de programación de C, para lograr la intermitencia de una salida:

```
# Include <wiringPi.h>
int main (void)
{
    wiringPiSetup ();
    pinMode (0, OUTPUT);
    for (; ;)
    {

        digitalWrite (0, HIGH); delay (500);
        digitalWrite (0, LOW); delay (500);
    }
    return 0;
}
```

Gracias al **WiringPi**, podemos establecer comunicaciones con otros dispositivos como, por ejemplo, el **bus I2C**. Podemos decir sobre el I2C que funciona bajo la topología maestro-esclavo, siendo, en este caso, el rol de maestro para la placa Raspberry Pi y esclavos sus periféricos, que –por supuesto– deberán ser aptos para este bus.

Las señales presentes serán SDA (datos) y SCL (reloj). Este último sincronizará los datos.

El maestro será quien inicie la comunicación, genere la señal de reloj y controle cuando se envíen y reciban datos. Consideremos que puede haber más de un esclavo en la red I2C.

Además, el maestro se comunica con un protocolo de pregunta/respuesta, por lo que dialoga con un esclavo a la vez. Para poder localizar cada dispositivo, es necesario que cada uno tenga una dirección única dentro del bus.

Consideremos que los datos se organizan en transferencias; las transferencias comienzan con una señal de Start y terminan con otra de Stop. Los datos tendrán una longitud de 1 byte y terminan con bit de ACK. Los modos de operación soportados por esta librería se relacionan con el parámetro velocidad:

- Standard Mode , con una velocidad de hasta 100 kbit/s.
- Fast Mode , con una velocidad de hasta 400 kbit/s.
- Fast Mode Plus , con una velocidad de hasta 1 Mbit/s.
- High-speed Mode , con una velocidad de hasta 3.4 Mbit/s.

En la Raspberry Pi, entre los pines GPIO, tenemos la señal SDA y SCL. Como sabemos, para poder utilizar este bus, se debe tener instalada la librería WiringPi y, luego, se deben instalar las herramientas propias para el I2C.

```
apt-get install i2c-tools
```

Para poder acceder a esta librería utilizamos:

```
gpio load i2c
```

Para establecer la velocidad del bus a 400 kbps, de la siguiente forma:

```
gpio load i2c 400
```

Para que se cargue el módulo I2C, una vez que arranque el sistema, es necesario incluir los módulos `i2c-bcm2708` y `i2c-dev` en el fichero `/etc/modules`. Por otro lado, se debe comentar las entradas con un `#` del fichero `/etc/modprobe.d/raspi_blacklist.conf` :

- `blacklist spi-bcm2708`
- `blacklist i2c-bcm2708`

Muchos son los dispositivos que se pueden conectar con este bus; por ejemplo, podemos comunicar la placa Raspberry Pi con una placa Arduino.



Ethernet y Raspberry Pi

La placa Raspberry Pi tiene un puerto RJ 45 que nos permitirá conectarla a una computadora mediante un cable de red. La velocidad de conexión es de hasta 100 Mbps.

Para establecer la comunicación se debe utilizar el **Centro de redes y recursos compartidos** de Windows. Lo interesante y práctico, más allá del procedimiento particular, es saber que también podemos utilizar uno de los puertos USB para colocar e integrar un dispositivo Wireless.

Es tanta la diversidad de componentes que existen en el mercado, que se debe tener en cuenta la compatibilidad a la hora de adquirir este o cualquier periférico. Si ese no es el problema, para utilizar un adaptador Wi-Fi, los comandos que necesitaremos son los siguientes:

```
sudo nano /etc/network/interfaces
```

La interfaz de red se presenta de la siguiente forma:

```
auto lo

iface lo inet loopback

iface eth0 inet dhcp
```

```
allow-hotplug wlan0
auto wlan0

iface wlan0 inet dhcp
    wpa-ssid "Red_taller"
    wpa-psk "mo2vc"
```

Luego de efectuar las modificaciones en el nombre de red y la clave, reiniciamos la Raspberry Pi:

```
sudo shutdown -r now
```

Podemos comprobar el funcionamiento con el comando:

```
ifconfig
```

Realizamos un ping a una Web y podremos comprobar qué sucede:

```
ping cursosdeelectronica.com.ar
```



Tecnologías VNC y SSH

La tecnología **VNC** permite controlar una computadora desde otra, teniendo a ambas conectadas en red. Es lo que se conoce como **Escritorio remoto**. Al usar esta tecnología, obtenemos un acceso al escritorio de la computadora que deseamos controlar.

Por otro lado, la tecnología **SSH** (*Secure Shell*) se refiere a un protocolo y a un software que nos permitirá acceder a otra computadora, es decir, a la maquina definida como remota.

La ventaja de aplicarlas en la placa Raspberry Pi es que podemos disponer de los puertos USB para conectar periféricos diversos, como,

por ejemplo, el adaptador Wi-Fi que ya mencionamos. De esta forma, sin conectar un mouse y un teclado, podemos operarla desde otra computadora y acceder al control desde otro escritorio.

Tengamos en cuenta que, si dejamos disponibles los puertos, podemos crear una aplicación más interesante y no necesariamente dependemos o necesitamos operar a la placa Raspberry Pi como una computadora convencional, sino como un centro de procesamiento de información.

Debemos saber que estas tecnologías pueden ser aplicadas desde cualquier sistema operativo (GNU/Linux o Windows); solo bastará con conocer la IP para acceder a la placa Raspberry Pi.

En primera instancia, se debe conectar el teclado y mouse a la placa para configurar el sistema de acceso remoto. Luego de este proceso, ya pueden ser desconectados. El comando que usaremos es el siguiente:

```
curl icanhazip.com
```

Para iniciar la aplicación SSH, escribiremos el código que sigue:

```
ssh nombre_usuario@IP
```

Se debe sustituir la dirección IP, el nombre de usuario y contraseña. Cabe aclarar que, en forma predeterminada, ya están asignados de la siguiente manera: pi como usuario y RaspberryPi como contraseña.

En internet se encuentran disponibles varias aplicaciones gratuitas para utilizar estos protocolos, por ejemplo el **TightVNC** o el **VNC Viewer** para Google Chrome. Para instalar TightVNC, usamos el siguiente comando:

```
sudo apt-get install tightvncserver
```

Cuando el proceso de instalación finalice, debemos configurar las características gráficas para la placa.

```
tightvncserver -geometry 800x600-depth 16 -pixelformat rgb565
```

De esta forma, quedará configurada la conexión remota.

Luego, necesitaremos una aplicación como **VNC Viewer** en la computadora desde la cual ejerceremos el control.

Si se desea aplicar una tecnología SSH, se puede iniciar con los siguientes comandos:

```
sudo apt-get install ssh
```

Una vez resuelto el paso de la instalación, se inicia con:

```
sudo /etc/init.d/ssh start
```

Para utilizar este servicio, es necesario instalar un programa que nos permita crear el cliente en Windows. Uno muy recomendado en la Web es **Putty**, que en su versión de instalador posee varias aplicaciones para realizar la conexión con la placa Raspberry Pi y efectuar la configuración relacionada con la seguridad en la navegación.



RESUMEN



En este capítulo, recorrimos todas las características del hardware y los secretos del software necesarios para mejorar el rendimiento de Raspberry Pi. Además, conocimos su interacción con Arduino y aprendimos a controlar dispositivos periféricos de entrada y salida, a conectarla a una computadora mediante un cable de red e incluso a controlarla desde otra computadora, estando ambas conectadas en red.

Actividades

TEST DE AUTOEVALUACIÓN

- 1 ¿Qué podemos hacer con una placa **Raspberry Pi**, a diferencia de **Arduino**?
- 2 ¿Cuántos **modelos de placas Raspberry Pi** existen en el mercado y qué características tienen?
- 3 ¿Qué relación existe entre el rendimiento de esta placa y el **consumo eléctrico**?
- 4 ¿Qué elementos es necesario combinar para mejorar el **rendimiento** de la Raspberry Pi?
- 5 ¿Qué **corriente mínima** debe entregar la fuente de alimentación?
- 6 ¿Qué debemos hacer para instalarle un **sistema operativo**?
- 7 ¿Para qué sirve y qué características tiene la placa **Raspberry Pi Expansion Board**?
- 8 ¿Cuál es el protocolo que se debe utilizar para que Raspberry Pi y Arduino interactúen?
- 9 ¿Qué **lenguaje de programación** es el que se utiliza para escribir el código necesario que comunica la placa Arduino con la Raspberry Pi?
- 10 ¿Para qué sirve la tecnología **VNC**?



PROFESOR EN LÍNEA



Si tiene alguna consulta técnica relacionada con el contenido, puede contactarse con nuestros expertos: profesor@redusers.com

Conectividad por cable

La capacidad de poder interconectar diferentes dispositivos es, hoy en día, más que una necesidad, un modelo para seguir. Esto se ha dado a partir de la multiplicación de redes informáticas y elementos de uso cotidiano que son utilizados en conjunto o que mejoran sus prestaciones al ser interconectados. En este capítulo, estudiaremos el tipo de conectividad no inalámbrica.

▼ Conectividad y protocolos	▼ USB	201
TCP/IP	Funcionamiento	203
	Formato	205
	Diagrama de capas	206
	Puerto USB	208
	Conectores	209
	Aplicaciones	210
▼ Protocolos TCP/IP.....	▼ Software y utilidades.....	212
Capas del modelo OSI.....	▼ Resumen.....	217
	▼ Actividades.....	218
▼ Modulación y estándares.....		
184		
▼ Bus I ² C y SMBus.....		
189		
▼ Estándar SPI.....		
192		
▼ Comunicación asíncrona		
y RS-232.....		
197		



Conectividad y protocolos TCP/IP

Cuando surge el momento de realizar algún tipo de comunicación informática, normalmente a través de redes, existe un concepto que, en general, el usuario desconoce: son los protocolos y las normas que

UN PROTOCOLO ES
UN LENGUAJE Y UN
CONJUNTO DE REGLAS
QUE PERMITEN
COMUNICAR

debe seguir una máquina o interfaz para realizar una comunicación exitosa en ambos lados del sistema de comunicación. Esto incluye a cualquier sistema –no solo informático– que permita establecer una comunicación a través de él.

Dentro de este sistema se encuentran: la red de transmisión, que es el soporte físico, y, luego, se podría hablar de emisor y receptor.

Un ejemplo de comunicación podría ser una conversación entre personas, en la que el emisor y el receptor van variando de acuerdo a cuál es

la persona que habla y cual la que escucha. En este caso, la red de transmisión podría ser el aire, o bien una red telefónica.

En el caso de las personas, estas no necesitan establecer un protocolo que indique las normas de comunicación porque estas ya vienen incorporadas en aquellas. Por el contrario, en la informática, se debe establecer un protocolo que vaya variando sus normas según sus objetivos, ya que permite definir la manera en que se transmite la información o las tramas de bit de un punto a otro y define todos los aspectos que intervengan en dicha comunicación.

Si bien existen muchos protocolos que varían en su propósito y sofisticación, la mayoría comparte características, como la detección



HANDSHAKING

Este término proviene de la palabra en inglés que significa “apretón de manos”. En informática, define un proceso automatizado que establece los parámetros de un canal de comunicaciones para que comience, de manera dinámica, la transferencia normal a través del canal determinado. Este acto es independiente de la conexión física del canal.

de la conexión física subyacente al conectarse por cable o de manera inalámbrica. Otra característica común es la de iniciar y finalizar un mensaje; dentro de este parámetro se encuentra el tipo de formato o formateo de este, que puede variar dependiendo del tipo de seguridad que utilice un protocolo. Por ejemplo, un mensaje podría estar encriptado por diferentes métodos para que solo sea decodificado por el receptor definido de forma previa.

Por ende, los protocolos permiten un flujo de comunicación entre equipos que manejan distintos lenguajes, adaptándolos para establecer un **idioma** común entre ambos terminales. Por ejemplo, si varios terminales se conectan a una misma red pero estos difieren en protocolos, la comunicación no será posible. Por lo tanto, los protocolos son la solución y el requisito para establecerla.

El protocolo TCP/IP fue creado para establecer la comunicación en internet. Por eso, cualquier dispositivo capaz de conectarse a internet, posee este protocolo instalado.

Existen diversos modos de establecer una red entre terminales receptores de datos (ETD). Dentro de las modalidades más amplias, se encuentran la red por cable y la red inalámbrica.

En el caso de la red por cable, esta consiste en conectar físicamente varios terminales, por lo general, a través de cables UTP, variando las topologías de conexión y la cantidad de pares de acuerdo con el ancho de banda necesario. En redes de computadoras, se utilizan los cables UTP, con normas entre 5 y 6, fichas RJ45 y terminales plásticos con cuatro pares de conexiones. La cantidad de pares permite que, con un mismo cable, la comunicación sea bidireccional y que cada par se

LOS PROTOCOLOS
COMUNICAN
ENTRE EQUIPOS
CON DISTINTOS
LENGUAJES



HYPertext TRAnSFER PROTOCOL



Es un protocolo utilizado para la transacción de datos en la red informática mundial o World Wide Web (WWW). Define la sintaxis y la semántica que debe utilizarse para todo el software que participe dentro de la arquitectura de internet. HTTP está orientado a transacciones y sigue un esquema de petición-respuesta entre un cliente y un servidor.

pueda utilizar para enviar o recibir dependiendo del momento. Esto flexibiliza drásticamente la comunicación, en comparación con otras conexiones con menos pares disponibles.

Las conexiones inalámbricas entre computadoras se establecen dentro de la banda ISM. Esta reserva bandas internacionales para un uso no comercial. Las bandas pertenecen a radiofrecuencia electromagnética. Hoy en día, se han popularizado gracias a su uso en comunicaciones del tipo WLAN, dentro de las cuales se encuentra el famoso Wi-Fi, y redes del tipo WPAN-Redes Personales, como el Bluetooth.



Figura 1. Antena para enrutamiento inalámbrico de una red. Posee una interfaz con ficha BNC y es capaz de transferir frecuencias de radio.

Estas redes se establecen dentro de la frecuencia de 2.4 GHz y se utilizan en informática para designar nodos dentro de una red, sin la necesidad de una conexión física a través de cable.

Una de sus principales ventajas, en comparación con las conexiones físicas, es el ahorro de costos a través de la distancia. Esto significa que armar una red a 1 m de distancia podría ser más económico a través de cable, pero, cuando la distancia aumenta, comienza a ser más conveniente utilizar una red inalámbrica. También posee una desventaja notable, ya que para establecer conexiones inalámbricas se requiere un nivel de seguridad mucho

más sofisticado que el utilizado en conexiones por cable. Este sistema de seguridad ocupa un cierto ancho de banda que es fijo y, por lo tanto, disminuye el uso eficaz total de la red.

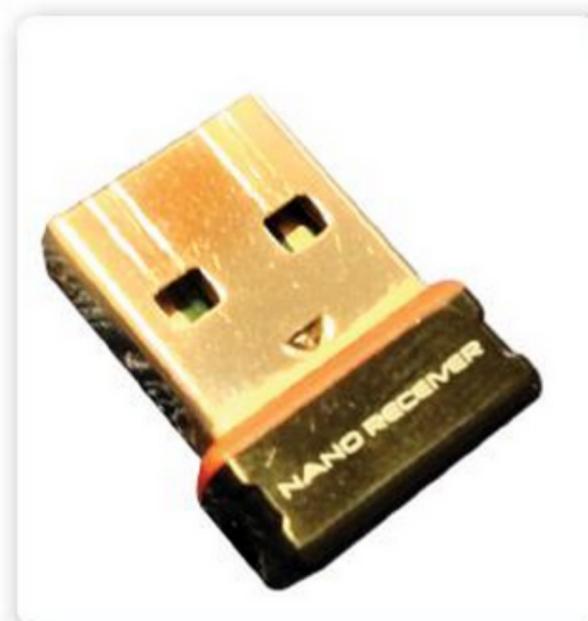


Figura 2. Cámara de seguridad que permite un monitoreo mediante una dirección web o una dirección local de IP. Posee conexión a la red mediante Wi-Fi.

A pesar de ser usadas de manera cotidiana mediante Bluetooth y Wi-Fi, las redes inalámbricas también son utilizadas a niveles mucho más grandes que el doméstico. Por ejemplo, las conexiones con satélites se realizan a través de microondas.

Las redes de teléfonos celulares trabajan últimamente bajo el estándar LTE (4G), que posee un ancho de banda adaptativo de entre 1.4 y 20 MHz. Además de tener muy baja latencia, posee velocidades pico de hasta 300 Mbps con antenas 4x4.

Figura 3. Receptor de teclados y mouse inalámbricos. Esta conexión implica protocolos de transferencia inalámbrica aislados de la red con interfaz Bluetooth.



En la actualidad, las redes inalámbricas son muy prometedoras y posibilitan la creación de nuevos protocolos e implementaciones para su desarrollo y evolución.

Protocolos TCP/IP

En general, los protocolos se clasifican en dos categorías según el nivel de control de datos: los orientados a conexión y los no orientados a conexión. Los primeros controlan la transmisión de datos durante la comunicación establecida entre los terminales. De esta manera, el equipo remitente es el responsable de validar los datos que envía como flujo de datos. TCP es uno de estos protocolos.



Figura 4. Cable UTP con ficha RJ-45. Se utiliza para redes locales y posee cuatro pares y una transferencia de 1 Gbps.



FAMILIA DE PROTOCOLOS DE INTERNET



Es un conjunto de protocolos de red sobre los cuales se basa internet. Por lo general, se hace referencia a esta familia de protocolos a través de los dos principales: TCP/IP. Ambos fueron los primeros en definirse dentro de la Web, aunque hoy en día en esta familia existen más de 100 protocolos diferentes; entre ellos, podemos destacar HTTP.

Por otro lado, existen los protocolos no orientados a conexión, mediante los cuales el equipo remitente envía datos sin notificar al receptor. Este recibe los datos como bloques o **datagramas**. UDP es un protocolo no orientado a conexión.

TCP es uno de los protocolos centrales de la familia de protocolos de internet. Fue presentado por primera vez en el año 1972 por el Departamento de Defensa de los Estados Unidos y ejecutado en su red ARPANET. Es tan común, que la familia, por lo general, suele llamarse TCP/IP.

Este protocolo provee una conexión confiable, ordenada y con control de errores, además de que permite enlazar computadoras que utilizan diferentes sistemas operativos, desde Mac OSX, Linux y Windows hasta terminales y computadoras centrales sobre redes privadas.

Los navegadores de internet utilizan TCP cuando se conectan a los servidores World Wide Web, y el protocolo es utilizado para entregar e-mails y transferir archivos de un punto a otro.

La familia de protocolos de internet puede describirse por analogía con el modelo de capas **OSI** (*Open System Interconnection*), que describe los distintos niveles de la pila de protocolos, aunque muchas veces en la realidad no se corresponda de manera muy exacta con este modelo.

En una pila de protocolos, cada nivel resuelve una serie de cuestiones o problemas relacionados con la transmisión de datos. Los niveles más altos son los más cercanos al usuario y tratan con datos más abstractos; en cambio, los niveles más bajos tienen la tarea de traducir los datos de modo que sean físicamente manipulables.

HAY PROTOCOLOS
QUE CONTROLAN LOS
DATOS DURANTE LA
COMUNICACIÓN ENTRE
LOS TERMINALES



APPLETALK

Es un conjunto de protocolos para la conexión de redes, desarrollado por Apple, que existe desde el año 1984. Se basa en el modelo OSI, pero, a diferencia de los sistemas LAN, este no fue construido en base a la interfaz Ethernet. En la actualidad, ya no se encuentra en uso debido a la masificación de la familia de protocolos de internet o TCP/IP.



Figura 5. Cable UTP con ficha RJ-11, que se utiliza para conexiones telefónicas.

Capas del modelo OSI

Los niveles del modelo OSI son los siguientes:

- Nivel de conexión física por cable UTP o coaxial, o bien, de conexión inalámbrica por microondas.
- Nivel de enlace de datos: encargado del direccionamiento físico, por interfaz Ethernet, Gigabit Ethernet, Token Ring, etcétera.
- Nivel de red: identifica el enrutamiento que existe entre una o más redes. Se utiliza direccionamiento con IPv4, IPv6, NetBEUI, AppleTalk, etcétera.
- Nivel de transporte: efectúa la transferencia de los datos desde su origen al destino, independizándolo de la red física. Se utilizan protocolos como TCP o UDP.
- Nivel de sesión: se mantiene y controla el enlace establecido. Se utiliza NetBIOS, RPC, SSL, aunque en muchos casos este nivel es prescindible.
- Nivel de presentación: encargado de la representación de la información, de manera que, aunque diferentes equipos tengan diferentes representaciones, los datos sean legibles.
- Nivel de aplicación: ofrece las posibilidades y aplicaciones de acceder a los servicios de red que definen los protocolos utilizados. Por ejemplo, el correo electrónico utiliza servicios POP y SMTP.

El protocolo TCP provee un servicio de comunicación intermedio entre el nivel de aplicación y el protocolo de internet IP. Esto ocurre cuando la aplicación desea enviar gran parte de la información a través de internet utilizando IP; en lugar de “partir” los datos en tamaños definidos por el protocolo IP, el software puede enviar una solicitud a TCP y dejar que este protocolo maneje los inconvenientes con el IP.



Figura 6. Entrada de red de una notebook a través de la interfaz Ethernet. Se utiliza para conexiones de redes privadas e internet.

Cada paquete es una secuencia de ocho bits y consiste de un header seguido por un body. Por un lado, el header describe la fuente, el destino y la información de control del paquete que está siendo enviado. Por otro lado, el body contiene la información IP que está siendo transmitida.

Su función principal es el uso de la comunicación bidireccional para transmitir datos mediante **paquetes**, que pueden ser transferidos a través de diferentes redes físicas determinadas por la norma de capas OSI. Con este propósito, IP define el formato de los paquetes y provee una dirección al sistema que tiene dos funciones: identificar al origen o host y proporcionar un servicio de localización.

EL PROTOCOLO IP
FUNCIONA MEDIANTE
EL INTERCAMBIO DE
PARTES O PAQUETES
DE INFORMACIÓN





Figura 7. Entrada de red de un teléfono fijo a través de **jack RJ-11**. Se utiliza para la conexión a una red telefónica.

La primera versión del protocolo IP fue la **4** o **IPv4**, y fue la primera en ser implementada de manera masiva. Trabaja bajo el modelo **best-effort**, que define una conexión sin la garantía de que los datos sean enviados o que el usuario obtenga una calidad de servicio determinada. La conexión planteada por el modelo establece que todos los usuarios obtienen el servicio con el “mayor esfuerzo posible”. Por lo tanto, el resultado obtenido puede variar en el tiempo de llegada o la tasa de transferencia, dependiendo del tráfico de datos que haya en ese momento.

La versión 4 del protocolo IP utiliza direcciones con un ancho de 32 bits o 4 bytes, lo que limita a un direccionamiento de hasta 2^{32} o 4.294.967.296 direcciones, muchas de las cuales están dedicadas a redes locales. Debido al gran crecimiento que tuvo internet y al gran desperdicio de direcciones de IP, la cantidad de direcciones está resultando insuficiente para el uso que se le da hoy en día.

La solución a este agotamiento viene de la mano de la última versión del protocolo IP. La versión 6 o **IPv6** provee identificación y ubicación para computadoras y terminales en red a través de internet. Esta versión fue diseñada para reemplazar a la versión 4, que posee la mayor parte del tráfico en internet.

Las direcciones son de un ancho de 128 bits, a diferencia de los 32 de IPv4, con una capacidad de direccionamiento de 2^{128} , más de

7.9 x 10²⁸ veces la cantidad de direcciones de la versión 4. Como los dos protocolos no fueron diseñados para operar juntos, esto dificulta la transición a IPv6.

Al trabajar ambas versiones (**IPv4** e **IPv6**) con el modelo best-effort, no garantizan un resultado consistente al otro punto de la conexión. Para realizar esto, se utiliza el protocolo TCP a otro nivel de conexión. Las aplicaciones que no requieran la confiabilidad de TCP podrían usar el protocolo UDP, que enfatiza la conexión de baja sobrecarga y baja latencia, en lugar de la validación del envío y el control de errores, que se enfoca en TCP.

El protocolo **UDP** (*User Datagram Protocol*) permite que aplicaciones de computadoras envíen mensajes o paquetes a otro host sin la necesidad de establecer una comunicación previa, ya que los propios paquetes tienen la información suficiente para direccionar dentro de su header. Al carecer de control de errores y de confirmación de recepción, los paquetes pueden adelantarse unos a otros. Su uso principal es para protocolos como DHCP o DNS, entre otros, en los que el intercambio de paquetes es mayor o no es rentable con respecto a la información transmitida (como la transmisión de audio y video en tiempo real, donde no es posible realizar retransmisiones).

Un subprotocolo que difiere de los propósitos de TCP y UDP es el ICMP. El protocolo de mensajes de control de internet se utiliza para enviar mensajes de error indicando que algún terminal o servicio no está disponible o no puede ser localizado. La principal diferencia es que no es usado de manera directa por las aplicaciones de usuario en una red.

Los mensajes de este protocolo se utilizan para determinar un diagnóstico o para aplicaciones de control que generen una respuesta de error en base a una dirección de IP. Estos errores se dirigen a la dirección de IP desde donde se envió el paquete originario.



DESPERDICIO DE DIRECCIONES EN IPV4



Este desperdicio y su consecuente agotamiento se deben a que se designaron demasiados bloques de direcciones para países y empresas. Otro factor es el crecimiento –mayor que lo previsto– de internet. Solo quedaron, para internet, una cantidad de aproximadamente 4,3 mil millones de direcciones, menos la cantidad asignada a empresas y países, que fue de alrededor de 16 millones.

El conjunto de protocolos en la familia de protocolos de internet está diseñado para enrutar y posee un grado elevado de fiabilidad, que va creciendo a medida que las prestaciones aumentan día a día. Si bien en los últimos años fueron cambiando las revisiones, adaptándose mejor a las velocidades que se espera manejar, cuando llegue el cambio más drástico, habrá disponible un modelo de protocolos con la flexibilidad para incorporar nuevas tecnologías.

➤ Modulación y estándares

La modulación de señales está presente en todos los procesos iniciales que implican a las telecomunicaciones. Durante el proceso para establecer una comunicación, por medio de una onda electromagnética, logramos propagarla a la velocidad de la luz, de manera independiente del medio y sin atenuación, lo que trae consigo la necesidad de utilizar un transductor que transforme las ondas mecánicas (como el audio, en caso de transmitir señales originadas en un estudio de radio, por ejemplo).



Figura 8. Antigua radio Panasonic. Solo poseía demodulador de AM. Fue una de las primeras generaciones de radios japonesas con transistores.

La señal por transmitir se denomina **banda base**, que podría ser originada por la voz humana, si tomamos como ejemplo el mismo caso del estudio de radio. Dicha señal no es periódica, lo que implica que por la antena transmisora pasarían los componentes espectrales periódicos de dicha señal. Esto se debe a que para que una antena funcione como transmisora se deben cumplir ciertos requisitos; uno de ellos es que la onda que se va a transmitir sea cosenoidal y periódica.

El método de modulación sirve para que se cumpla esta condición, además de hacerla apta para la transmisión a largas distancias, con la posibilidad de ser interpretada por uno o varios receptores. Consiste en superponer la señal, que siempre es de baja frecuencia, donde tengamos la información o **banda base** –ya sea de audio o datos digitales–, a otra señal de más alta frecuencia, denominada **portadora**, que es la señal cosenoidal periódica capaz de ajustarse a los requisitos de la antena y de ser transmitida a grandes distancias.

El proceso de interpretar la señal final que es transmitida al espacio se llama **demodulación** y consiste en eliminar la frecuencia portadora para luego obtener la señal que contiene la información y amplificarla, en caso de ser audio, o desviarla al lugar correspondiente, en caso de contener datos.

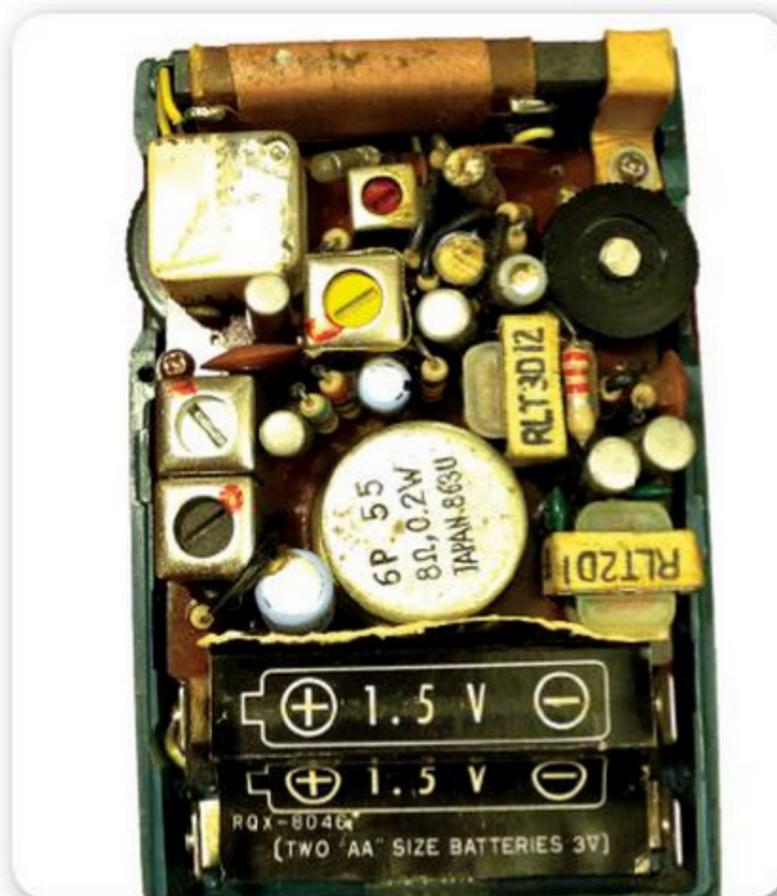
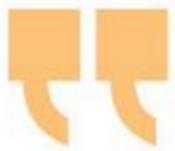


Figura 9. El sistema interno de la radio posee un ajuste fino de sintonización para no perder datos a la hora de la modulación.

LA ANTENA FUNCIONA
COMO TRANSMISORA
SI LA ONDA ES
COSENOIDAL Y
PERIÓDICA



Uno de los dispositivos más populares encargados de modular y demodular una señal es el **módem**. Este dispositivo viene utilizándose durante los últimos cincuenta años, por lo menos, y tiene el objetivo de producir una señal que pueda ser transmitida de manera fácil y decodificada para reproducir los datos codificados anteriormente. Los módems pueden ser usados con cualquier medio para transmitir señales analógicas, ya sea la transmisión por LED o radio.

El ejemplo más familiar de su uso es en una computadora con conexión a internet, donde los datos transferidos a través del proveedor del servicio son demodulados y enviados a la PC en cuestión. Otro ejemplo más fácil de visualizar, para los que no se encuentran tan familiarizados con las computadoras, es el teléfono, donde el rango de frecuencias de la voz humana es modulado y enviado por las líneas de teléfono para después ser demodulado por otro módem, del lado del receptor, que recupera los datos.

Por lo general, los módems son clasificados por el volumen de datos que pueden transmitir por segundo: bits por segundo (bit/s o bps) o bytes por segundo (B/s).



Figura 10. Los routers poseen un sistema complejo de modulación y demodulación, basado en estándares para redes.

También pueden ser clasificados por la cantidad de veces por segundo que el sistema es capaz de cambiar su estado físico. Esta unidad de tasación recibe el nombre de **baudio** y se utiliza cuando se necesita conocer el volumen de transferencia de manera independiente del tipo de modulación, ya que en virtud de eso, puede variar la tasa en bps.

Para que un dispositivo se pueda comunicar con otro, debe existir un conjunto de reglas y normas que permita la comunicación entre ellos. Ese conjunto de reglas debe ser compartido por ambos equipos que deseen establecer una comunicación y es llamado **protocolo**.



Figura 11. Módem de la marca Motorola con entrada de cable coaxial. No posee capacidades de enrutamiento, a pesar de tener una suerte de división de la red a través de Wi-Fi.



MÓDEM ROUTER



Existen equipos de módem con un circuito enrutador de redes para PC. Esto es posible por la reducción de tamaño que sufrieron los circuitos de módem, que permite la integración con los circuitos de routers. Es una gran ventaja debido a que se ahorra un dispositivo dentro de la red hogareña. La desventaja es que la red inalámbrica suele tener poco alcance por su antena de baja ganancia.

Los sistemas de comunicación utilizan diferentes formatos para intercambiar mensajes. Cada mensaje tiene un único significado y un objetivo de provocar una respuesta particular en el receptor.



Figura 12. Módem de la marca Huawei con entrada de datos ADSL y capacidad de conectar un teléfono IP. Permite enrutar tanto por cable UTP como por red inalámbrica.

UN PROTOCOLO ES
A LA COMUNICACIÓN
LO QUE UN LENGUAJE
DE PROGRAMACIÓN
ES A LOS CÓMPUTOS



Un protocolo debe definir la sintaxis, la semántica y la sincronización de la comunicación. El comportamiento especificado es, por lo general, independiente de cómo debe implementarse. Por lo tanto, un protocolo puede ser implementado tanto a nivel **hardware** como **software**, o bien, en ambos.

Todas las partes involucradas dentro de un sistema de comunicación deben acordar los protocolos que utilizarán. Para llegar a este acuerdo, un protocolo puede ser convertido en un estándar. Un lenguaje de programación describe lo mismo para los cómputos, así que existe una analogía estrecha entre protocolos y lenguajes de programación: Los protocolos son a las comunicaciones como los lenguajes de programación son a los cómputos.



Bus I²C y SMBus

El **bus I²C** o **IIC** está dedicado a las comunicaciones en serie. Fue inventado por **Phillips** y, como su nombre lo indica (*Inter-Integrated Circuit*), fue diseñado para la interconexión de periféricos de baja velocidad que se conectan a un motherboard, teléfono celular u otro dispositivo electrónico.



Figura 13. Disco IDE de marca Seagate configurado como master gracias a la posición del jumper blanco a la derecha de la interfaz IDE.

Su primera versión, 1.0, fue lanzada en 1992, mientras que su segunda versión, 2.1, fue lanzada en el año 2000. A mediados de los años 90, muchas compañías pusieron a la venta productos con el sistema I²C. Dentro de estas empresas se encontraban **Motorola** y **Siemens AG**, cuyo nombre actual es **Intel Mobile Communications**. Desde octubre de 2006, no se requieren licencias para implementar este protocolo, aunque sigue siendo necesaria una pequeña cuota a la hora de utilizar las direcciones **slave** del protocolo I²C, que posee la empresa **NXP**, una de las líderes mundiales en la venta de semiconductores.

Dentro de la industria de los semiconductores, este protocolo es muy utilizado, principalmente cuando hablamos de la comunicación entre microcontroladores y posibles periféricos (siempre en el caso de periféricos de sistemas integrados u **onboard**).

Muchas veces, se suele confundir al I²C con el término **Two Wire**, ya que este bus utiliza dos líneas para transmitir información. La primera, llamada SDA, transmite los datos en serie, y la segunda, denominada SCL, transmite la señal de **Clock**. Además, es necesaria una tercera línea, llamada GND, para la referencia de masa o tierra. Como fue mencionado anteriormente, los periféricos que se suelen conectar a través de este bus son dispositivos **onboard**, lo que significa que están dentro de la misma placa; por lo tanto, esta tercera línea de masa se suele omitir.

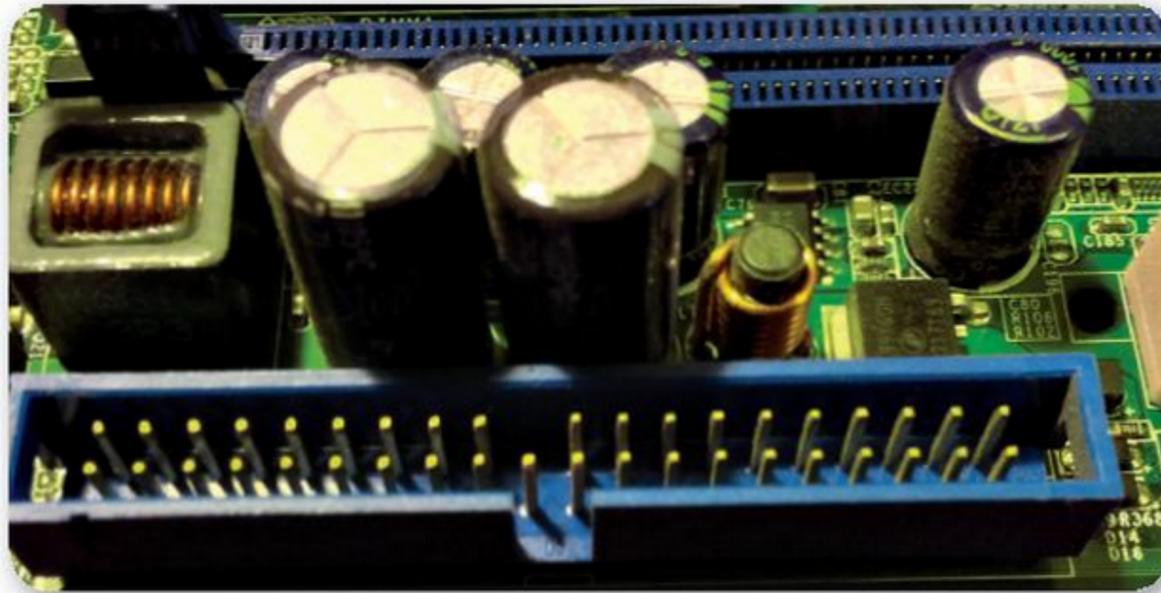
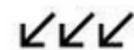


Figura 14. Un motherboard posee pines de selección master y slave, que serán descartados o tenidos en cuenta de acuerdo con la configuración que se haya hecho con jumpers.

En teoría, el I²C permite la conexión de más de 1000 dispositivos en paralelo a sus dos líneas de datos y **Clock**. Las transferencias de datos nunca se realizan de a más de dos dispositivos a la vez, y la relación que se establece para la comunicación es la ya conocida por cualquiera que tenga mínimos conocimientos sobre los discos IDE: **master-slave** o **maestro-esclavo**, en los que, generalmente,



SELECCIÓN DE MASTER EN DISCOS IDE



Lo más próximo al concepto de master y slave pueden ser los discos IDE. Estos discos tienen un sistema de selección que indica qué disco es master y cuál es slave, en caso de poseer varias unidades. El sistema consiste en pines que se encuentran a un costado de la interfaz IDE, y se deben puentear con jumpers con la configuración que indique el fabricante sobre la selección de jerarquía del disco.

los dispositivos **master** son microcontroladores y los **slave** son memorias, conversores AD, pantallas LCD o algún tipo de sensor.

La ventaja de este tipo de transferencias es lograr una transmisión sin pérdida de datos. Sus velocidades van desde los 10 kbit/s en modo baja velocidad, o bien 100 kbit/s en modo estándar. Nuevas revisiones han logrado transmisiones de host más veloces, incorporando el modo rápido con velocidades de 400 kbit/s, y un segundo modo rápido, que alcanza velocidades de 1 Mbit/s. Por último, encontramos el modo de alta velocidad donde se obtienen velocidades de hasta 3.4 Mbit/s.

Por lo general, el master es un microcontrolador o microprocesador, ya que es el encargado de generar la señal de Clock y enviarla por el bit SCL; además, inicializa la comunicación enviando un bit de Start y, luego, procesa la información que fue transferida. Pero la definición de un master dentro de una comunicación implica al dispositivo responsable de iniciar la comunicación, llevando consigo la transmisión del ritmo de transferencia o pulsos de Clock. En cambio, el nodo slave es pasivo, ya que no emite ningún tipo de señal sin la orden apropiada del master; es el encargado de recibir los pulsos de Clock y, a raíz de esto, establecer una comunicación con el master. La comunicación finaliza cuando el master envía la señal de Stop y se dejan de emitir los pulsos de Clock.

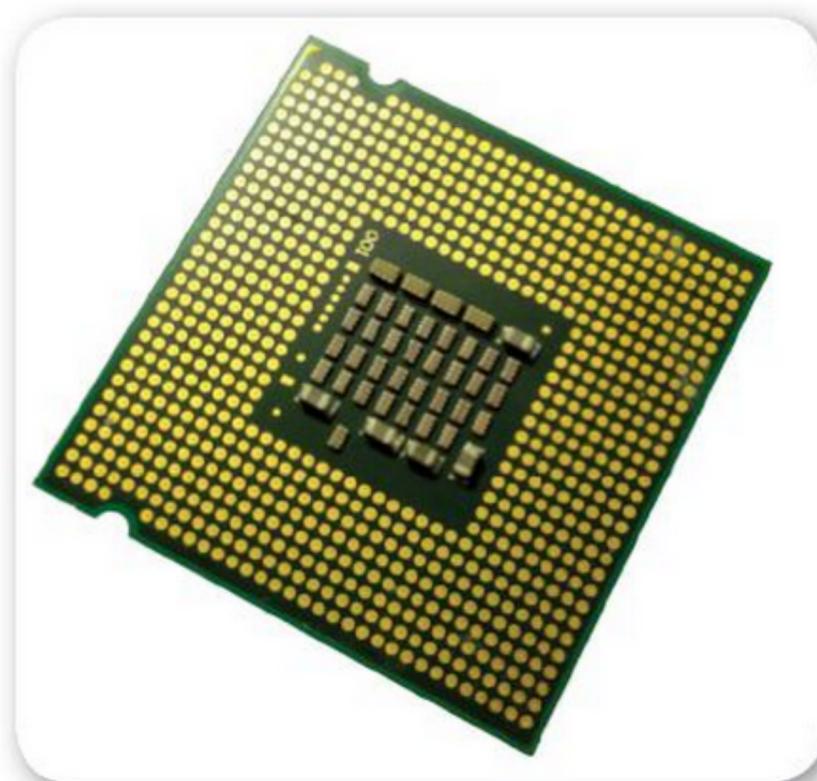


Figura 15. La CPU es la encargada, a nivel comunicación de la PC, de gestionar la entrada de datos y su procesamiento. Es el ejemplo más complejo de establecimiento de comunicaciones.

MASTER-SLAVE
IDENTIFICA LOS
NODOS DENTRO DE
LA COMUNICACIÓN
DE DATOS



Existen buses **multi-master**, lo que significa que puede encontrarse más de un nodo master dentro de un mismo sistema de comunicación I^{2C}. Esto implica un sistema mucho más flexible, ya que no depende de un solo dispositivo master, y se pueden utilizar los periféricos slave de una manera más dinámica, debido a que las transferencias I^{2C} permiten una sola comunicación a la vez, y tener uno o varios nodos master implicaría varias conexiones simultáneas y un mejor

aprovechamiento del tiempo. El **SMBus** es una rama del I^{2C}, definido por Intel en el año 1995, que define protocolos de comunicación más estrictos que su protocolo de origen. Uno de los propósitos de SMBus es promover la robustez e interoperabilidad, lo que se traduce en conexiones con comunicaciones fieles sin caídas ni pérdidas y la capacidad de habilitar dos o más sistemas para intercambiar información.

Según las últimas revisiones de I^{2C}, este incorpora muchas reglas y políticas del protocolo SMBus, fortaleciendo cada vez más la prevalencia de este último.

Estándar SPI

El **SPI** (*Serial Peripheral Interface*) es un bus de transferencia de datos de forma sincrónica y serial, como su nombre lo indica. La transferencia sincrónica implica que el master define un tiempo en el cual se supone que el slave va a responder, aunque, en realidad, esta suposición se basa en cálculos sobre la cantidad de ciclos de máquina que consume una tarea específica en el master y en el slave. Prestablecer ese período le ahorra tiempo de espera al master; en teoría, estos mínimos ahorros de tiempo se traducen en un mejor tiempo total de transferencia. En la lectura sincrónica, el master pone en el bus la dirección deseada y supone que, del otro lado, el slave, a su vez, pone el dato en el bus en un tiempo preestablecido T, momento en el que el master lo toma. Mientras que, en la escritura, el master pone la dirección y el dato en el bus suponiendo que el slave toma esta información antes del tiempo T.

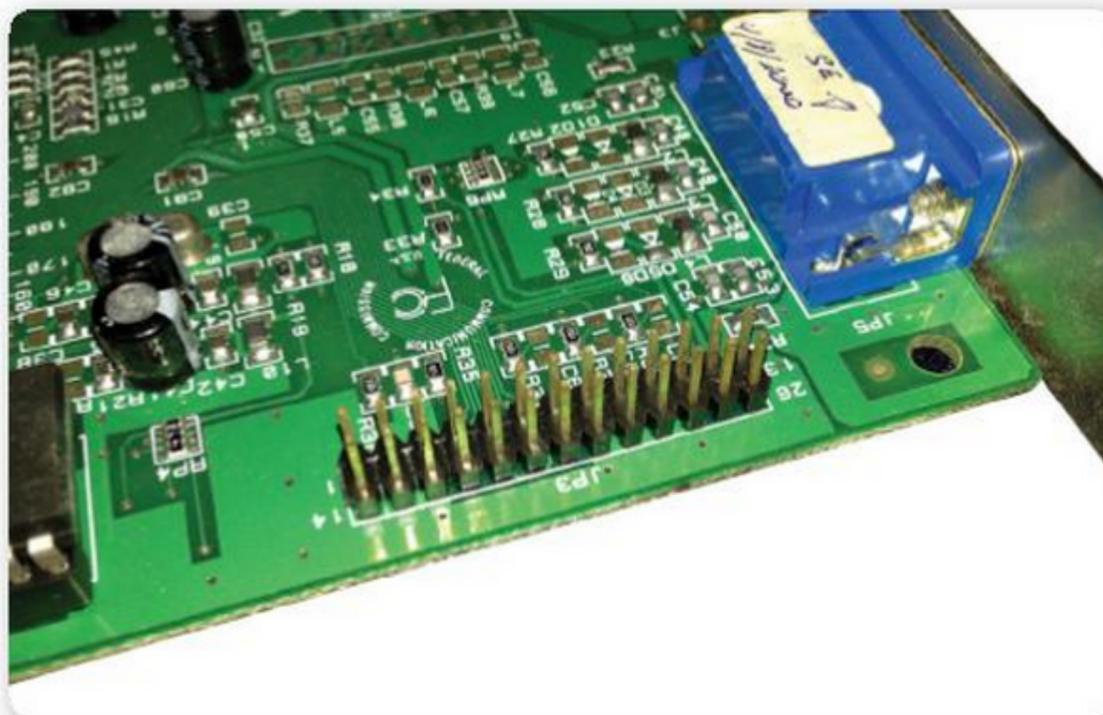


Figura 16. Muchas veces se puede tener pines dispuestos para una conexión personalizada, o bien, conexiones a dispositivos dispuestos por el fabricante.

El estándar SPI fue creado por Motorola y es el primero que opera en modo bilateral. El modo bilateral implica que ahora se poseen líneas o bits independientes para la entrada y la salida de datos. No debemos confundirlo con la transferencia en paralelo, ya que cada línea entrante o saliente maneja transferencias del tipo serie.

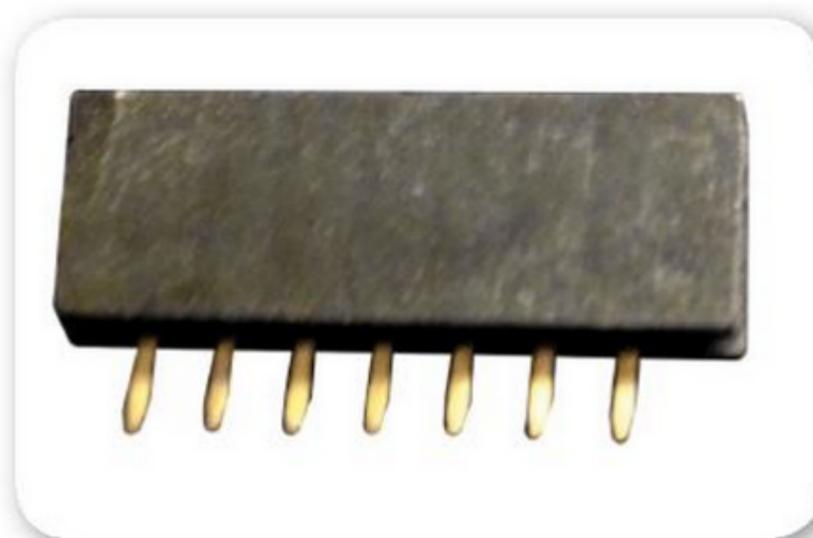


Figura 17. Para realizar las conexiones con los pines que se puedan llegar a disponer en un circuito impreso, se deberá armar un cable con el conector apropiado.

Con la nueva implementación de transferencia bilateral o full duplex, múltiples dispositivos slave son soportados contando con una línea o bit llamado slave select o selector de esclavo.

En muchas ocasiones, se suele encontrar al bus SPI como bus de **cuatro líneas**, en contraste con los buses de una, dos o tres líneas. Esto se debe a su interfaz de 4 bit, donde el primer bit, llamado **SLCK**, posee la señal de Clock que con cada pulso se lee o se envía un bit. También es posible encontrarlo con la nomenclatura **CLK** o **TAKT**, que proviene del idioma alemán.

El segundo bit es el **MOSI** (*Master Output Slave Input*), que implica la salida de datos del master y la entrada de datos del slave. También es posible encontrarlo con la nomenclatura **SIMO**.

La contraparte del bit anterior es el llamado **MISO** (*Master Input Slave Output*), que implica la entrada de datos del master y la salida de datos del slave. También es posible encontrarlo con la nomenclatura **SOMI**.

Por último, se encuentra el bit **SS** (*Slave Select*), que es el encargado de seleccionar el dispositivo slave con el cual se va a establecer la comunicación. El master indica con un 0 lógico, en este bit, que desea escribir. De modo contrario, el bit SS tendrá un 1 lógico cada vez que el master desee leer.

Para comenzar la comunicación, el bus master configura el Clock usando una frecuencia menor o igual a la máxima frecuencia a la que el slave, con el que desea comunicarse, puede trabajar. Estas frecuencias, por lo normal, se encuentran entre los 10 kHz y los 100 Mhz.

Luego, el master transmite un 0 lógico por el SS para seleccionar el dispositivo con el que desea comunicarse. En caso de realizarse una conversión analógico-digital, este deberá esperar por lo menos un determinado tiempo (establecido por el fabricante) antes de empezar a emitir pulsos de Clock.

Las transmisiones, en general, involucran a dos registros de un tamaño de 8 bits. La información usualmente se transmite con el bit más significativo en primer registro, mientras que se conmuta hacia el



DISPOSITIVOS CON SALIDAS TRI-STATE



Muchos slaves poseen salidas de tipo tri-state, por lo tanto, la señal proveniente del bit MISO es interpretada como de alta impedancia (es decir, desconectado). Esto ocurre cuando el dispositivo no fue seleccionado por el bit SS de selección. Por lo tanto, los dispositivos sin salidas tri-state no podrán compartir el bus SPI con otros mecanismos.

otro registro con los bits menos significativos. Luego de que los registros fueron transmitidos, el master y el slave procesan los datos y, si aún hay más para intercambiar, los registros se cargan con nuevos datos y se repite el proceso anterior.



Figura 18. La conexión entre ambos dispositivos se suele hacer con un cable plano, en el que es más fácil identificar las líneas con números o letras.

Todas estas comunicaciones pueden involucrar cualquier cantidad de ciclos de Clock. Cuando no haya más datos para transmitir, el master detendrá su pulso de Clock; normalmente, luego deseleccionará al slave.

Generalmente, las transmisiones constan de registros de 8 bits, donde el master puede inicializar múltiples transmisiones si así lo requiere. Aunque pueden ser encontrados registros con más bits, como los controladores de pantallas táctiles, los códecs de audio que utilizan 16 bit, o bien, varios conversores analógico-digital o digital-analógico que utilizan registros de 12 bit.



ACTIVACIÓN DE SLAVES



Cada slave en el bus SPI que no sea activado usando la línea correspondiente SS, deberá ignorar la señal de Clock –a pesar de no ser seleccionados, todos los slaves recibirán señal de Clock–, además de ignorar las señales MOSI y MISO, ya que producirá malos funcionamientos, probablemente por exceso de consumo de corriente, ya que el master puede seleccionar y soportar un slave a la vez.



Figura 19. Las tarjetas SD poseen el protocolo de transferencia **serie SPI**, aunque están reducidas sus posibilidades debido a sus prestaciones.

Por último, este estándar trae muchos avances en comparación con I²C y SMBus, como el gran incremento de la velocidad debido a su comunicación bilateral o full duplex. Vale destacar que este protocolo es mucho más flexible, ya que se puede tener un control absoluto sobre los bits que fueron enviados y recibidos. El tamaño de los registros no está limitado; si bien se usa normalmente un ancho de 8 bits, antes nombramos aplicaciones que van hasta los 16 bits. Además, este estándar consume menos corriente, ya que estamos hablando de circuitos más sencillos y de menor cantidad de componentes. Otra de las ventajas de este bus es que el Clock es compartido con todos los slaves; por lo tanto, no hace falta que estos posean su propio Clock. Tampoco se requiere la aplicación de un circuito transceptor, ya que todo dispositivo conectado puede ser configurado para enviar y/o recibir.

A pesar de todas las ventajas de este estándar, también cabe mencionar sus puntos bajos, en comparación con los buses vistos con anterioridad. Por ejemplo: requiere más pines que el bus I²C, inclusive en la configuración más simple, de tres líneas. Carece de un control de asentimiento, lo que implica que un master podría estar enviando información sin conocer que del otro lado está conectado un slave.

Por otro lado, no se permiten conexiones con varios master, como es el caso del SMBus. Por último, es importante destacar que solo funciona a distancias cortas, por eso se lo utiliza casi únicamente onboard. Esta será la principal diferencia en comparación con el próximo bus.



Comunicación asíncrona y RS-232

El concepto de sincronismo se puede aplicar en los buses a distintos niveles. El enfoque de esta sección será el de las transferencias elementales entre dos dispositivos de forma similar a como se observó con los estándares I²C y SPI.



Figura 20. Puerto Serie DB-9 de un panel trasero de una computadora de escritorio.

La comunicación asincrónica es aquella que se establece entre dos o más entes de manera independiente del tiempo. Pero, en lo que refiere a la electrónica, los entes son representados por dispositivos master-slave, y el tiempo está definido por la señal u orden que envía el master para comenzar una comunicación. En otras palabras: el remitente puede transmitir datos en cualquier momento y el destinatario deberá recibir la información cuando llegue, a diferencia de las transmisiones sincrónicas que requieren de un flujo muy preciso de bits, donde el principio del mensaje contiene un bit de activación o llamada a sincronismo.

Las transmisiones asincrónicas se logran gracias a que se sincronizan los dispositivos que se van a comunicar mediante un mecanismo de

RS-232 TOMA TODAS
LAS VENTAJAS DE
LAS TRANSFERENCIAS
SERIE SINCRÓNICAS
Y ASINCRÓNICAS



Clock compartido u otra técnica. Toda esta preparación es realizada con la primera puesta en marcha. Estas transmisiones se utilizan cuando los datos se envían de manera periódica sin un intervalo especificado; por ejemplo, un usuario que utiliza un teclado y se espera que presione las teclas, pero no hay un tiempo de intervalo entre cada tecleo. El sistema receptor, durante ese período, “sabe” esperar hasta el próximo dato, que puede llevar bastante tiempo. Por el contrario,

las transmisiones sincrónicas se suelen utilizar entre terminales que transfieren grandes cantidades de información de modo regular.



Figura 21. Osciloscopio con salida RS-232 a través del puerto DB-9.

Las dos acciones fundamentales de cualquier transferencia son la lectura y la escritura. En el caso de la lectura asincrónica, el master pone en el bus la dirección deseada, espera que el slave conteste y confirma que ha puesto el dato en el bus, o bien, cuando la espera sobrepasa cierto tiempo, genera una señal de error. En cambio, en la escritura asincrónica, el master pone en el bus el dato y la dirección y, luego, espera que el slave confirme que ha tomado esta información. Si la espera sobrepasa un cierto intervalo, el master genera una señal de error.

Esta norma se estableció para la conexión de equipos informáticos a módems. Aunque, en realidad, se estuvo (y en algunas ocasiones todavía se está) utilizando para interconectar equipos informáticos

directamente entre sí, sin el empleo de módems. Esto hace que no se pueda aplicar de manera directa la norma y que la interpretación dada a sus distintas señales difiera de un fabricante a otro. Sin embargo, está muy difundida, en especial, en el ámbito de las máquinas pequeñas, como los dispositivos de cobranza de un supermercado.

El bus **RS-232** está diseñado para conectar un equipo informático (DTE) a un módem (DCE). Las velocidades de transmisión previstas fueron inicialmente de 110 a 19200 baudios.

El estado lógico de las señales que se transmiten por este estándar se determina por niveles de tensión. Los valores son los siguientes:

- De +15 V a +3 V, la señal se considera un “0” lógico.
- De -15 V a -3 V, la señal se considera un “1” lógico.

Esta norma considera veinticinco señales, de las que se suelen emplear menos de diez. Las más importantes son la DB-25, comúnmente utilizada para la instalación de una impresora; la DB-9, típica de los puertos COM serie de una PC; y la MJ8.



Figura 22. Cable para transferencias serie RS-232, a través de un puerto DB-25 con puntas hembra y macho.

Dado que la norma se aplica para interconectar equipos de manera directa, asignarles a estos el tipo **DTE** y **DCE** es completamente arbitrario, lo que puede producir problemas a la hora de conectar equipos de distintos fabricantes, puesto que ambos pueden ser DTE

o DCE. Esto obligará a intercambiar los hilos de transmisión 2 y 3, así como a adaptar las señales de control empleadas, ya que estas también varían según el fabricante.

El método de transmisión establece la transferencia elemental del bus serie. Existen dos procedimientos básicos para la conexión de terminales, denominados asincrónico y sincrónico. El asincrónico se usa para aplicaciones de baja velocidad y suele emplear el estándar RS-232-C. Además, los sistemas que utilizan este método de transmisión suelen codificar la información en ASCII, claro ejemplo de un teclado. Por el contrario, el sistema sincrónico permite velocidades que llegan hasta los 500 kbaudios y suele emplear el estándar RS-422 o transmisores especiales de cable coaxial.

Existen pastillas denominadas **USART** (*Universal Synchronous/Asynchronous Receiver/Transmitter*), que se encargan de realizar todas las operaciones necesarias en la transmisión RS-232.

Hoy en día, el bus RS-232 ha sido reemplazado en la PC por el USB, que es mucho más rápido, usa menores tensiones, pero no es tan sencillo de conectar y de poner en funcionamiento. Además, está limitado por norma a 5 metros de cable, lo que es un punto favorable para RS-232 cuando se requieren distancias más grandes. Ambos estándares tienen software que los respaldan en varios sistemas operativos, aunque el USB está diseñado para simplificarle la tarea a los controladores de dispositivos, al conectarse con el hardware. No obstante, no hay forma analógica directa de conectarse al terminal usado para que los usuarios se comuniquen de manera directa con los puertos serie.

El estándar USB es más complejo que el RS-232, porque incluye protocolos de transferencia de datos hacia dispositivos, y esto requiere más software de protocolo. En cambio, el RS-232 solo estandariza las tensiones de referencia y las funciones de cada pin.



VLSI

Esta sigla proviene de **Very Large Scale Integration** y refiere al proceso de creación de circuitos integrados que combina miles de transistores en un mismo encapsulado. Este método surgió en 1970 y cambió de manera radical la posibilidad de construir periféricos. Además, el avance de sus recursos de hardware hizo posible poner cada vez más transistores en una superficie menor.



Figura 23. Frente de expansión con puerto DB-9 para conectar a la placa de expansión de puertos serie.

Los puertos serie en la PC también son usados para controlar el hardware de manera directa, ya que la interfaz de control puede ser fácilmente manipulada por software. Esto no es posible con USB, debido a que requiere algún tipo de receptor que decodifique la información.

Si bien cada vez se utiliza menos este estándar, podemos seguir encontrándolo en aplicaciones o hardware externo de prestaciones muy básicas, que no requieren las velocidades que ofrecen otros estándares. También es una forma muy sencilla de introducirse en el mundo del diseño de periféricos, para aquellos que recién comienzan y desean aprender lo básico sobre las interfaces.

HOY, EL DISEÑO
DE RS-232 ESTÁ
SIMPLIFICADO POR
LOS CIRCUITOS VLSI
DISPONIBLES



USB

El **bus universal serie**, conocido como **USB** (*Universal Serial Bus*), es un estándar industrial desarrollado alrededor del año 1990, que no solo define un protocolo sino también cables y conectores. Todos estos elementos se utilizan en un bus serie, a través del cual se puede conectar, comunicar y transmitir alimentación entre computadoras y dispositivos externos.

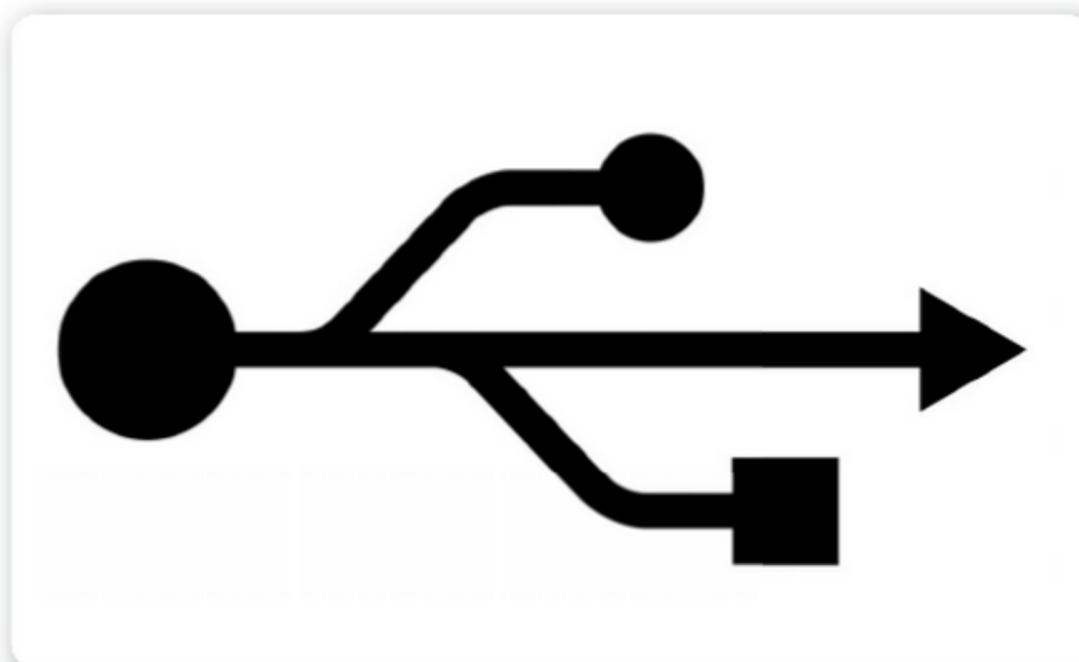


Figura 24. Este es el símbolo con el que se identifica a los puertos o conectores USB. Según la versión, su color puede cambiar o mantenerse y agregar el número de versión.

En la década del 90, grandes empresas, como Compaq, IBM, Microsoft, Intel, NEC y Telecom, entre otras, se unieron para diseñar una nueva interfaz, que hoy en día se conoce como USB. Se empezaron a realizar pruebas sobre su implementación y, en 1996, se presentó la interfaz al usuario.

USB ES UN ESTÁNDAR INDUSTRIAL QUE FUE DESARROLLADO ALREDEDOR DEL AÑO 1990



El USB fue revolucionario, porque corrigió y reemplazó a los puertos series y paralelos, aumentando su velocidad de transferencia a unos 12 Mbps. A su vez, unificó los tipos de cable que le permitían al usuario cumplir con todas sus necesidades de conexión, brindándole la posibilidad de manejar muchos periféricos al mismo tiempo. El máximo de conexiones permitidas por puerto es de 127 dispositivos. Estos puertos trabajan con una tecnología **plug & play**, que permite conectar o desconectar sin necesidad de recargar el sistema operativo.

Para mediados del año 2000, un nuevo grupo, integrado por Compaq, Hewlett-Packard, Lucent, Intel, Microsoft, Philips y NEC, presentó el estándar USB revisado. Esta versión se llamó 2.0, lo que indicaba un incremento en la velocidad (**high speed**). Su principal característica era el aumento en el rendimiento general del estándar. Aumentó la velocidad de 12 Mbps a 480 Mbps. Además, era compatible

con la versión anterior (1.0), porque el conector y el cableado no sufrieron ninguna modificación.

Con el paso del tiempo, las exigencias sobre el USB 2.0 fueron aumentando y se creó, así, el sucesor de dicho estándar. Para fines de 2008, se presentó la versión llamada 3.0 (*super speed USB*), que incluyó nuevas tecnologías, como **Optimized Power Efficiency** y la tecnología **Sync-N-Go**, que permite minimizar el tiempo de espera para el usuario. Posee una velocidad de 5 Gbps, y los puertos siguen siendo compatibles con sus antecesores.

EL USB FUE
REVOLUCIONARIO
PORQUE REEMPLAZÓ
A LOS PUERTOS
SERIES Y PARALELOS



COMPARACIÓN DE VERSIONES

▼ VERSIÓN DEL DISPOSITIVO	▼ CAPACIDAD (MEGABYTES Y GIGABYTES)	▼ VELOCIDAD (MEGABYTE Y GIGABYTE POR SEGUNDO)
USB 1.1	Primeras: 8 – 16 – 32 – 64 – 128 – 256 MB.	1.5 Mbps a 12 Mbps.
USB 2.0	16 – 32 – 64 – 128 – 256 GB.	Hasta 480 Mbps.
USB 3.0	Primeras: 16 – 32 GB.	Hasta 3.2 Gbps.

Tabla 1. Comparación de las distintas versiones del estándar con respecto a las diferentes capacidades de almacenamiento y velocidad.

Funcionamiento

Esta interfaz permite la transmisión de datos y la distribución de energía eléctrica. Mejoró de manera drástica las lentas velocidades que tenían otras interfaces, como la RS-232 y la paralela. Utiliza cuatro hilos para transmitir los datos y brindar 5 V. El bus en el que está basada esta interfaz es semejante a otros buses, como los que presenta una red local en forma de anillo. El controlador distribuye testigos en el bus, y el dispositivo conectado mediante el cable USB, cuya dirección coincide con el testigo que mandó el controlador, responde de dos maneras: acepta la conexión o envía información.

El bus emplea una topología particular en forma de estrellas apiladas que posibilita un funcionamiento simultáneo de 127 dispositivos.

A diferencia de las otras arquitecturas, el USB no es un bus de almacenamiento, sino que recibe y transmite datos, por lo que no produce ningún tipo de retraso en el envío de un paquete de información.

Las señales se transmiten a través de un cable de par trenzado con una impedancia característica de $90 \Omega \pm 15 \%$, cuyos hilos se denominan D+ y D-. Utiliza una conexión half-duplex, a excepción de la versión 3.0, en la que, mediante un segundo par de hilos, se realiza una comunicación full-duplex.

La transmisión **half-duplex** reduce el efecto del ruido electromagnético en los cables D+ y D-. Los valores de la señal varían de 0 a 0.3 V para niveles bajos; de 2.8 a 3.6 V para niveles altos, en las versiones 1.0 y 1.1; y en ± 400 mV en alta velocidad (2.0). En las versiones 1.0 y 1.1, los alambres de los cables no estaban conectados a masa, pero en la versión de alta velocidad se tiene una terminación de 45Ω a masa o un diferencial de 90Ω para acoplar la impedancia del cable. A su vez, este puerto solo admite la conexión de dispositivos cuyo consumo máximo sea 100 mA, por lo que permite, únicamente, dispositivos de bajo consumo.

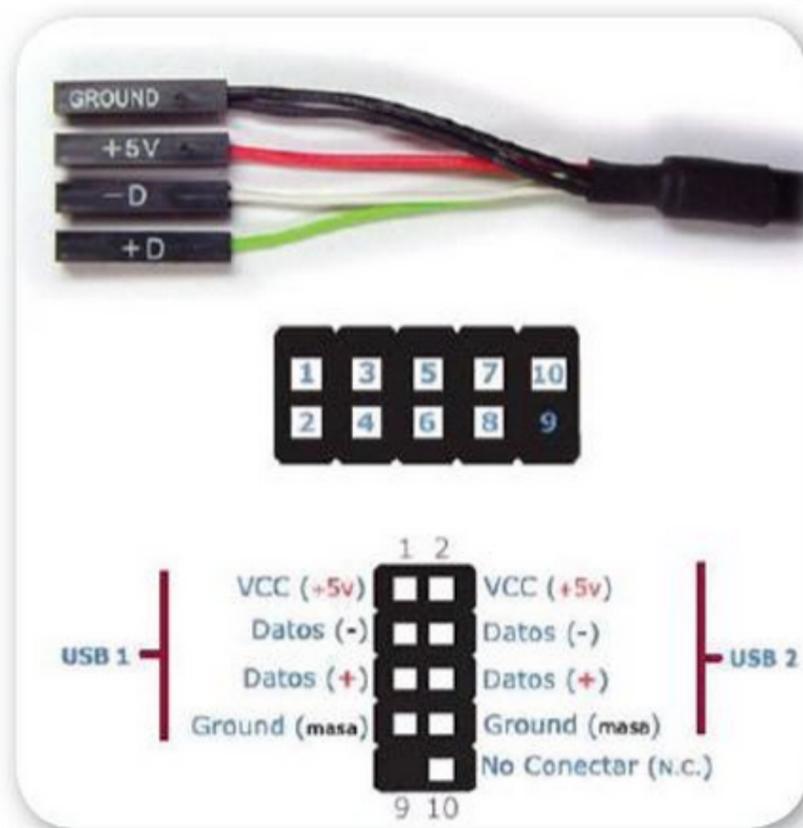


Figura 25. Identificación de color y descripción de cada conductor presente en el cable USB.

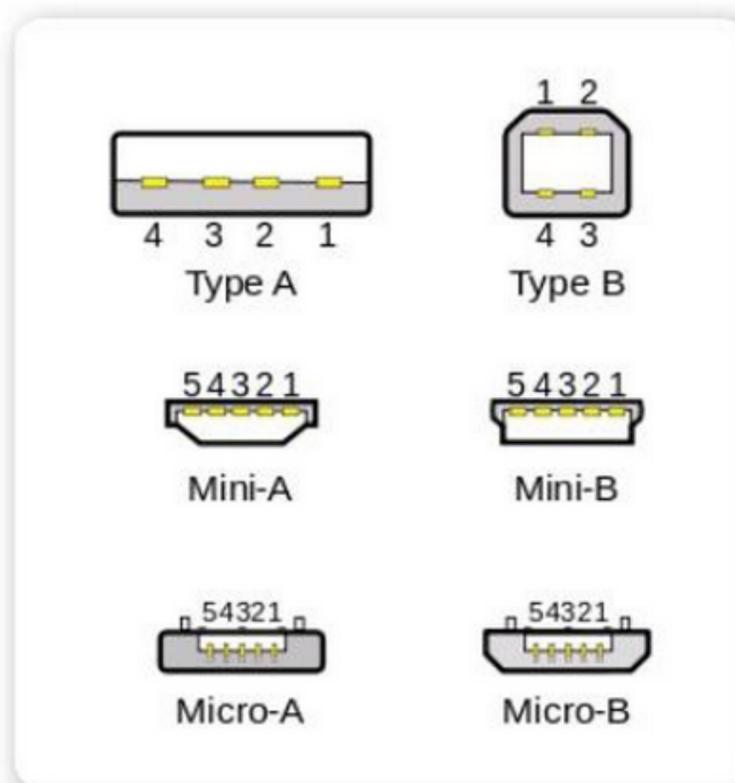


Figura 26. Distribución de cada conductor en los distintos conectores. Cada conector tiene una distribución específica en la numeración de los contactos.

Formato

Los dispositivos USB se pueden clasificar según su velocidad de transferencia de paquetes de datos. Existen cuatro tipos, de acuerdo a esta clasificación:

- Los dispositivos de **baja velocidad**, que corresponden al estándar 1.0, cuya tasa de transferencia es de hasta 1,5 Mbps. Mayormente, son utilizados por dispositivos de interfaz humana, como teclados, mouses, entre otros.
- Los de **velocidad completa**, que pertenecen a la versión 1.1. Su tasa de transferencia es de hasta 12 Mbps; fueron la versión más rápida antes de la llegada del 2.0.
- Los de **alta velocidad**, llamados 2.0, poseen una tasa de transferencia de hasta 480 Mbps, pero, en la realidad, no alcanzan los 280 Mbps. El cable implementado para esta versión utiliza líneas: dos de datos y dos de alimentación.
- Los de **super alta velocidad**, denominados 3.0, poseen una tasa de transferencia de hasta 4.8 Gbps. Su velocidad de bus es diez veces mayor a la de su antecesor, el 2.0. Esto se debe a la inclusión de cinco contactos adicionales.

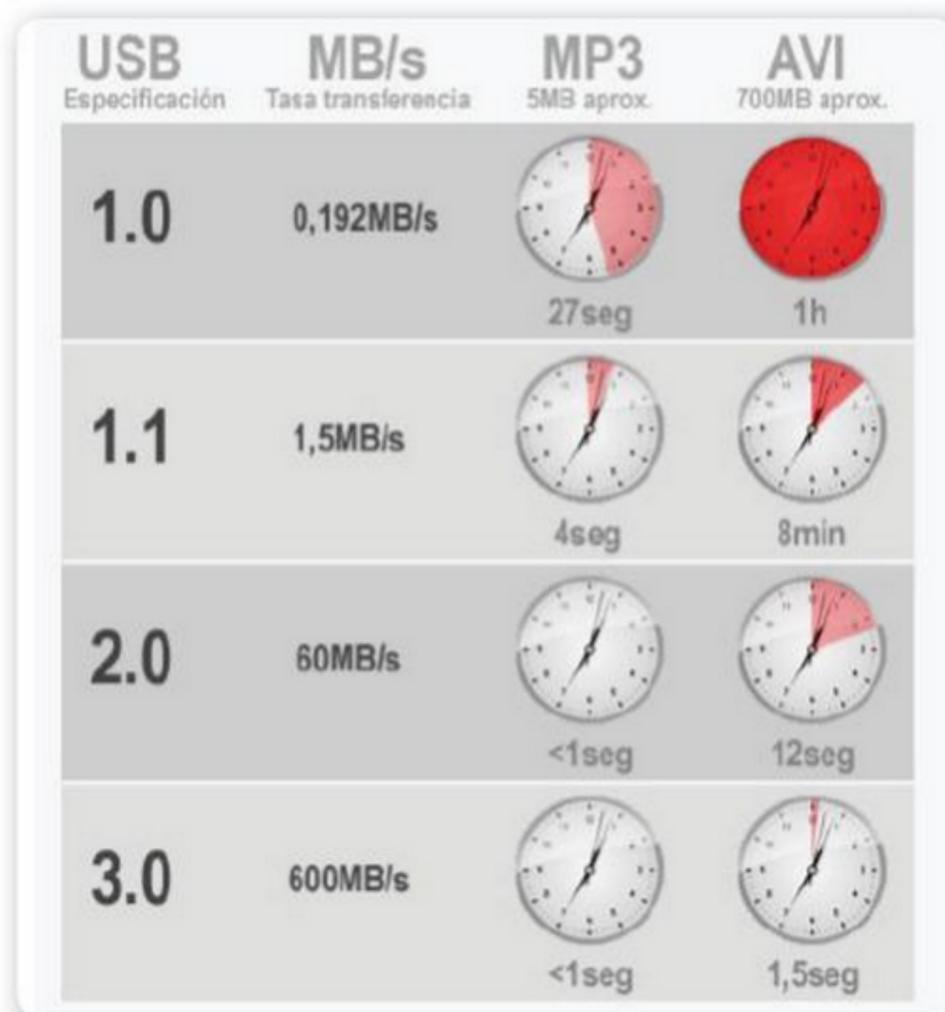


Figura 27. Los distintos estándares de USB comparados de acuerdo a su velocidad de transferencia para distintos tipos de archivos.

Diagrama de capas

En la conexión se pueden apreciar dos partes: el controlador o host y el dispositivo o periférico.

El software cliente se ejecuta en el host, y el sistema operativo se vincula con el dispositivo. El controlador anfitrión USB está constituido por el hardware y el software que permite a los dispositivos USB conectarse al anfitrión.

La conexión entre el host y el dispositivo requiere la interacción entre distintas capas. La capa de interfaz del bus USB proporciona la conexión entre el host y el dispositivo. La capa del dispositivo USB es la que permite la interacción entre el sistema y el dispositivo. La capa de función proporciona las capacidades adicionales necesarias al host mediante una adecuada capa de software cliente.

Las capas de función y de dispositivos tienen una comunicación lógica a su nivel, aunque la comunicación entre estas dos capas se realiza por la capa de la interfaz del bus.

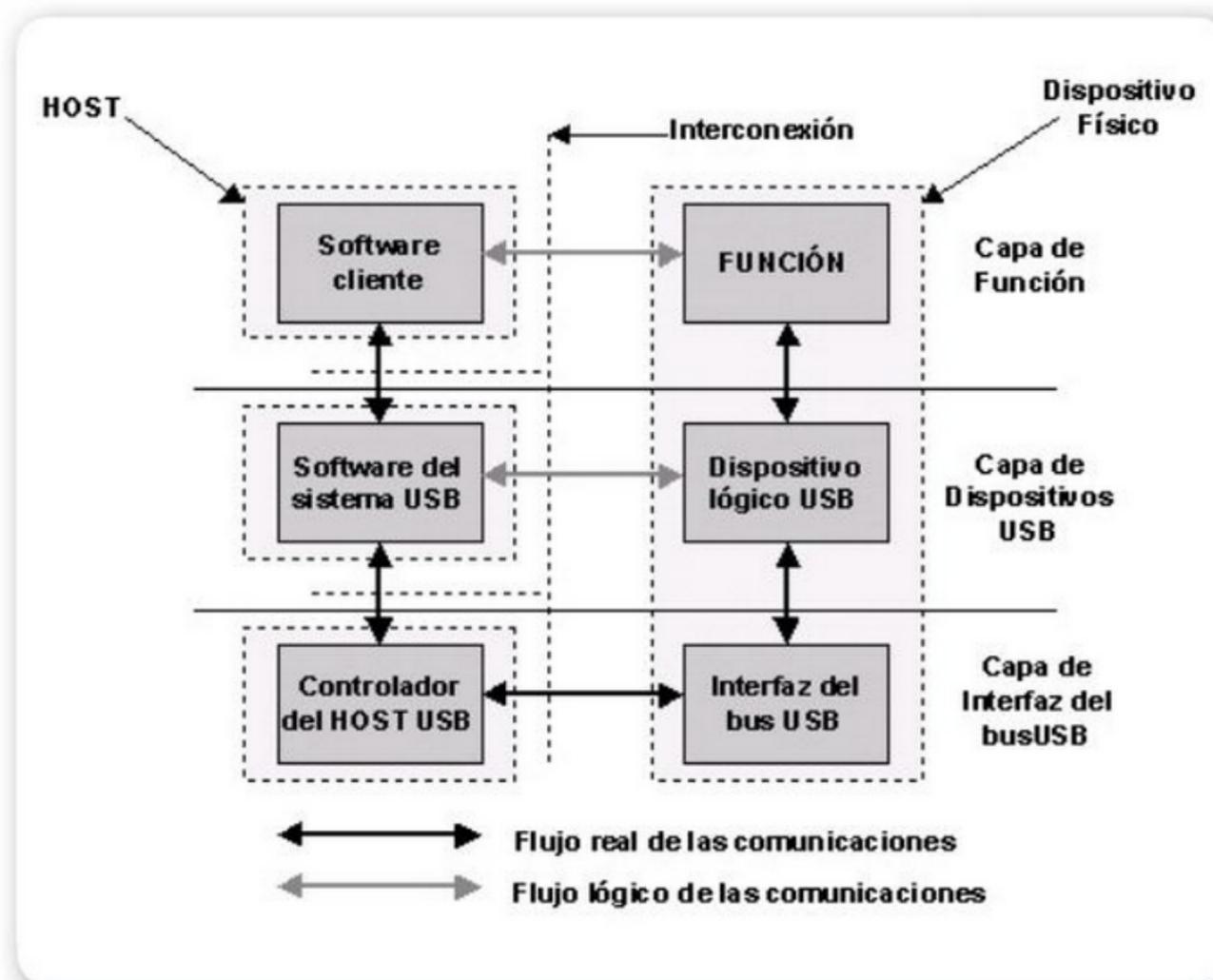


Figura 28. Distintas capas que intervienen en la comunicación USB: una es el host y, la otra, el dispositivo físico.

Los **drivers** o **controladores** son programas que sirven como intérpretes entre el sistema operativo y los dispositivos que se pueden interconectar en una computadora. Sin estos, ningún sistema operativo o programa podrían comunicarse con los periféricos o dispositivos.

El software del USB es el que permite la compatibilidad con el sistema operativo; hay ciertos dispositivos que llevan drivers genéricos y otros que poseen drivers dedicados, que, en muchos casos, el sistema operativo no detecta de manera automática.



CONCENTRADORES O HUBS USB

Son distribuidores inteligentes de datos y alimentación que conectan 127 dispositivos a un único puerto USB. Reparten de forma selectiva la información. Permiten también las comunicaciones desde el periférico hacia la PC, con un sistema full-duplex. Los que no poseen alimentación externa, toman la energía del propio USB. Los que poseen dicha fuente admiten más cantidad de puertos.



Figura 29. Esquema de una instalación de un driver USB en el sistema operativo Windows.

Puerto USB

El **puerto USB** es una entrada de tipo rectangular, a través de la cual el usuario tiene acceso a los contenidos del dispositivo y, a su vez, puede compartir la información que se encuentra almacenada en él. Los dispositivos conectados a estos puertos no necesitan una alimentación externa para poder funcionar, porque el puerto está diseñado para alimentar al dispositivo conectado en él.

Por otro lado, es posible conectar varios dispositivos al mismo tiempo, sin tener que recurrir a una fuente externa de alimentación.

Una de las principales características del puerto es una capacidad denominada **plug and play**. Esta denominación hace referencia a la cualidad que tiene el puerto USB, por la cual, con solo conectar el dispositivo al servidor central (una computadora o cualquier herramienta de trabajo que disponga de un puerto USB), es capaz de traducir la información almacenada y reproducirla.

En otras palabras, hace que el dispositivo y la computadora hablen en el mismo idioma y así se puedan entender mutuamente. A la vez, por esta misma característica, al desconectar o conectar diferentes dispositivos no es necesario reiniciar el equipo en el que

se esté trabajando. La evolución de los puertos USB y su integración en la vida cotidiana es cada vez mayor, porque mejora la velocidad de transferencia, su conexión es extremadamente simple y rápida y no requiere de grandes conocimientos. Hoy en día es muy común ver dispositivos o equipos que posean estos puertos; por eso, podemos decir que ya están incorporados a nuestra vida.

Conectores

Los **conectores USB**, tanto el conector como el receptáculo, se pueden clasificar en dos tipos: los **serie A** y los **serie B**. Los serie A presentan una distribución de los cuatro conectores, correspondientes a los cuatro conductores, en un mismo plano, de forma alineada. Su color es una especie de blanco opaco.

En cambio, los receptáculos se presentan de cuatro maneras distintas: vertical, formando un ángulo recto, en forma de panel y apilado en ángulo recto (montaje pasamuro). Este tipo se utiliza para aquellos dispositivos en los que el cable externo está permanentemente unido a ellos, como teclados, mouses y concentradores o hubs.

Los conectores de la serie B disponen sus contactos en forma distribuida en dos planos paralelos entre sí, con dos conectores en cada plano.

Esta norma se utiliza para aquellos dispositivos que deben tener un receptáculo para conectar un cable USB, como impresoras, módems y escáneres.

LOS CONECTORES
USB SE CLASIFICAN
EN DOS TIPOS:
LOS SERIE A
Y LOS SERIE B



PERIFÉRICOS

Los USB soportan periféricos de velocidad baja o media, por eso para la transmisión de datos emplean dos velocidades: 1.5 y 12 Mbps. De esta manera, se consigue una mejor utilización de los recursos. Para los periféricos de baja velocidad, como teclados, mouses y joysticks, se utiliza la conexión de 1.5 Mbps y, para los monitores, impresoras, módems y equipos de audio, 12 Mbps.

	Tipo A		Tipo B	
	Macho	Hembra	Macho	Hembra
USB estándar				
Mini USB 5 pines				
Mini USB 8 pines				
Micro USB				
USB 3.0 estándar				
Micro USB 3.0				

Figura 30. Diferencia que existe entre los dos tipos de conector con respecto a la versión del estándar USB.

Aplicaciones

En la actualidad, los dispositivos USB se encuentran en los más diversos y cotidianos elementos de nuestra vida.

Las **memorias USB**, también conocidas como **pen drive**, son muy comunes entre las personas que transportan información de un lugar a otro, como desde la casa a la oficina o viceversa. Estos dispositivos tienen la capacidad de almacenar datos durante alrededor de veinte años y de grabarse un millón de veces. Es para destacar que no solo fueron diseñados para guardar datos, sino que es muy común utilizarlos para almacenar programas o archivos de cualquier tipo,



WIRELESS USB

El **Wireless USB**, W-USB o WUSB, es un protocolo de comunicación inalámbrica por radio con gran ancho de banda. Combina la sencillez de uso del USB con la versatilidad de las redes inalámbricas. Como base de radio, utiliza la plataforma Ultra-WideBand, que logra tasas de transmisión de hasta 480 Mbit/s con un alcance de 3 metros, y 110 Mbit/s, de diez metros. Su frecuencia va de 3.1 a 10.6 GHz.

ya que en ese aspecto se comportan como cualquier sistema de almacenamiento de archivos.

Como los pen drive son tan masivos en la actualidad, los costos se han reducido mucho, y esto ha provocado que sean utilizados en forma masiva, desde promociones, marketing e, incluso, como regalos. Por lo general, se utilizan de esta manera en ámbitos relacionados con la industria de la tecnología o la computación. A menudo, se distribuyen de forma gratuita o se hacen sorteos, cuyos premios son estos dispositivos y, en otros casos, se venden por debajo del precio de fabricación o se incluyen con la compra de otro producto o de una marca en particular. También, a veces, se personalizan mediante un grabado en la superficie de la carcasa de la memoria USB. Este grabado puede ser el logotipo de una empresa, el nombre de una marca o, simplemente, el nombre de una persona.

Estos dispositivos pueden incluir o llevar información precargada, como gráficos, documentación pertinente, enlaces de páginas web, aplicaciones, demos o, incluso, algún archivo multimedia. Además, pueden almacenar la información como datos de solo lectura o tener dos particiones, una de solo lectura y otra en donde se pueda grabar y leer. Los que poseen estas dos particiones son un poco más caros.

Por otro lado, las memorias USB se pueden configurar como dispositivos de autoarranque. Esto quiere decir que, con solo insertar el dispositivo en el puerto USB, el sistema operativo lo reconoce y arranca de manera automática un archivo específico, que, en el caso de Windows, se llama `autorun.inf`. La función que permite el autoarranque, en algunas computadoras, se encuentra deshabilitada o, incluso, se la puede deshabilitar de modo manual, ya que de esta manera se dificulta la propagación de virus y troyanos, que se aprovechan de esta función para infectar tanto a los dispositivos USB como a las computadoras.



EL USB OTG

El **USB OTG (On-The-Go)** es una extensión del USB 2.0 que les permite a los dispositivos con puertos USB obtener una mayor flexibilidad en la gestión de dicha conexión. En otras palabras, hace posible conectar cualquier dispositivo mediante USB a un smartphone o tablet, como, por ejemplo, una cámara digital, un pen drive, un módem USB, un teclado o incluso un mouse.

En el sistema operativo Windows pueden correr aplicaciones especiales para estos dispositivos, las cuales permiten tener, en un mismo lugar, muchas aplicaciones juntas e incluso algunas portables. Estas aplicaciones portables son aquellas que se pueden ejecutar sin necesidad de tenerlas instaladas en el disco duro de la computadora. Un ejemplo claro podría ser la página **PortableApps.com**, que en su sección de descargas brinda una aplicación con un conjunto enorme de programas portables.

Otra utilidad de estos dispositivos es que, si el BIOS de la computadora lo admite, pueden arrancar un sistema operativo sin la necesidad de un CD, DVD o, incluso, un disco duro.

Software y utilidades

El software de diagnóstico nos permite acceder a cierta cantidad de información importante para equipos PC, realizando un extenso y detallado análisis para detectar errores del sistema. Principalmente, revisa registros de inicio, disco duro, memoria, procesador, entre otros.

Debe contar con soporte para multiprocesador, detección de dispositivos PCI y gran variedad de diagnósticos que nos permitan identificar el problema o componente que falla, como, por ejemplo: test de CPU (**Core, FPU** y extensiones multimedia **MMX**), test de motherboard (**Controlador DMA, timers, controlador de teclado e interrupciones**), test de puertos series y paralelos (test internos de **handshake** y **loopback**), test de CD-ROM y DVD (velocidad de transferencia, velocidad de búsqueda aleatoria, de audio y de lectura de datos), test de placas de video y monitores (generador de caracteres, memoria, modos de video, etcétera), test de módems e impresoras, teclados y mouses, etcétera.

Entre el software destacado, se encuentra **Docklight**, que sirve como herramienta de simulación para los protocolos de comunicación serial RS-232, RS-485/422 y otros, y ofrece la posibilidad de supervisar la comunicación entre dos dispositivos seriales o probar la comunicación serial de un solo dispositivo.

Como características, podemos destacar las siguientes:

- **Simular protocolos seriales:** Docklight puede enviar secuencias definidas por el usuario según el protocolo utilizado y reaccionar a las secuencias entrantes. Esto permite simular el comportamiento de un dispositivo de comunicación serial, específicamente útil para generar las condiciones de prueba que son duras de reproducirse con el dispositivo original.
- **Datos de registro RS-232 :** todos los datos de la comunicación serial se pueden registrar usando dos formatos de archivo: texto llano, para las enormes cantidades de registro, o un archivo de HTML, con el texto labrado que permite distinguir de manera fácil entre los datos entrantes y salientes o la información adicional.
- **Detección de secuencias específicas de los datos :** en múltiples casos de prueba, necesitaremos comprobar si hay una secuencia específica dentro de los datos RS-232, que señalan una condición del problema. En este sentido, Docklight maneja una lista de secuencias de los datos y puede realizar acciones definidas por el usuario después de detectar una secuencia.

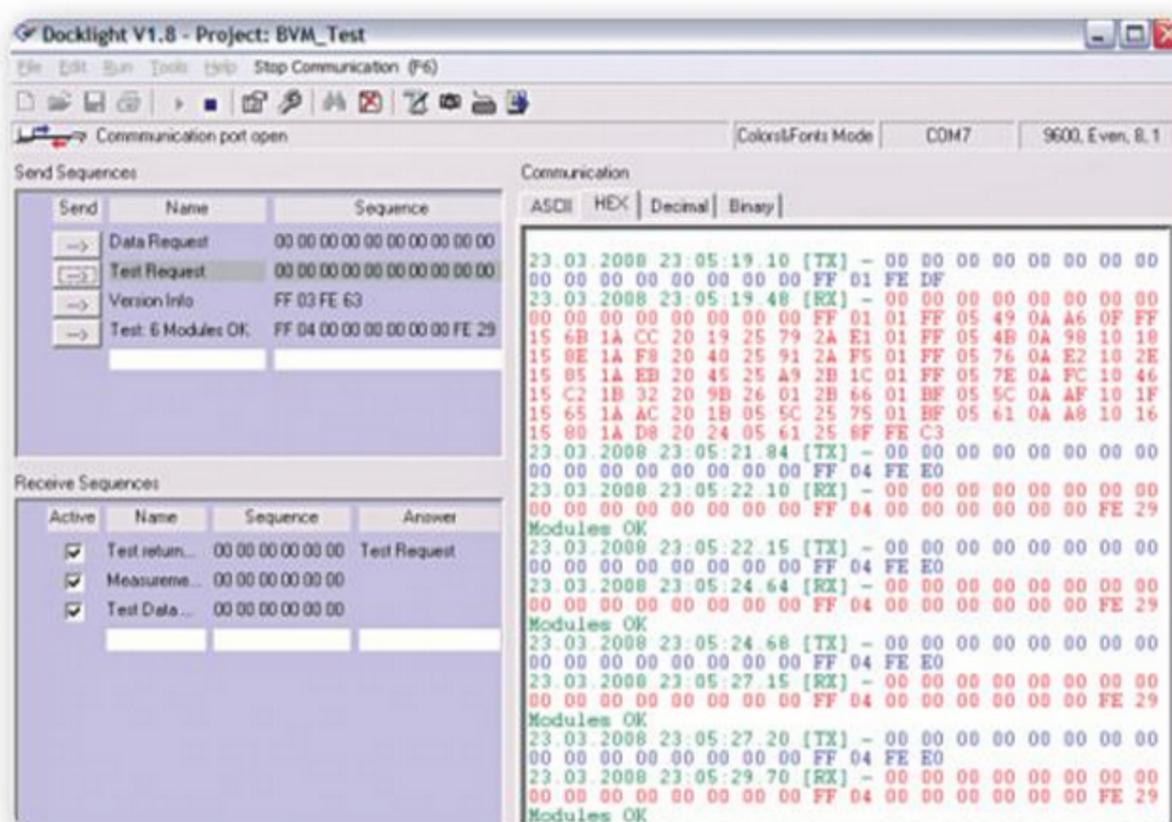


Figura 31. Docklight ofrece la posibilidad de supervisar la comunicación entre dos dispositivos seriales o probar la comunicación serial de uno solo.

Netcat es una herramienta de red desarrollada en el año 1996 por Hobbitt y liberada bajo licencia de software libre permisiva para UNIX. Realiza y acepta conexiones TCP y UDP, es decir, escribe y lee los datos

en este tipo de conexiones hasta que se cierren. Proporciona un subsistema de conexión a red básico basado en TCP/UDP, que permite a los usuarios interactuar en forma normal o mediante secuencias de comando con aplicaciones de red y servicios sobre la capa de aplicación. Hace posible ver datos TCP y UDP en bruto, antes de que sean recubiertos por la siguiente capa superior, tal como FTP, SMTP o HTTP. Una de sus características principales es que es construido en capacidades de escaneo de puertos con aleatoriedad.

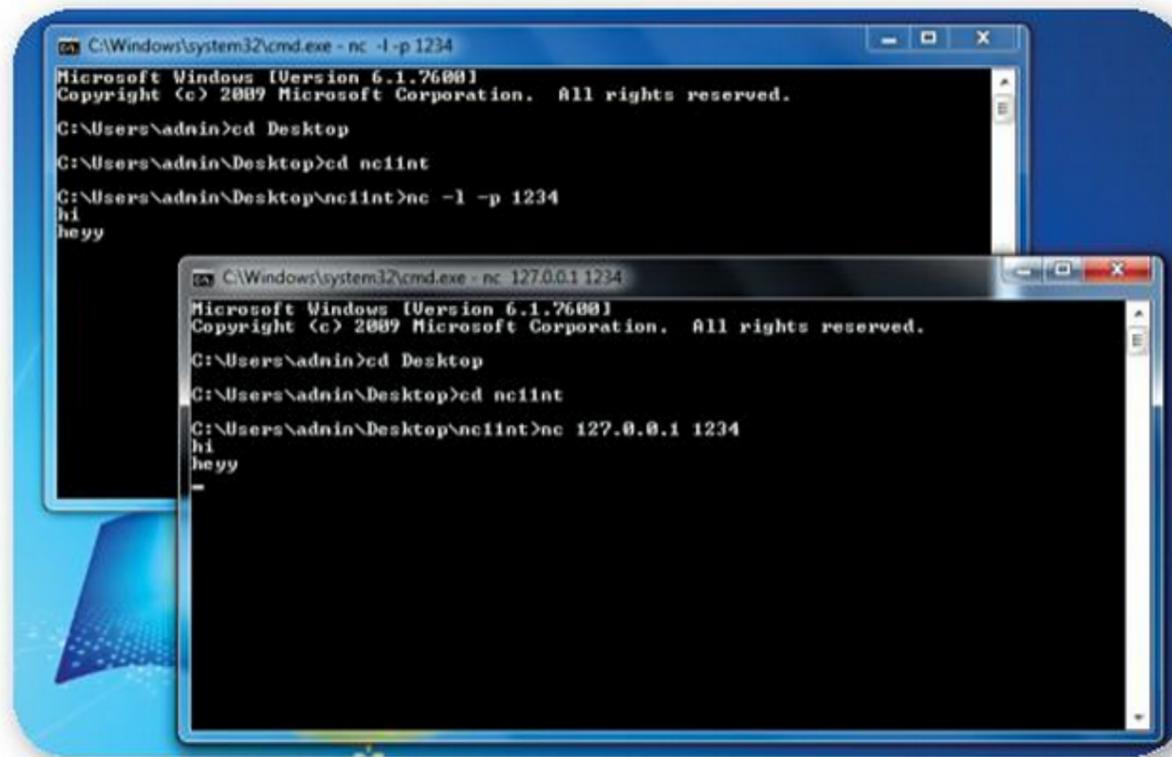


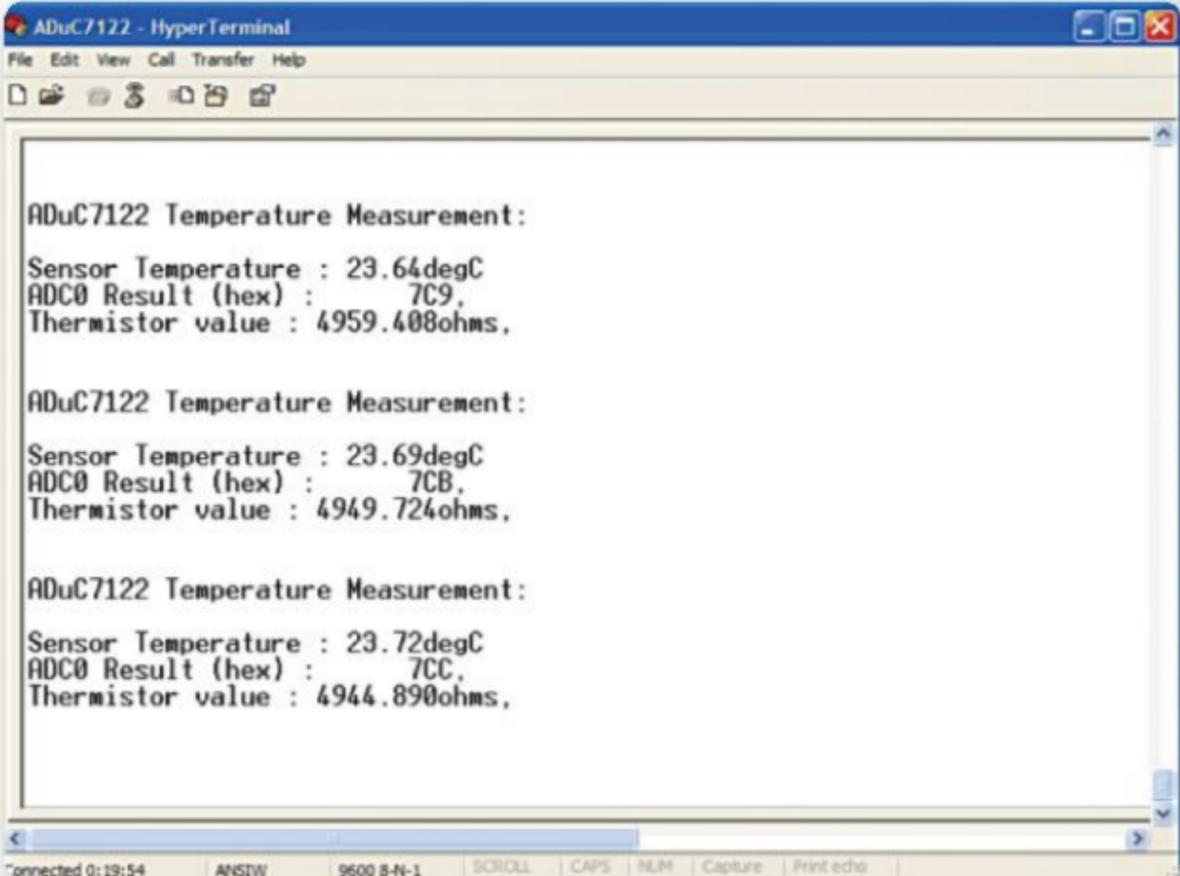
Figura 32. Netcat es una herramienta de red que permite, a través de intérpretes de comando y sintaxis sencilla, abrir puertos TCP/UDP en un host.

EL SOFTWARE DE
DIAGNÓSTICO ACCEDA
A INFORMACION
IMPORTANTE PARA
EQUIPOS PC



Hyperterminal es un programa diseñado para realizar las funciones de comunicación y de emulación de terminal. Permite conectar dispositivos informáticos o computadoras, sitios telnet de internet, servicios en línea y otros, por medio de un módem. Para hacer esto, no necesita usar comandos de línea y es un medio útil para configurar conexiones con otros sitios web. Permite enviar cualquier tipo de información o archivos pesados, ya sean documentos,

imágenes, videos, entre otros, a través del puerto serie. Una de sus ventajas es que no necesita un equipo especial adicional para enviar información; solo requiere tener una conexión para módem.



```
AduC7122 - HyperTerminal
File Edit View Call Transfer Help
AduC7122 Temperature Measurement:
Sensor Temperature : 23.64degC
ADC0 Result (hex) : 7C9,
Thermistor value : 4959.408ohms,

AduC7122 Temperature Measurement:
Sensor Temperature : 23.69degC
ADC0 Result (hex) : 7CB,
Thermistor value : 4949.724ohms,

AduC7122 Temperature Measurement:
Sensor Temperature : 23.72degC
ADC0 Result (hex) : 7CC,
Thermistor value : 4944.890ohms,

Connected 0:19:54 ANSIV 9600 8-N-1 SCROLL CAPS NUM Capture Print echo
```

Figura 33. HyperTerminal realiza funciones de emulador, haciendo posible el uso de los recursos de otro equipo mediante una conexión.

TelNet (*Telecommunication Network*) es una herramienta de protocolo de red que nos permite realizar conexiones de manera remota a una PC, para ejecutar programas en ella, arreglar fallas a distancia, consultar datos que sean importantes, entre otras tareas. Recibe la entrada/salida de los datos.

De todas maneras, es importante destacar que solo sirve para acceder en modo terminal, es decir, sin gráficos.

Aparte de sus variados usos, se ha utilizado (y aún hoy se puede usar en su variante SSH) para abrir una sesión con una máquina UNIX.



MÁS SOFTWARE

Además de los programas mencionados, existe una amplia variedad de software como herramienta de verificación. **Aida 32** es un programa que tiene gran utilidad en el diagnóstico de hardware, ya que genera información detallada de los componentes instalados en una PC. **CheckDrive**, por otra parte, analiza de una manera simple los discos duros en busca de errores; puede detectarlos y repararlos.

PAP. CONEXIÓN CON DISPOSITIVOS POR SOFTWARE



01 Abra el intérprete de comandos, más conocido como CMD, en Windows, y escriba `ctelnet <ip del server> <Puerto del server>` , como se aprecia en la imagen.

```
Microsoft Windows [Versión 6.1.7600.1]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.
C:\Users\Luiggy>telnet 192.168.0.1 23
```

02 Ingrese el usuario y la contraseña del dispositivo (servidor), pulse ENTER. Estará listo para trabajar con el dispositivo.

```
login :
```

03 La lista de comandos disponibles mediante TelNet es amplia, aunque varía según el tipo de dispositivo al cual esté conectado; en el ejemplo, la conexión es a un Router.

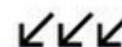
```
Menu 3.2 - TCP/IP and DHCP Setup

DHCP Setup
DHCP= None
Client IP Pool Starting Address= N/A
Size of Client IP Pool= N/A
Primary DNS Server= 80.58.61.250
Secondary DNS Server= 80.58.61.254
Remote DHCP Server= N/A
TCP/IP Setup:
IP Address= 192.168.1.2
IP Subnet Mask= 255.255.255.0
RIP Direction= Both
  Version= RIP-2B
Multicast= IGMP-v2
IP Policies=
Edit IP Alias= No

Press ENTER to Confirm or ESC to Cancel: █
```



RESUMEN



En este capítulo, evidenciamos la necesidad del uso de protocolos de comunicación para generar conexiones efectivas y desarrollamos ejemplos sobre el estándar de protocolos de red más difundido del mundo: TCP/IP. También analizamos la ventaja de utilizar técnicas de modulación y su correspondiente demodulación, y algunos estándares actuales, como I²C, SMBus y SPI. Además, estudiamos la comunicación serie y el estándar RS-232, así como el paso a estándares más flexibles y veloces, como USB (Universal Serial Bus), con sus funciones, conectores, usos y evolución. Por último, presentamos algunos programas y utilidades de diagnóstico usadas comúnmente para el análisis de conexiones de dispositivos electrónicos por cable, y realizamos una práctica de conexión de un dispositivo por medio del uso de software.

Actividades

TEST DE AUTOEVALUACIÓN

- 1 ¿Qué son los **protocolos** y las normas que debe seguir una máquina o interfaz para realizar una comunicación?
- 2 ¿Cuáles son las características del **protocolo TCP/IP**?
- 3 ¿Cuáles son las analogías posibles entre los **protocolos de internet** y el modelo de **capas OSI**?
- 4 ¿Qué es una **modulación de señal**?
- 5 ¿Cómo se aplica el concepto de **sincronismo** en los buses a distintos niveles?



PROFESOR EN LÍNEA



Si tiene alguna consulta técnica relacionada con el contenido, puede contactarse con nuestros expertos: profesor@redusers.com



Conectividad inalámbrica

Transmitir datos por un medio físico que no sea el aire fue la manera trivial de comunicarnos, hasta que alguien descubrió que las ondas electromagnéticas atraviesan el aire. En este capítulo, veremos los fundamentos de la comunicación sin cables y cómo ha transformado la manera de comunicarnos en la actualidad.

▼ Comunicación sin cables220	▼ Wi-Fi.....238
▼ Antenas223	▼ WiMAX.....242
▼ Comunicaciones a corta distancia225	▼ Comunicaciones a larga distancia.....244
▼ Bluetooth228	▼ Sistemas satelitales248
▼ Comunicaciones a media distancia232	▼ Resumen.....253
▼ VHF236	▼ Actividades.....254





Comunicación sin cables

La **tecnología inalámbrica** se originó a principios del siglo XX, con el uso de la radiotelegrafía desarrollada por Guillermo Marconi. Más tarde, con el descubrimiento de la modulación de ondas, se logró transmitir voces y música. El medio descubierto fue llamado **radio**.

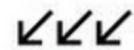


Figura 1. Dispositivo de telegrafía sin hilos (**TSH**), desarrollado por Guillermo Marconi. Podemos ver las baterías a la derecha y las esferas de transmisión a la izquierda.

La tecnología inalámbrica para la comunicación de datos se originó como un experimento en 1979 en los laboratorios de IBM en Suiza. Este experimento se desarrolló con el fin de crear una red local mediante



RÉCORD MUNDIAL



Científicos alemanes que investigan física aplicada de estado sólido lograron transmitir datos a una velocidad de 40 gigabits por segundo a una distancia de un kilómetro. Para alcanzar esta velocidad, diseñaron circuitos integrados activos, compuestos por semiconductores de alta movilidad, y los datos se transmitieron mediante una señal de radio a una frecuencia de 240 GHz.

señales infrarrojas; el resultado fue el punto de partida para la evolución de las distintas redes inalámbricas que conocemos hoy.

La transmisión de datos inalámbrica normalmente trabaja con ondas de radio de baja potencia sobre una banda específica para comunicar los distintos dispositivos entre sí. Esta comunicación facilita la conexión a dispositivos electrónicos que no tienen una ubicación fija o, simplemente, para evitar el cableado, como, por ejemplo, en edificios de oficinas, hoteles o restaurantes.

La comunicación sin cables o wireless no solo se aplica al campo de la informática sino también a los sistemas de domótica, seguridad, televisión, telefonía, medicina, mediante dispositivos de baja potencia a corta distancia, como los transmisores ZigBee o infrarrojos.

Tecnología Wi-Fi

La tecnología Wi-Fi fue diseñada con el objetivo de ofrecer compatibilidad entre distintos dispositivos y fomentar la conectividad wireless; permitió que smartphones, notebooks, consolas de videojuegos, reproductores de música, sistemas de seguridad, sistemas de domótica y Smart TV, entre otros, se conectaran a internet desde un punto de acceso inalámbrico o compartieran datos mediante una red inalámbrica (*wireless LAN*).

La tecnología Wi-Fi es una marca de la corporación Wi-Fi Alliance y, originalmente, se denominó **IEEE802.11 de secuencia directa**.

VELOCIDADES		
▼ PROTOCOLO	▼ FRECUENCIA DE TRABAJO	▼ VELOCIDAD MÁXIMA
802.11	2.4 GHz	2 Mbits/s
802.11 ^a	5G Hz	54 Mbits/s
802.11b	2.4 GHz	11 Mbits/s
802.11g	2.4 GHz	54 Mbits/s
802.11n	2.4 GHz y 5 GHz	600 Mbits/s

Tabla 1. Velocidad de algunos protocolos Wi-Fi que fueron certificados por el IEE (Instituto de Ingenieros Eléctricos y Electrónicos).

Tecnología WiMAX

La tecnología WiMAX (interoperabilidad mundial para acceso por microondas) permite la transmisión inalámbrica de datos con una cobertura de hasta 80 km con una velocidad de transferencia de 75 Mbits/seg. Fue creada por las empresas Intel y Alvarion en

EN EL AÑO 2011,
LA IEE APROBÓ
WIMAX 2, BAJO
EL ESTÁNDAR
IEE 802.16M

el año 2002 y certificada por el IEE (Instituto de Ingenieros Eléctricos y Electrónicos) bajo el estándar IEE 802.16. Esta tecnología trabaja en el espectro de frecuencia de los 2.3 a 3.5 GHz.

Una de las ventajas de esta tecnología es que permite brindar el servicio de banda ancha en zonas rurales, donde el índice demográfico es muy bajo y los costos por usuario serían muy altos si se desplegara una red cableada o de fibra óptica.



Tecnología GPRS y 3G

La **telefonía móvil**, también llamada **telefonía celular** (debido a que el sistema funciona mediante una red de celdas, donde cada antena repetidora de señal es una célula), es un sistema de comunicación inalámbrico que opera en el espectro de frecuencia de los 2.4 GHz.

La telefónica móvil surgió comercialmente en la década del 80. El primer teléfono móvil fue presentado en 1973 por Martin Cooper; sin embargo, recién en 1979 la empresa **NTT** (*Nippon Telegraph and Telephone Corporation*) lanzó las primeras versiones comerciales en Tokio, Japón.

La tecnología GPRS (servicios generales de paquetes por radio) es una extensión del sistema GSM (sistema de comunicaciones móviles) creado en la década del 80. Está ubicada entre la tecnología de segunda (2G) y tercera generación (3G) de tecnología móvil digital.

Admite una transmisión de datos de hasta 114 Kbps mediante redes de telefonía celular. Esto nos permite usar el teléfono celular o la computadora de bolsillo para navegar por internet, descargar datos, revisar nuestro correo, realizar videollamadas y enviar SMS (mensajes de texto) y MMS (mensajes multimedia) a nuestros contactos. La tecnología 3G ofrece mayor seguridad, velocidad y versatilidad, comparada con la tecnología GPRS.

La velocidad de transmisión de datos del sistema 3G puede superar los 3 Mbits/s.

Antenas

Las antenas son conductores metálicos que cumplen la función de emitir y recibir señales electromagnéticas o de radiofrecuencia en el espacio, ya sea espacio abierto o entre obstáculos (árboles, casas o partes irregulares de terreno).

Una **antena receptora** transforma señales electromagnéticas en voltaje para que el equipo de recepción pueda descifrar la señal e interpretarla mediante imágenes, sonidos o datos; en cambio, las **antenas emisoras** hacen el proceso inverso, es decir que convierten el voltaje en señales electromagnéticas. Sin embargo, la mayoría de las antenas puede cumplir ambas funciones sin problemas.

Patrón de radiación

El **patrón de radiación** es la energía que la antena puede radiar en el espacio abierto. Normalmente, se representa de dos formas: **acimut** y **elevación**. El patrón acimut es una gráfica de la energía radiada vista desde arriba, mientras que el patrón elevación es un gráfico de la energía radiada vista de perfil; ambos datos pueden ser representados mediante una gráfica tridimensional.

Existen diferentes tipos de antenas, de acuerdo con el tipo de señal o área donde se quiera radiar o cubrir. Veremos las que están agrupadas en variantes de directividad o cobertura; estas son las omnidireccionales y direccionales con la subvariante bidireccional.

Omnidireccionales

Son antenas que tienen un amplio rango de radiación horizontal y vertical, es decir, que irradian y captan señal en todas direcciones (en teoría, 360°). Son las que, por lo común, vemos en los routers inalámbricos o dispositivos Wi-Fi.

Figura 2. Pequeña antena omnidireccional que normalmente podemos encontrar en los routers o dispositivos inalámbricos que usamos en nuestros hogares.



Direccionales

Este tipo de antena emite y capta señal de largo alcance, pero concentra toda la potencia en una dirección específica y deja prácticamente nula la emisión o la recepción de señales hacia la otra dirección.

Bidireccionales

Al igual que las antenas direccionales, las bidireccionales tienen áreas específicas (en forma de ocho) en las cuales irradian la mayor parte de su señal, mientras que en otras áreas la señal es nula.

Ganancia de las antenas

La ganancia de las antenas es la relación entre la potencia

SIEMPRE
DEBEMOS ESCOGER
LA ANTENA QUE
MÁS DBD DE
GANANCIA TENGA

que entra en la antena y la potencia que sale de esta, y se expresa normalmente en dBi (decibelio isótropo).

Es importante saber que las medidas en dBi a veces son engañosas, ya que están calculadas para una antena en teoría ideal o en base a antenas isotrópicas (que tienen un perfecto rango de radiación a su alrededor), y no para antenas como las que tenemos normalmente en nuestro hogar.

Lo que realmente nos debe importar al ver la ganancia en las antenas son los dBd. Esta medida de ganancia está calculada para una antena estándar; la ganancia dBd siempre será mucho menor que la ganancia dBi.



ANTENAS DE INTERNET SATELITAL



Para internet satelital se usan antenas direccionales parabólicas que son las que mayor ganancia tienen, ya que el plato refleja el 60 % de los datos en el foco o dispositivo central de recepción (LNB). Pero solo tienen la capacidad de recibir señales, y, para poder navegar por internet, el envío de señales se debe realizar por otro medio, como líneas cableadas o puntos de acceso inalámbricos.

Comunicaciones a corta distancia

La comunicación mediante **infrarrojo** es un medio de comunicación que usa rayos de luz dentro del espectro no visible para el ojo humano. A continuación, veremos cómo funciona esta tecnología. Los infrarrojo fueron descubiertos en el año 1800 por William Herschel, un astrónomo inglés.

Herschel realizó este hallazgo al colocar un termómetro en el espectro obtenido por un prisma de cristal, con el fin de medir el calor emitido por cada color. Observó que el calor era más fuerte al lado del rojo del espectro y que allí no había luz; con esto, descubrió que el calor puede transmitirse por una forma invisible de luz. Herschel nombró a esta radiación como **rayos calóricos**, y esta denominación fue bastante utilizada a lo largo del siglo XIX para, finalmente, dar paso al nombre moderno de **radiación infrarroja**.

LA TECNOLOGÍA IRDA
FUE MUY UTILIZADA
EN MÚLTIPLES
DISPOSITIVOS
A FINES DE LOS 90



Comunicación infrarroja o IrDA

IrDA (*Infrared Data Association*) es una organización creada con el fin de desarrollar normas para software y hardware empleados para la transmisión inalámbrica de datos mediante infrarrojo. Fue fundada en 1993 por IBM, Sharp y HP, entre otros, y diseñó la tecnología que lleva su mismo nombre. La tecnología IrDA permite realizar una conexión inalámbrica mediante luz infrarroja. La velocidad de transmisión puede llegar hasta 4 Mbit/s con un alcance de un metro.

En los últimos años, la tecnología IrDA fue desplazada, de manera gradual, por Bluetooth y Wi-Fi.

Usos de la tecnología infrarroja

Un uso muy común de la tecnología infrarroja es en los controles remotos de los televisores, que utilizan infrarrojo en vez de ondas de radio, ya que estos no interfieren con otras señales.

Los infrarrojos eran muy utilizados, también, para comunicar a corta distancia algunas computadoras con sus periféricos.

Los aparatos que utilizan este tipo de comunicación cumplen, por lo general, un estándar publicado por IrDA. La luz utilizada en las fibras ópticas es de infrarrojo.

El infrarrojo se utiliza en los equipos de visión nocturna, cuando la cantidad de luz disponible no es suficiente para ver los objetos. La radiación se recibe y después se refleja en una pantalla. Los objetos con más calor se convierten en los más luminosos.



Figura 3. Cámara con visión infrarroja. Este tipo de cámaras están equipadas con LEDs infrarrojos que le permiten, al lente de la cámara, “ver” durante la noche.

Redes por infrarrojo

Las redes por infrarrojo permiten la comunicación entre dos nodos a través de una serie de LEDs infrarrojos. Se trata de emisores/receptores de las ondas infrarrojas entre ambos dispositivos. Cada dispositivo necesita al otro para realizar la comunicación; por esto, es escasa su utilización a gran escala. Esta es su principal desventaja, a diferencia de otros medios de transmisión inalámbricos como Bluetooth, wireless, etcétera.

Comunicación en modo semidifuso

La comunicación en modo difuso es un método de emisión radial; es decir, que cuando una estación emite una señal óptica, esta puede ser recibida por todas las estaciones al mismo tiempo en la célula. En el modo semidifuso, las estaciones se comunican entre sí por medio de superficies reflectantes (por ejemplo, espejos curvos). No es necesaria la línea de visión entre dos estaciones, pero sí debe existir una línea de visión con la superficie de reflexión.

Además, es recomendable que las estaciones estén cerca de la superficie de reflexión, que puede ser pasiva o activa.

En las células basadas en reflexión pasiva, el reflector debe tener altas propiedades reflectivas y dispersivas, mientras que, en las basadas en reflexión activa, se requiere de un dispositivo de salida reflexivo, conocido como satélite, que amplifica la señal óptica. La reflexión pasiva requiere más energía por parte de las estaciones, pero es más flexible de usar.

LA TECNOLOGÍA
INFRARROJA
PUEDE SOPORTAR
EL ANCHO DE BANDA
DE ETHERNET



Comunicación en modo difuso

El poder de salida de la señal óptica de una estación debe ser suficiente para llenar la totalidad del cuarto mediante múltiples reflexiones, en paredes y obstáculos. Por lo tanto, la línea de vista no es necesaria y la estación se puede orientar hacia cualquier lado. El modo difuso es el más flexible en términos de localización y posición de la estación; sin embargo, esta flexibilidad es a costa de excesivas



CÓMO PROBAR UN DIODO INFRARROJO



Si utilizamos un puerto de infrarrojos (IR) con un emisor de luz de diodo (LED) y necesitamos probarlo, podemos usar los dispositivos de acoplamiento de carga (CCD) que vienen en algunas cámaras digitales y teléfonos celulares y son sensibles a la luz infrarroja. Para probar el LED, usamos una cámara digital o celular en una habitación poco iluminada para que sea más fácil observar la luz.

emisiones ópticas. Por otro lado, la transmisión punto a punto es la que menor poder óptico consume, pero no debe haber obstáculos entre las dos estaciones. Por eso, es más recomendable y más fácil de implementar el modo de radiación semidifuso.

Bluetooth

La **tecnología Bluetooth** fue diseñada especialmente para dispositivos de bajo consumo que no requieren un gran alcance de transmisión (por ejemplo, para telefonía celular, cámaras digitales, impresoras, mouses o teclados).

BLUETOOTH

POSIBILITA LA

TRANSMISIÓN DE

VOZ Y DATOS, CON UN

ENLACE DE 2.4 GHZ

Está basada en transceptores (circuitos electrónicos transmisores y receptores) de radio de baja potencia y bajo costo.

Esta tecnología fue presentada oficialmente en 1998, aunque antes estuvo en desarrollo en los países escandinavos. Tanto el nombre Bluetooth como el logo asociado a él han sido registrados por **SIG** (*Bluetooth Special Interest Group*).



Numerosas empresas pertenecen a SIG, incluyendo Ericsson, IBM, Microsoft, Intel, Nokia, Toshiba y Agere. La asociación comercial de SIG continúa trabajando para desarrollar e investigar el uso de la tecnología Bluetooth en las industrias de la informática, la automatización, la automoción y las telecomunicaciones.

El nombre Bluetooth proviene de **Harald Bluetooth**, vikingo y rey de Dinamarca desde el año 940 hasta 981, que fue reconocido por su capacidad de ayudar a la gente a comunicarse. Durante su reinado, unificó y convirtió al cristianismo a Dinamarca y a Noruega.



COMUNICACIÓN INFRARROJA PUNTO A PUNTO



Los patrones de radiación del emisor y del receptor deben estar lo más cerca posible y su alineación tiene que ser la correcta. Como resultado, el modo punto a punto requiere una línea de visión entre las dos estaciones que se van a comunicar; de lo contrario, la comunicación quedará trunca o será errónea.

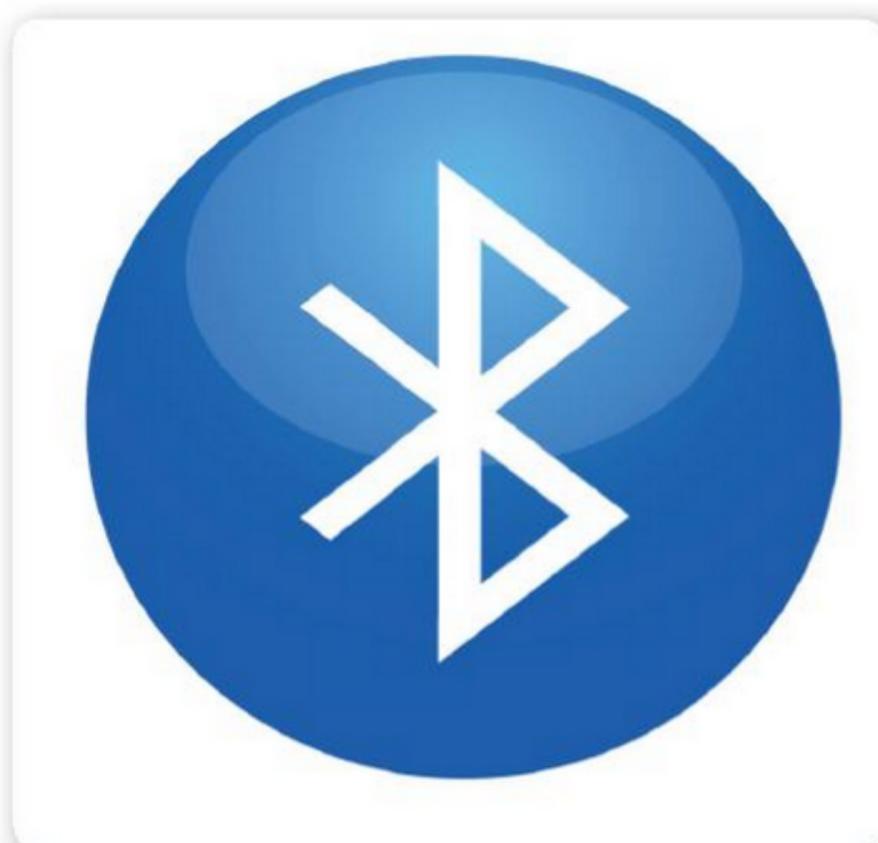


Figura 4. El logo de Bluetooth está basado en las runas de las iniciales del nombre y el apellido de Harald Bluetooth.

Características

La especificación de Bluetooth define un canal de comunicación a un máximo de 720 kbits/s (1 Mbit/s de capacidad bruta) con un rango óptimo de 10 m (opcionalmente, 100 m con repetidores). Opera en la frecuencia de radio de 2.4 a 2.48 GHz con amplio espectro y saltos de frecuencia, con la posibilidad de transmitir en full-duplex con un máximo de 1600 saltos por segundo. Los saltos de frecuencia se dan entre un total de 79 frecuencias, con intervalos de 1 MHz; esto permite dar seguridad y robustez.

La potencia de salida para transmitir a una distancia máxima de 10 metros es de 0 dBm (1 mW), mientras que la versión de largo alcance transmite entre 20 y 30 dBm (entre 100 mW y 1 W).



RELOJES INTELIGENTES CON BLUETOOTH



Casio ya se ha aventurado en el terreno de los relojes de la familia G-Shock. Recientemente, lanzó dos nuevos modelos G-Shock, cuyos códigos son GB-6900B y GB-X6900B, que cuentan con Bluetooth 4.0 de bajo consumo para conectarse a dispositivos Android.

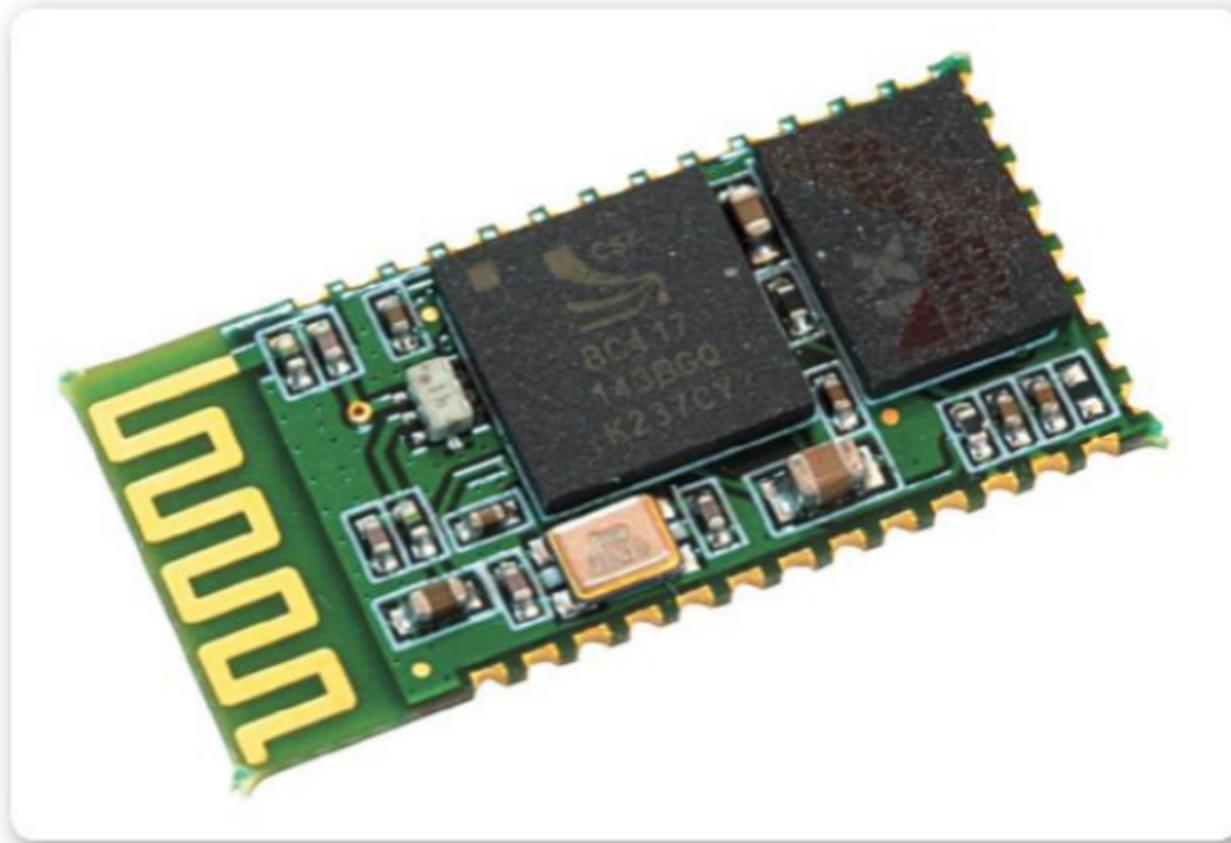


Figura 5. Módulo Bluetooth. Esta pequeña placa es usada normalmente para transmisión-recepción de datos entre circuitos con microcontroladores.

Alcance y velocidad de los dispositivos Bluetooth

Bluetooth permite transmitir datos y voz mediante un canal inalámbrico de radiofrecuencia que usa la banda de los 2.4 Ghz. El alcance para la transmisión de datos es definido por la potencia o clase del dispositivo Bluetooth.

CLASES DE DISPOSITIVOS BLUETOOTH			
▼ CLASE	▼ ALCANCE APROXIMADO	▼ POTENCIA MÁXIMA (MW)	▼ POTENCIA MÁXIMA (DBM)
Clase 1	100 m	100 mW	20 dBm
Clase 2	10 m	2.5 mW	4 dBm
Clase 3	1 m	1 mW	0 dBm

Tabla 2. Comparación de las características de las diferentes clases de dispositivos Bluetooth.



Figura 6. Este tipo de auricular Bluetooth es muy útil, ya que funciona como un dispositivo de manos libres inalámbrico y se puede acoplar a cualquier equipo que tenga esta tecnología.

ANCHO DE BANDA SEGÚN VERSIÓN	
▼ VERSIÓN	▼ ANCHO DE BANDA
Versión 1.2	1 Mbit/s
Versión 2.0	3 Mbits/s
Versión 3.0	24 Mbits/s
Versión 4.0	24 Mbits/s

Tabla 3. El ancho de banda es definido por la versión del transceptor del dispositivo Bluetooth.

Desde que llegó la versión 1.1, la tecnología Bluetooth ha seguido avanzando. La versión 2.1 fue lanzada en 2003, la versión 2.0 **EDR** (*Enhanced Data Rate*) salió al mercado en 2005 y la versión 2.1 EDR, en 2007. En 2009, fue lanzada la 3.0 HS (alta velocidad). Cuanto más avanza la tecnología, los datos son transferidos más rápidamente. El consumo de energía para los dispositivos Bluetooth decrece mientras que la fiabilidad de la tecnología mejora.

Bluetooth se utiliza en muchos productos, tales como teléfonos, impresoras, módems y auriculares. Su uso es adecuado cuando hay dos o más dispositivos en un área reducida, sin grandes necesidades de ancho de banda. Se suele utilizar integrado a teléfonos y PDA, por medio de auriculares Bluetooth o de transferencia de archivos. Además, realiza enlaces o vincula distintos dispositivos entre sí.

Bluetooth simplifica el descubrimiento y configuración de los dispositivos, ya que estos pueden indicar a otros los servicios que ofrecen, lo que permite establecer la conexión de forma rápida (esto se refiere solo a la conexión, no a la velocidad de transmisión).



Figura 7. Joystick Bluetooth para smartphone. Con este joystick, sumado a un emulador de antiguas consolas de juegos (Nintendo, NES, Sega Genesis), podemos asegurarnos horas de diversión y nostalgia.

Comunicaciones a media distancia

La **radiofrecuencia** es la característica que define a un grupo particular de ondas electromagnéticas. Llevan distintas denominaciones de acuerdo con el rango de frecuencias en el que se propagan.

La propagación de las ondas electromagnéticas fue descubierta y descrita por J. C. Maxwell y corroborada entre los años 1886 y 1888 por H. R. Hertz. La implementación de esta tecnología por primera vez es atribuida a diferentes personas, como Popov en Rusia, Tesla en Estados Unidos y G. Marconi en el Reino Unido. Sin embargo, el primer sistema de comunicación por radiofrecuencia fue diseñado por Marconi. Esta comunicación fue realizada en el año 1901 y, a su vez, fue la primera emisión radioeléctrica transatlántica.

Clasificación

Las ondas electromagnéticas se pueden clasificar según su banda de frecuencia de trabajo, por lo que se dividen en varias categorías.

Las frecuencias extremadamente bajas o **ELF** (*Extremely Low Frequencies*) se encuentran en la banda entre 3 a 30 Hz, las frecuencias superbajas o **SLF** (*Super Low Frequencies*) son aquellas que están en el intervalo comprendido entre 30 a 300 Hz.

Por otra parte, las frecuencias ultrabajas o **ULF** (*Ultra Low Frequencies*) se hallan en el intervalo de 300 a 3000 Hz.

Las frecuencias muy bajas o **VLF** (*Very Low Frequencies*), por lo general, son utilizadas en comunicaciones gubernamentales o militares y su rango de frecuencias es de 3 a 30 kHz. Las frecuencias bajas o **LF** (*Low Frequencies*) van desde los 30 a 300 kHz y se utilizan principalmente en la navegación aeronáutica y marítima.

Las frecuencias medias o **MF** (*Medium Frequencies*) van desde el intervalo de 300 a 3000 KHz. En este rango, se encuentran las señales de radiodifusión AM, que van desde los 530 a 1605 kHz.

Las frecuencias altas o **HF** (*High Frequencies*) están comprendidas en el intervalo de 3 a 30 MHz. A este rango de frecuencia se lo conoce como frecuencias de onda corta. Es uno de los más utilizados, ya que en él hay una gama importante de radiocomunicaciones, que van desde la radiodifusión, las comunicaciones gubernamentales y militares, y hasta es usado por los radioaficionados y la banda civil.

Otro de los rangos más utilizados es el de frecuencias muy altas o **VHF** (*Very High Frequencies*) que van de los 30 a 300 MHz. Es empleado por la radio móvil, las comunicaciones marinas y aeronáuticas, las transmisiones de FM (que van de 88 a 108 MHz) y, principalmente, por los canales de televisión del 2 al 12, según el estándar europeo.

Las frecuencias ultraaltas o **UHF** (*Ultra High Frequencies*) cubren el rango de 300 a 3000 MHz. Este rango es utilizado por los canales de televisión del 21 al 69 y por servicios móviles de comunicación en tierra, de telefonía celular o de comunicaciones militares.

Las frecuencias superaltas o **SHF** (*Super High Frequencies*) son aquellas que van desde los 3 hasta los 30 GHz. Este rango es muy utilizado en comunicaciones satelitales y radioenlaces terrestres.

Por último, están las frecuencias extremadamente altas o **EHF** (*Extrematedly High Frequencies*). El intervalo en el que están comprendidas es de 30 a 300 GHz. Como los equipos para utilizar este rango son complejos y costosos, no es muy usado.

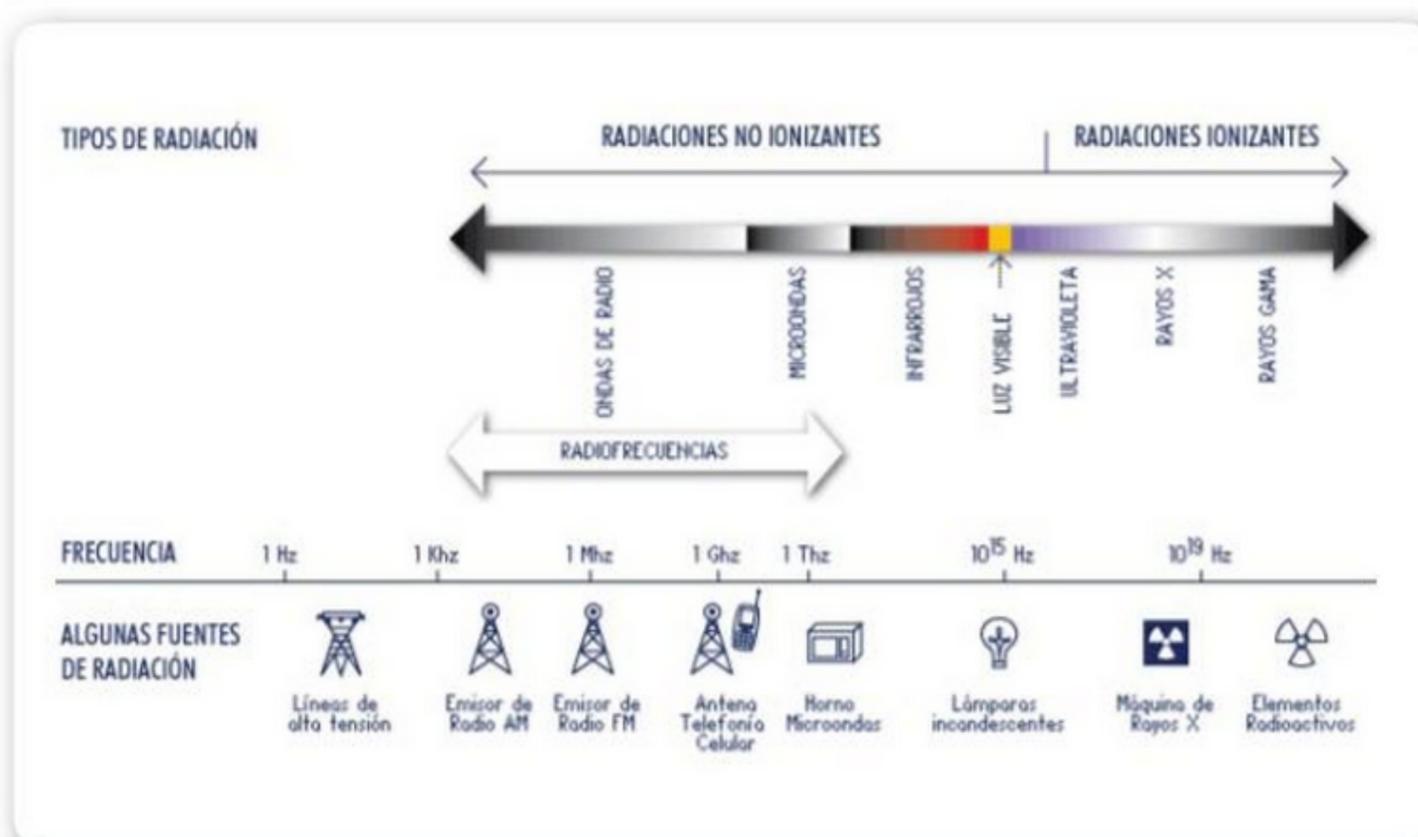


Figura 8. En esta imagen, se pueden apreciar los distintos tipos de radiación y la frecuencia en que irradian. A la vez, se pueden ver ejemplos de las fuentes de radiación a una frecuencia determinada.

Ventajas y desventajas de la RF

Las **señales de RF** pueden transmitirse a grandes distancias, porque sus frecuencias suelen ser muy bajas. A la vez, generan un ahorro en cableado eléctrico y mantenimiento, ya que solo se necesita un emisor y un receptor para la señal, y la cantidad necesaria de repetidores depende de la distancia. Estas señales tienen numerosos usos que van desde radio y televisión hasta telefonía celular.

Como la radiofrecuencia tiene muchas aplicaciones y utilidades, existen demasiadas señales trabajando en las mismas frecuencias, lo que ocasiona la presencia de diferentes tipos de interferencias en la señal que se pretende transmitir.

Además, si la antena que la mayoría de los dispositivos utiliza para recibir la señal transmitida es obstruida o hay demasiada interferencia hasta donde se encuentra la señal, puede que esta no llegue o no se reciba bien.

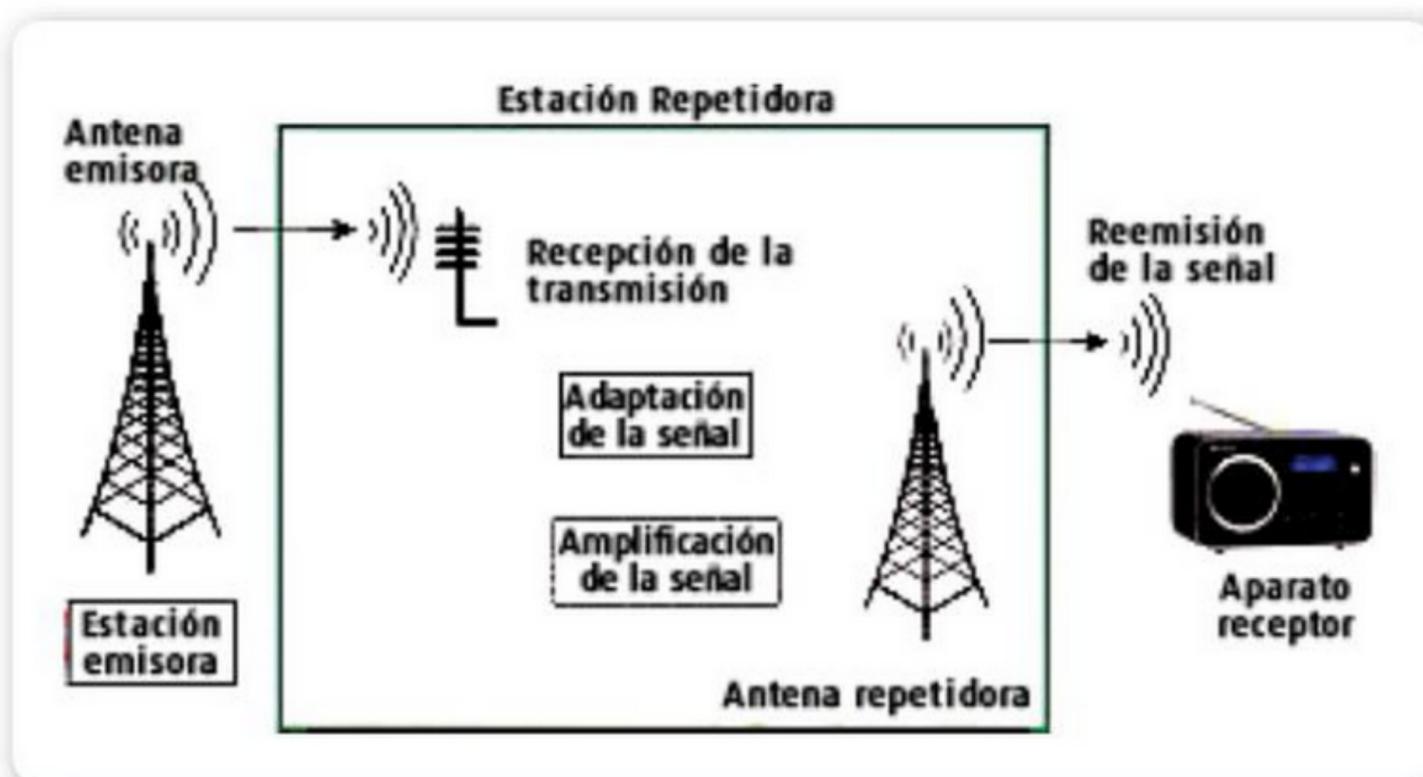


Figura 9. En esta imagen se puede ver, a grandes rasgos, una transmisión de radiofrecuencia en distancia, ya que interviene una antena repetidora.

Usos de la RF

Algunos de los usos de la RF los podemos encontrar en los siguientes campos: en las comunicaciones de televisión, radio, radar, telefonía móvil, servicios de emergencia y navegación marítima; en astronomía, ya que muchos objetos astronómicos emiten en radiofrecuencias definidas o de rangos extensos; en los sistemas de radar, se utilizan para medir distancias, altitudes, direcciones, velocidades, control de tráfico aéreo y terrestre y una gran variedad de usos militares; en la medicina, no solo es aplicable en la parte estética, donde tiene un rol importante, sino en ciertos tratamientos como la resonancia magnética, entre otros; además, en metalurgia se utiliza para el templado de metales o para soldaduras.

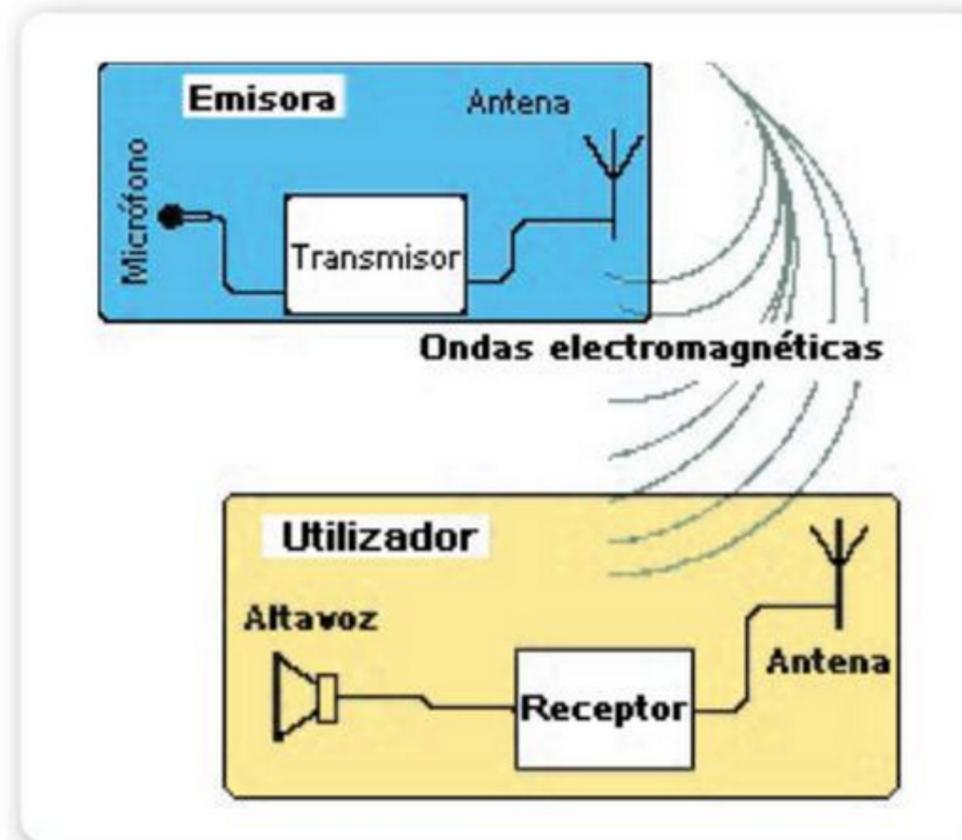


Figura 10. En esta imagen se puede apreciar un sistema básico y genérico de un emisor y un receptor. Este sistema se puede aplicar desde una estación de radio hasta el usuario, o en una comunicación entre dos puntos.



VHF

VHF (*Very High Frequency*) es la banda del espectro electromagnético que ocupa el rango de frecuencias de 30 a 300 MHz. Superados los 50 MHz, encontramos, según los países, las frecuencias

VHF ES LA BANDA
DEL ESPECTRO QUE
OCUPA EL RANGO
DE FRECUENCIAS
DE 30 A 300 MHZ

asignadas a la televisión comercial en el rango de canales bajos (o sea, del canal 2 al 13); entre los 88 y los 108 MHz, están las frecuencias asignadas a las radios FM; entre los 108 y 136.975 MHz, la banda de frecuencia usada en aviación. Los radiofaros utilizan las frecuencias entre 108.7 y 117.9 MHz.

Las comunicaciones por voz se realizan por encima de los 118 MHz e implementan la amplitud modulada. En 137 MHz se encuentran

las señales de satélites meteorológicos. Entre 144 y 146 MHz –e incluso 148 MHz–, están las frecuencias de radioaficionados. Entre 156 y 162 MHz, se encuentra la banda de frecuencias VHF

internacional, que está reservada al servicio radio marítimo. Por encima de esta barrera de frecuencia, hay otros servicios, como bomberos, ambulancias y radiotaxis.

UHF

UHF (*Ultra High Frequency*) es una banda del espectro electromagnético que ocupa el rango de frecuencias de 300 MHz a 3 GHz.

Entre los 470 y 862 MHz, según los países, algunos canales ocupan las frecuencias comprendidas en esta banda de frecuencias. En la actualidad, se utiliza la banda UHF para emitir la televisión digital terrestre (TDT).

Los radioaficionados también cuentan con dos bandas UHF. La primera es la banda de 70 cm, entre los 430 y 440 MHz; como los radioaficionados son considerados de carácter secundario y no prioritario, deben compartir estas frecuencias con otros servicios, como transmisores de baja potencia para la apertura de garajes, repetidoras hogareñas de televisión y dispositivos de comunicación de baja potencia. La segunda banda es la correspondiente a los 1200 MHz. En la telefonía celular, las primeras frecuencias utilizadas fueron de alrededor de 400 MHz. Con la llegada de la norma internacional GSM, las frecuencias UHF se sitúan alrededor de los 900 MHz.

UHF ES UNA BANDA DEL ESPECTRO QUE OCUPA EL RANGO DE FRECUENCIAS DE 300 MHz A 3 GHz



Dispositivos RFID

RFID es la sigla de identificación por radiofrecuencia (*Radio Frequency Identification*). Es un sistema de almacenamiento



ANTENAS



La antena es un dispositivo fabricado sobre un conductor metálico que está diseñado para emitir o recibir ondas electromagnéticas. Una antena transmisora transforma voltajes en ondas electromagnéticas, y una receptora realiza la función inversa. Existen antenas que expanden lo más posible la potencia irradiada y otras que son direccionales para canalizar la potencia en una sola dirección.

y recuperación de datos remotos que usa dispositivos denominados etiquetas, tarjetas o tags RFID. El propósito fundamental de esta tecnología es transmitir la identidad de un objeto, un número de serie único, mediante ondas de radio. Las tecnologías RFID se agrupan dentro de las denominadas **Auto ID** (*Automatic Identification*).

Wi-Fi

Es el nombre que lleva el mecanismo de conexión inalámbrica de dispositivos electrónicos. Es también una marca de la Wi-Fi Alliance, que anteriormente se llamaba **WECA** (**Wireless Ethernet Compatibility**). Esta organización es la que se encarga de adoptar, probar y certificar que todos los equipos cumplan con los estándares 802.11. Esta norma fue diseñada para sustituir a las capas físicas y MAC de la norma 802.3, que es la norma de Ethernet para

WI-FI ES UNA
CONEXIÓN
INALÁMBRICA
DE DISPOSITIVOS
ELECTRÓNICOS

características inalámbricas. Por esta razón, la única diferencia entre una red Ethernet y una Wi-Fi reside en cómo se transmiten las tramas o los paquetes de datos. Esto quiere decir que una red Wi-Fi es compatible con todas las características y servicios que brinda una red local cableada. El término Wi-Fi no es un acrónimo de **Wireless Fidelity**; la WECA contrató una empresa para que diseñara un nombre y un logo para la marca, que fuera fácil de recordar y de identificar. Esta empresa se llama Interbrand y es responsable de crear

nombres como Compaq, Imation, entre otros.



LAS ZONAS DE FRESNEL

La zona de Fresnel es el volumen de espacio entre emisor y receptor RF, de modo que el desfase entre las ondas en dicho volumen no supere los 180°. Cuando transmitimos en forma terrestre, tenemos rebotes en el suelo. Para comunicarnos a una distancia D con una señal portadora de frecuencia f , debemos conseguir que la altura r de la primera zona de Fresnel esté libre de obstáculos o, al menos, el 80 % de r .



Figura 11. En la imagen, vemos el logo que mandó a crear la WECA. Tiene un estilo similar al dibujo del ying y yang, a pedido de esta organización.

Historia

Esta tecnología nació por la necesidad de establecer, de alguna manera, una conexión inalámbrica que fuese compatible entre distintos dispositivos. Debido a esta necesidad fue que, en el año 1999, las empresas Airones, 3com, Intersil, Nokia, Lucent Technologies y Symbol Technologies se reunieron para tratar de lograr este objetivo.

La función de estas empresas fue crear una tecnología que les permitiera fomentar una conexión inalámbrica y, de esta manera, certificar la compatibilidad de equipos. Para mediados de abril de 2000, la WECA certificó la interoperabilidad de los equipos, de acuerdo a la norma IEEE 802.11.

Todo esto con el sello de la marca Wi-Fi, que certifica que un dispositivo puede trabajar sin problemas, de manera independiente del fabricante. En el año 2002, la organización estaba integrada por casi 150 miembros.

Estándares

Como la IEEE 802.11 fue evolucionando, existen varios tipos de Wi-Fi aprobados en el mercado. Los estándares correspondientes a **IEEE 802.11b, g y n** poseen una aprobación mundial debido a la banda de frecuencia en la que trabajan. Esta banda es de 2.4 GHz y, por lo general,

está siempre disponible con una velocidad promedio de 11 Mbit/s para la norma b, 54 Mbit/s para la norma g y 300 Mbit/s para la norma n.

En la actualidad, ya se está manejando un nuevo estándar denominado IEEE 802.11a, conocido como **Wi-Fi 5**, que opera en la banda de los 5 GHz, a diferencia de los otros estándares. La ventaja que presenta es que, como prácticamente a esa frecuencia no hay emisiones, no hay distorsión en la señal. La desventaja es que tiene un alcance menor, debido a que, a mayor frecuencia, menor es el alcance. Esta disminución es de alrededor del 10 %.

Seguridad y fiabilidad

Como sabemos, uno de los principales problemas que se presenta en la actualidad de la tecnología 802.11 es la creciente saturación del espectro en el que trabaja, o sea, el radioeléctrico. Esto se debe a que cada vez son más los usuarios que utilizan esta tecnología y a que ya viene incluida de fábrica en los nuevos productos. Cabe destacar que esta tecnología fue diseñada para conectar dispositivos a la red a una distancia reducida, por lo que cualquier uso que supere los 100 m estará expuesto a un riesgo excesivo debido a la interferencia.

Por otro lado, cuando se instala una red inalámbrica, muchas veces no se tiene en cuenta su seguridad, ya que la configuración por defecto es la de una red abierta. Este tipo de red no protege la información circulante, por lo que es vulnerable al acceso no autorizado de terceros, lo que puede ser muy perjudicial para el propietario de los contenidos. Una vez que se accede a una red, es posible monitorearla, registrar toda la información que circula por ella y, a su vez, se pierde el ancho de banda correspondiente al que utiliza la persona ajena a la red. Por esta razón, para lograr



LOS DISPOSITIVOS



En el mercado existen dos grandes grupos de dispositivos Wi-Fi: los de distribución o de red y los dispositivos terminales. Los primeros están compuestos por puntos de acceso, repetidores de señal y routers. Los segundos están integrados por placas PCI, tarjetas PCMCIA y USB Wi-Fi. Los puntos de acceso son los que generan la red, mientras que los repetidores extienden la cobertura de la señal Wi-Fi.

que la red sea segura, debemos tener en cuenta ciertos consejos y configuraciones. Lo primero que tenemos que hacer es modificar el SSID (nombre de la red) que viene predeterminado.

Luego, hay que cambiar el cifrado WEP a uno WPA2, desactivar el broadcasting SSID y el DHCP. Por último, indicamos qué dispositivos están autorizados a conectarse. Esto se hace otorgando una IP fija a cada dispositivo, o mediante la identificación de cada dirección MAC. Es fundamental que, una vez realizadas todas las configuraciones para asegurar la seguridad de la red, modifiquemos la clave de acceso al router que viene por defecto. Dos consejos muy útiles son: por un lado, el cambio frecuente de la contraseña de acceso a la red y, por el otro, utilizar diversos caracteres, como minúsculas, mayúsculas y números.



Figura 12. Necesitamos establecer una configuración básica de conexión de un router con respecto a los demás dispositivos a los que se puede vincular, tanto de manera inalámbrica como cableada.



VENTAJAS Y DESVENTAJAS DE UNA RED WI-FI



Por un lado, las redes Wi-Fi ofrecen una mayor comodidad, ya que la conexión se puede realizar desde cualquier punto dentro del radio de cobertura y no producen gasto en infraestructura. A la vez, aseguran la compatibilidad entre dispositivos de cualquier parte del mundo. Sin embargo, son menos veloces que una red cableada porque sufren pérdidas de señal e interferencias.



Figura 13. Aquí vemos uno de los tantos dispositivos que, hoy en día, se pueden vincular mediante una conexión inalámbrica provista por un router.



WiMAX

WiMAX es la sigla de *Worldwide Interoperability for Microwave Access* (interoperabilidad mundial para acceso por microondas). Se trata de una norma de transmisión de datos, donde el medio está constituido por las ondas de radio a una frecuencia que va desde los 2.3 a 3.5 GHz, con una cobertura aproximada de unos 60 a 70 km.

WIMAX SIGNIFICA
INTEROPERABILIDAD
MUNDIAL PARA
ACCESO POR
MICROONDAS

El estándar inalámbrico metropolitano WiMAX es el IEEE 802.16, creado en el año 2002 por Intel y Alvarion para proporcionar un acceso a internet de alta velocidad con una cobertura mucho mayor que Wi-Fi, aproximadamente del orden de los 70 km.

Se estipula que esta nueva red puede proporcionar velocidades de 70 Mbits/s en un rango aproximado de 50 km, pero solo puede traspasar obstáculos pequeños, como, por ejemplo, una casa o un árbol, y no edificios altos o montañas.

Cuando la señal se topa con ciertos obstáculos, su velocidad puede llegar a ser inferior a los 20 Mbits/s.

Funcionamiento

En este tipo de tecnología, lo más importante es el transceptor de la estación base. Este transceptor es una antena central que permite la comunicación con las demás antenas a las que se puede vincular. Utiliza un sistema de comunicación denominado enlace punto a multipunto.

Presenta características muy específicas, ya que puede cubrir, de manera fácil, una distancia de aproximadamente 80 km, con la ayuda de antenas direccionales y de alta ganancia. Posee velocidades de hasta 70 Mbits/s cuando el canal no presenta saturación, ya que, de lo contrario, la velocidad se ve reducida.

Tiene facilidad para extender sus canales en funcionamiento, o sea, es posible agregar más canales de acuerdo con las regulaciones del país en donde se utilice la tecnología. WiMAX permite configurar los anchos de banda con respecto a la relación del espectro. A la vez, se puede dividir el canal por el que se producen las comunicaciones en pequeñas subportadoras. Esto quiere decir que hay dos tipos de guardas y datos.

Este estándar de transmisión inalámbrica se clasifica según dos variantes: la de acceso fijo y la de acceso de movilidad completa. La primera, denominada 802.16d, establece una conexión por radio entre la estación base y el equipo que utiliza el usuario situado en un lugar fijo como, por ejemplo, el espacio de trabajo. En teoría, las velocidades de esta variante rondan los 70 Mbits/s, con un ancho de banda de 20 MHz, pero en la práctica estos valores cambian drásticamente. Se consiguen velocidades de 20 Mbits/s en un radio cercano a los 6 km.

La segunda, denominada 802.16e, permite el desplazamiento del usuario de una manera similar a las conexiones 3G o GSM/UMTS, y está tratando de ganarse un lugar en la vida cotidiana del usuario. Sin embargo, tiene que competir con las tecnologías móviles, como la LTE, que están basadas en fotocélulas conectadas a través de un cable. A su vez, tiene que enfrentarse con el estándar Wi-Fi 802.11n, debido a que la mayoría de los dispositivos con posibilidad de conexión inalámbrica están certificados para el uso de este estándar.

EL ESTÁNDAR
INALÁMBRICO
METROPOLITANO
WIMAX ES
EL IEE 802.16





Comunicaciones a larga distancia

Las **microondas** son aquellas ondas electromagnéticas cuyas frecuencias van desde los 500 MHz hasta los 300 GHz o cifras mayores. Por lo tanto, las señales de microondas, a causa de sus altas frecuencias, tienen longitudes de onda relativamente pequeñas. Las longitudes de las frecuencias de microondas van de 1 a 60 cm; son un poco mayores que en la energía infrarroja.

Las microondas terrestres (*Radio Relay System*) proveen conectividad entre dos sitios (estaciones terrenas) con punto de visión (*Line of Sight*, **LOS**), que usan un equipo de radio con frecuencias portadoras por encima de 1 GHz. La forma de onda emitida puede ser analógica (FM) o digital.



Figura 14. Las antenas repetidoras de microondas se instalan en cualquier sitio que garantice a la compañía una buena cobertura.

La antena más comúnmente utilizada es la de tipo parabólico. El tamaño típico es de un diámetro de 3 metros, según la longitud de onda. Esta antena se fija de manera rígida y, en este caso, el haz estrecho debe estar perfectamente enfocado hacia la antena receptora.

Las antenas de microondas se sitúan a una altura apreciable sobre el nivel del suelo para conseguir mayores separaciones entre ellas y evitar posibles obstáculos en la transmisión.

Para llevar a cabo transmisiones a largas distancias, se utiliza la concatenación de enlaces punto a punto entre antenas situadas en torres adyacentes, hasta cubrir la distancia necesaria.

El uso principal de los sistemas de microondas terrestres son los servicios de telecomunicación de larga distancia, como alternativa al cable coaxial o a las fibras ópticas. Para una distancia dada, las microondas requieren menor número de repetidores o amplificadores que el cable coaxial, pero necesitan que las antenas estén perfectamente alineadas. El uso de las microondas es común en la transmisión de televisión y voz, en la telefonía básica (canales telefónicos) y en la telefonía celular (entre troncales).

PARA TRANSMITIR A
LARGA DISTANCIA
SE CONCATENAN
LOS ENLACES DE
PUNTO A PUNTO



Otro uso frecuente es en enlaces punto a punto a cortas distancias entre edificios. En este caso, las aplicaciones típicas son los circuitos cerrados de televisión o la interconexión de redes locales. Además, las microondas a corta distancia se utilizan en las aplicaciones de bypass, con las que una determinada compañía puede establecer un enlace privado hasta el centro proveedor de transmisiones a larga distancia, evitando contratar el servicio a la compañía de telefónica local.

El rango de las microondas cubre una parte sustancial del espectro electromagnético. La banda de frecuencias está comprendida entre 2 y 40 GHz. Cuanto mayor es la frecuencia utilizada, mayor es el ancho de banda potencial y, por lo tanto, mayor es la posible velocidad de transmisión.

Existen dos formas para transmitir por microondas terrestres: en la primera, ocupamos una porción de banda de manera exclusiva; en la segunda, podemos compartirla:



LA PARTE LEGAL



Las licencias o permisos para operar enlaces de microondas pueden resultar difíciles de obtener, ya que hay que asegurarse que ambos enlaces no causen interferencia a los enlaces ya existentes. El clima y el terreno son los mayores factores para considerar antes de instalar un sistema de microondas.

Las montañas o grandes masas de agua pueden ocasionar reflexiones de multitrayectorias.



Figura 15. Los arreglos de antenas telefónicas se caracterizan por ser bidireccionales (emisión y recepción) de baja potencia.

- **Enlace con asignación de banda** : es la forma más directa de transmitir por radiofrecuencia y la más usada durante mucho tiempo para las aplicaciones punto a punto de cierta distancia. Habitualmente, esta técnica de transmisión con banda regulada se utiliza cuando las transmisiones son PaP o PMP, cuando la potencia debe ser mayor que un vatio (1 Watt) o cuando la aplicación requiere frecuencias altas. Ejemplos de estas transmisiones son los enlaces punto a punto de alta capacidad y el acceso inalámbrico a internet con tecnología MMDS o LMDS.
- **Enlace con espectro ensanchado** : esta técnica tiene una alta inmunidad al ruido, comparada con la transmisión por radio convencional. Muchos de los usuarios pueden compartir la misma frecuencia. Las reglas de transmisión fueron diseñadas para la implementación de comunicaciones de datos locales o de alcances ilimitados. Entre las ventajas de este sistema de comunicación por microondas terrestres, podemos destacar la poca interferencia debido a su amplio espectro y su adecuación a distancias cortas. Cabe mencionar que, a frecuencias superiores, las antenas son más pequeñas y, por ende, más baratas.

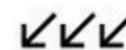


Figura 16. Una antena convierte potencia de RF (radiofrecuencia) en campos electromagnéticos.

Figura 17. Antena de uso doméstico (TV satelital), empleada en este caso para un enlace punto a punto de microondas terrestres.



¿CÓMO ESTÁ COMPUESTO?



Un sistema de microondas está formado por tres componentes principales: una antena con una guía de onda corta y flexible, una unidad externa de RF (radiofrecuencia) y una unidad interna de RF. Las principales frecuencias utilizadas en microondas se encuentran alrededor de los 12, 18 y 23 GHz. El equipo de microondas que opera entre 2 y 6 GHz transmite a distancias entre 20 y 30 millas.



Sistemas satelitales

Las **microondas satelitales**, a diferencia de las terrestres, retransmiten información. El satélite básicamente actúa como una estación de relevación o repetidor, amplificando, corrigiendo y retransmitiendo la señal a una o más antenas ubicadas en la Tierra.

Una instalación básica de red por satélite requiere un dispositivo de conectividad o un controlador de antena que se conecte a una antena parabólica (se suele denominar plato de satélite) de 0.75 a 2.4 m a través de un medio de cable. La antena refleja las señales generadas por una baliza a un satélite en órbita geoestacionaria a 22300 millas de la Tierra. A continuación, estas señales se vuelven a apuntar a una estación maestra (MES) de la Tierra a una ubicación de concentrador.



Figura 18. Radiotelescopio satelital que recibe información de los satélites y sondas orbitales alrededor de la Tierra.

Por último, las señales se reciben en la MES o en otra antena de la red y se suministran a la red adecuada mediante el dispositivo de conectividad de red adecuado. Se pueden configurar sistemas sencillos punto a punto y sistema múltiples transmisor/receptor.

Un **transpondedor** (*transponder*) recibe señal de un transmisor, luego la amplifica y la retransmite a la Tierra a una frecuencia diferente. Cabe destacar que la estación terrena transmisora envía señal a un solo satélite, pero el satélite envía señal a cualquiera de las estaciones terrenas receptoras en su área de cobertura.

Las transmisiones de microondas por satélite requieren aproximadamente el mismo tiempo para cruzar continentes u océanos que para cruzar algunos kilómetros; esto se debe a que, sin importar la distancia entre las antenas, la señal siempre debe ir hasta el satélite retransmisor.

Debido a la gran distancia que debe recorrer la señal, las transmisiones por satélite sufren una demora bastante larga (demora de propagación). Esta demora oscila entre 500 milisegundos y más de 5 segundos; es notoria en comunicaciones de voz, como la telefonía celular de larga distancia (intercontinental). Sin embargo, pueden proporcionar una señal al lugar más lejano y subdesarrollado de la Tierra, como, por ejemplo, las bases navales y las bases antárticas.

EL SATÉLITE ACTÚA
AMPLIFICANDO,
CORRIGIENDO Y
RETRANSMITIENDO
LA SEÑAL



Figura 19. Estación satelital terrestre encargada de monitorear distintos satélites gubernamentales.

Los **satélites geoestacionarios** (*Geosynchronous Earth Orbit* , **GEO**) también operan en una serie de frecuencias llamadas transponders. Es importante que los satélites se mantengan en una órbita geoestacionaria, ya que, de lo contrario, perderían su alineación con respecto a las antenas ubicadas en la Tierra.

Un satélite es geoestacionario cuando permanece inmóvil sobre un determinado punto de la Tierra. Para obtener este efecto es

necesaria que la órbita del satélite se encuentre sobre el plano del Ecuador terrestre y que el satélite realice una vuelta alrededor de nuestro planeta al mismo tiempo que efectúa una rotación completa sobre su propio eje. Las órbitas geoestacionarias son útiles para los satélites de telecomunicaciones: puede actuar de puente radio para comunicaciones telefónicas, o la difusión mundial de señal de televisión, entre otras.

Por otro lado, los **satélites de órbita baja** (*Low Earth Orbit, LEO*) ofrecen otras alternativas. Giran alrededor de la tierra a más de 2000 millas, proveen comunicaciones de datos a baja velocidad y, aunque son incapaces de manipular voz, señales de video o datos a altas velocidades, tienen ventajas que los satélites GEO no tienen. Por ejemplo, no existe retardo en las transmisiones, son menos sensibles a factores atmosféricos y transmiten a muy poca potencia. Estos satélites operan a frecuencias asignadas entre 1545 y 1645 GHz (Banda L). La transmisión satelital puede ser usada para proporcionar una comunicación punto a punto entre dos antenas terrestres alejadas entre sí, o bien para conectar una estación base transmisora. Para esto, es necesario que los satélites estén separados en, por lo menos, tres grados (desplazamiento angular medio desde la superficie terrestre) en la banda 6/4 GHz y al menos 4 grados en la banda 14/12 GHz. Por lo tanto, el número máximo de satélites posibles es bastante limitado.



Figura 20. Antena satelital de uso doméstico, normalmente empleada para la recepción de televisión satelital.



Figura 21. Vista desde el espacio de un satélite geoestacionario. Las grandes extensiones de paneles solares permiten la recarga de sus baterías.

Las comunicaciones satelitales tienen numerosas aplicaciones, entre las que podemos mencionar: difusión de televisión, transmisión telefónica a largas distancias y en redes privadas, entre otras. Uno de los usos más conocidos que se le ha dado a los satélites es la difusión directa vía satélite (*Direct Broadcast Satellite, DBS*), en la que la señal de video se transmite directamente del satélite a los domicilios de los usuarios.

Esto se logra mediante la implantación de un antena de bajo costo en el domicilio de cada usuario, logrando así que la cantidad de canales ofrecidos aumente de manera notable. El satélite funciona como una estación repetidora que recoge la señal de algún



HISTORIA

La idea de comunicación satelital fue materializada por Arthur Clarke, en el año 1945, quien propuso una serie de puntos importantes para los cuales serían utilizados los satélites, por ejemplo, como repetidores de comunicaciones. Casi todos los puntos propuestos fueron llevados a cabo muchos años después, con la evolución de la tecnología de cohetes.

transmisor en Tierra y la retransmite, difundiéndola entre una o varias estaciones terrestres receptoras. El satélite puede regenerar dicha señal, o bien, limitarse a repetirla.

Ventajas y desventajas

La transmisión por satélite ofrece muchas ventajas en el sentido corporativo, ya que los precios de renta de espacio satelital son más estables que los que ofrecen las compañías telefónicas, a raíz de

LAS TRANSMISIONES
POR SATÉLITE SON
MUY VENTAJOSAS
PARA LAS
CORPORACIONES

que la transmisión por satélite no es sensitiva a la distancia y, además, existe un gran ancho de banda disponible. Entre los beneficios ya mencionados, destacamos la transferencia de información a altas velocidades (Mbits/s, Gbits/s) y que es ideal para comunicaciones entre puntos distantes y no fácilmente accesibles.

Como desventajas, cabe destacar que la comunicación por satélite puede sufrir retardos (250-500 milisegundos de propagación), es sensible a efectos atmosféricos, erupciones

solares y eclipses, y requiere una alta precisión en la alineación de las antenas, tanto receptoras como transmisoras, con el satélite; la desviación no puede ser mayor a 3 grados.



Figura 22. Satélite geostacionario GOES, que brinda información climática a gran parte de Latinoamérica. Actualmente, existen 13.

Satélites GEO

El primer satélite geoestacionario fue el conocido con la sigla **Syncom 3**, y se lanzó en agosto de 1964. En realidad, a causa de las influencias gravitacionales de la Luna y del Sol, el satélite no se queda exactamente fijo en un punto geográfico sobre la Tierra, sino que tiende a desplazarse. Para volver a la posición deseada, está provisto de pequeños motores que realizan las maniobras de corrección de posición.



RESUMEN



En este capítulo, recorrimos los fundamentos de la comunicación sin cables: su historia, la explicación del espectro de frecuencias, la necesidad del uso de antenas y sus principios de funcionamiento, los usos principales y cómo todo esto ha transformado de manera radical la manera de comunicarnos en la actualidad. La tecnología electrónica ha desarrollado, asimismo, distintas tecnologías y sistemas para interconectar dispositivos, que dependen en parte de la distancia a la que estos se encuentran. En cuanto a la media distancia, vimos los sistemas tradicionales de radiofrecuencia, tanto de alta frecuencia como de muy alta frecuencia (VHF) y ultraalta frecuencia (UHF). Estudiamos la tecnología Wi-Fi –habitualmente utilizada en informática y ya por completo estandarizada– y, también, el último de los intentos de establecer comunicaciones eficientes en áreas no demasiado grandes: WiMAX. Por último, conocimos las comunicaciones en largas distancias, entre las que analizamos los sistemas de microondas terrestres y satelitales.

Actividades

TEST DE AUTOEVALUACIÓN

- 1 Describa el origen de la **tecnología inalámbrica** .
- 2 ¿Cuáles son las características principales de la **tecnología Wi-Fi**?
- 3 ¿Qué son las **antenas** y qué función cumplen?
- 4 ¿Cuáles son las características de la **comunicación infrarroja** y **Bluetooth** ?
- 5 ¿Qué es la **radiofrecuencia** y cómo se compara con los **sistemas satelitales**?



PROFESOR EN LÍNEA



Si tiene alguna consulta técnica relacionada con el contenido, puede contactarse con nuestros expertos: profesor@redusers.com

Fuentes de alimentación

Conoceremos aquí las fuentes de alimentación, las distintas etapas del circuito, los dispositivos reguladores de voltaje y cómo armar una fuente completa. Además, la generación alternativa de energía eléctrica con sistemas de alimentación ininterrumpida y los generadores eléctricos a combustible.

▼ Fuentes lineales y conmutadas.....256	▼ Reguladores de voltaje285
▼ Fuentes lineales260	▼ Generación alternativa de 220 V291
▼ Fuentes conmutadas269	▼ Generadores a combustible ...296
Las tres configuraciones básicas 272	▼ Resumen.....299
Diseño online de una fuente conmutada..... 273	▼ Actividades.....300
▼ Diseño y reguladores279	



Fuentes lineales y conmutadas

Si bien en la mayoría de los países nos encontramos con redes domiciliarias que alimentan de energía eléctrica alterna a los usuarios, en un número reducido de países todavía se utilizan tensiones continuas, a pesar de la mayor complejidad y eficiencia en la generación, transmisión y conversión de la energía eléctrica. En algunos países, se utiliza una tensión alterna de 220 V y 50 Hz, mientras que en otros se utiliza una de 110 V y 60 Hz.

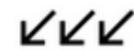
En electrónica, la mayoría de los dispositivos necesita una tensión continua para funcionar, y la fuente de alimentación es la que cumple específicamente la función de conversión. Una fuente de alimentación es un dispositivo electrónico que convierte la tensión de red alterna (CA) en tensión continua (CC).

Clasificamos las fuentes de alimentación en **fuentes no reguladas** y en **fuentes reguladas**. Las fuentes de alimentación reguladas pueden ser **reguladas linealmente** o **conmutadas**. Las fuentes de alimentación conmutadas pueden ser **conmutadas en primario** o **conmutadas en secundario**.

Una fuente lineal responde al esquema transformador, rectificador, filtro, regulación y salida; tiene un diseño relativamente simple, que se complejiza a medida que aumenta la capacidad de la corriente de salida, pero se caracteriza porque la regulación de tensión es poco eficiente en comparación con una fuente conmutada. Una fuente conmutada (en inglés, *Switching*) de la misma potencia que una fuente lineal es más pequeña y, en general, más eficiente, aunque más compleja y, como consecuencia, más susceptible a fallas. Se



BATERÍAS Y PORTABILIDAD



En una red de sensores inalámbricos, la vida de la batería depende de los requerimientos de diseño de la red, de las características de los módulos y del protocolo utilizado (por ejemplo, ZigBee u otro). Podemos hacer un análisis de la vida de la batería con las herramientas disponibles en www.learnzigbee.com/Calculators.html.

caracteriza porque la transformación de la energía se realiza mediante transistores de conmutación (por ejemplo, transistores bipolares y MOSFETs) que funcionan como interruptores. Por ejemplo, una fuente de computadora es una fuente conmutada.



Figura 1. Es importante disponer de una fuente de alimentación de laboratorio que pueda funcionar como fuente de corriente o de tensión.



Figura 2. El multímetro es un instrumento de medición fundamental en electrónica para tensiones o corrientes alternas o continuas.

LA FUENTE DE
ALIMENTACIÓN
CONVIERTE LA
TENSIÓN EN LA
CONTINUA ADECUADA



En una fuente de alimentación, nos interesa la tensión y la corriente de salida, el **rendimiento**, el cociente de la potencia total de salida y el valor de la potencia activa de entrada. El **factor de potencia** es el cociente entre la potencia activa y la potencia aparente de entrada, que utilizamos para medir la calidad de la energía eléctrica utilizada. La **regulación** es la habilidad para mantener constante la tensión de salida, independientemente de la variación en la tensión de entrada (regulación de entrada) o de la carga (regulación de carga).

Ahora bien, la seguridad del personal y del equipo electrónico en las instalaciones es una prioridad, incluso en las fuentes de alimentación. En una fuente de alimentación electrónica, debemos prever protecciones en el caso de **sobrecarga** o de **cortocircuitos**. Si el circuito es sencillo, utilizamos un **fusible** eléctrico conformado por un filamento de metal de muy bajo punto de fusión dentro de un tubo de vidrio intercalado en un circuito eléctrico, de modo de fundir, por **Efecto Joule**, el filamento ante un gran aumento en la intensidad de la corriente, sea por un cortocircuito o por exceso de carga.

Un **varistor** nos brinda una protección adicional confiable y económica contra alto voltaje producido por descargas eléctricas atmosféricas, conmutaciones o ruido eléctrico en líneas de potencia de CA o CC. Ante esto, el varistor cambia su resistencia de un valor alto a otro valor muy bajo, permitiendo que circule corriente a través de él y protegiendo los componentes sensibles del circuito. En el sitio web www.gmelectronica.com.ar/catalogo/pag38-39.html podemos seleccionar varistores.

En circuitos electrónicos complejos, esta configuración se mejora mediante un **circuito de protección** específico dentro de la fuente, que funciona bajo el principio de **realimentación** (en inglés, **feedback**). Así, se activa el circuito de protección cuando la corriente excede el máximo valor permitido. Si esta situación ocurre, tanto la tensión como la corriente de salida de la fuente se reducen prácticamente a cero como consecuencia de la acción del circuito de protección, con lo que también se reduce la potencia disipada y la temperatura, tanto del elemento regulador como de sus componentes asociados.



Figura 3. A partir de una fuente de alimentación tipo **ATX** para PC podemos construir una fuente de alimentación de laboratorio más económica.

Las baterías son dispositivos que emplean procedimientos electroquímicos para almacenar energía eléctrica en forma química. Al conectar un circuito eléctrico, la energía química se convierte en eléctrica. Todas las baterías están compuestas por un número determinado de celdas, constituidas en su interior por un electrodo positivo (**ánodo**) y uno negativo (**cátodo**).

Las baterías de plomo (**Pb**) son las más comunes y las utilizamos en los automóviles. Producen energía a partir de que el plomo se sumerge en ácido sulfúrico, que actúa como electrolito. Proporcionan un voltaje de 2 V por elemento y pueden ser recargadas mediante un proceso lento. Su característica más importante es la gran capacidad de entregar corriente, lo que



FUNCIÓN DEL TRANSFORMADOR



Un **transformador** aísla eléctricamente la tensión de la red de la fuente y reduce la tensión alterna a un valor aceptable para el circuito rectificador. Su funcionamiento se basa en la inducción electromagnética. Está formado por un núcleo de hierro sobre el que se devana tanto el bobinado primario como el secundario (uno o varios), esquema conocido como transformador **E-I**.

SI LA CORRIENTE
ES MENOR QUE
LA REQUERIDA,
SOBRECARGAMOS
LA FUENTE



las vuelve ideales para utilizar en motores de arranque. Cada elemento contiene ácido sulfúrico y una serie de placas de plomo dispuestas alternadamente entre positivas y negativas, separadas por un aislante resistente al ácido. El número de placas y su espesor determinan la cantidad de corriente que entrega cada elemento. Debajo de las placas hay un espacio en el cual se depositan los desprendimientos.

Las baterías de níquel-cadmio (**Ni-Cd**) son las pilas recargables más comunes y proporcionan 1.2 V por elemento. Contienen en su interior un cadmio negativo y el hidróxido de níquel positivo, separados por un electrolito de hidróxido de potasa. Más allá del peligro del cadmio, desde el punto de vista ambiental, las pilas presentan como inconveniente el efecto memoria, por lo que se hace necesario agotarlas en su totalidad antes de recargarlas.

Por último, las baterías de níquel-hidruro metálico (**Ni-MH**) tienen un 40 % más de rendimiento que las de níquel-cadmio a igual cantidad de mA·H (miliamperio/hora). No poseen efecto memoria y admiten alrededor de 700 ciclos de carga.

Fuentes lineales

En una **fuentes de alimentación no regulada** , la tensión de red aplicada a la entrada del circuito se transforma en un valor menor y se rectifica. Por último, un condensador suaviza la tensión de salida del rectificador.

Como consecuencia del diseño del circuito electrónico, la tensión de salida depende directamente de la tensión de entrada, lo que implica que las variaciones en la tensión de red impactan directamente en la salida. Como no se regula en el secundario del transformador, la tensión de rizado es del orden de los voltios y la especificamos como un porcentaje de la tensión continua de salida. Es un diseño sencillo, razón por la cual lo utilizamos en aplicaciones donde no requiera una tensión de salida regulada.

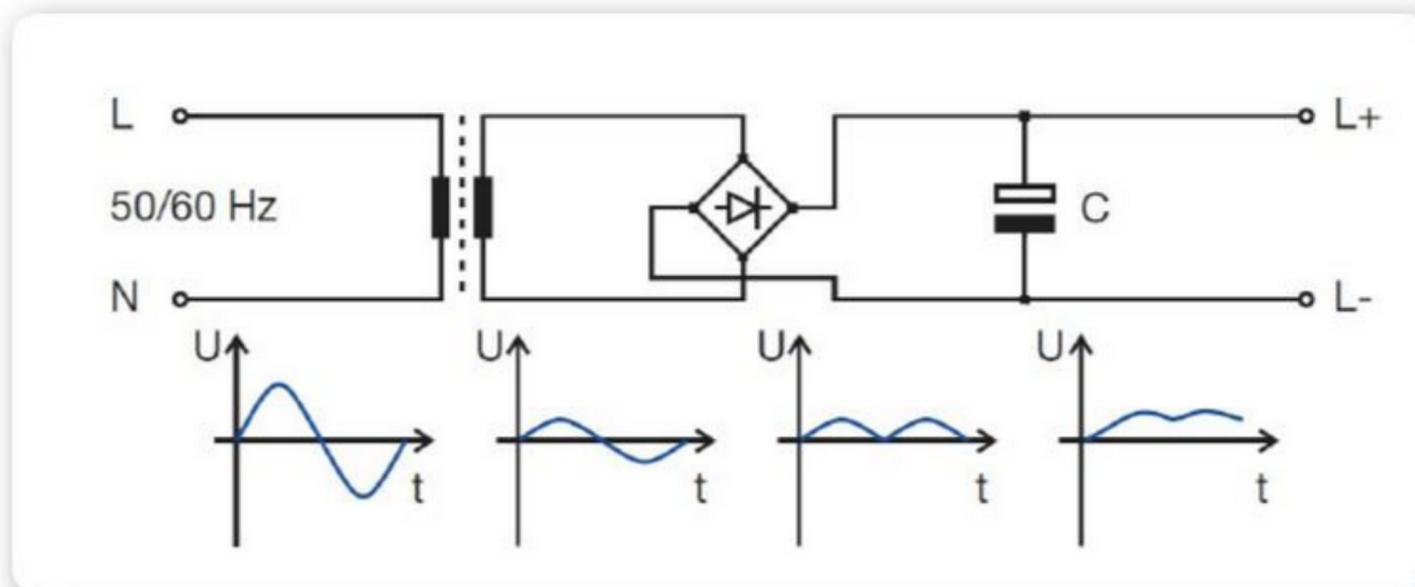


Figura 4. Una fuente de alimentación no regulada se identifica por la amplitud del ripple a la salida y la sencillez del esquema electrónico.

En una **fuentes de alimentación regulada linealmente**, mediante un transformador reducimos la tensión alterna de entrada a un voltaje alterno menor, se rectifica por medio de diodos y se alisa recurriendo a un condensador. A continuación, se regula la tensión, normalmente mediante un transistor de potencia, que actúa como una resistencia variable controlada para mantener la tensión de salida constante dentro de un rango determinado de la tensión de entrada.

Debido a las elevadas pérdidas en el transistor de potencia, el rendimiento de las fuentes de alimentación reguladas linealmente alcanza el 50 % aproximadamente mientras que la energía eléctrica restante se transforma en calor. Así, es indispensable disipar el calor lo suficiente como para mantener una temperatura adecuada en la fuente de alimentación.

EL PUNTO MEDIO Y EL EXTREMO LIBRE DE LA RESISTENCIA ESTÁN CONECTADOS A UN PUNTO COMÚN

”



ESPECIFICACIONES EN UN DIODO

Un diodo tiene especificaciones que lo caracterizan y que tenemos que conocer. Los tres valores más importantes son: **tensión inversa repetitiva de pico (VRRM)**, 50 V para un diodo 1N4001, **tensión máxima aplicable desde el rectificador cuando el diodo está polarizado en inverso** y la **corriente máxima con polarización directa (I0)**, 1 A en el 1N4001.

Si comparamos esta fuente con la fuente no regulada, la fuente regulada linealmente presenta un ripple menor, del orden de los milivolts. Por esta razón, empleamos las fuentes de alimentación con regulación lineal cuando se requiere mantener una tensión de salida exacta, por ejemplo, en dispositivos biomecátrónicos de alta precisión, osciladores, entre otros.

Podemos regular la tensión de salida de una fuente lineal mediante un **regulador**. Por medio de él, compensamos las posibles variaciones de la tensión de salida por cambios en la tensión no regulada y en la corriente de salida por variaciones en la resistencia de carga mediante variaciones de la caída de tensión de un elemento conectado en serie con la carga, **regulador serie**, o en paralelo con ella, **regulador paralelo**. Desde el punto de vista práctico, el regulador serie es el que más utilizamos. A pesar de que su principal desventaja es la disipación de potencia en el transistor en serie con la carga ($P_D = V_{CE} * I_C$), dejamos los reguladores en paralelo para circuitos que demandan corrientes bajas.

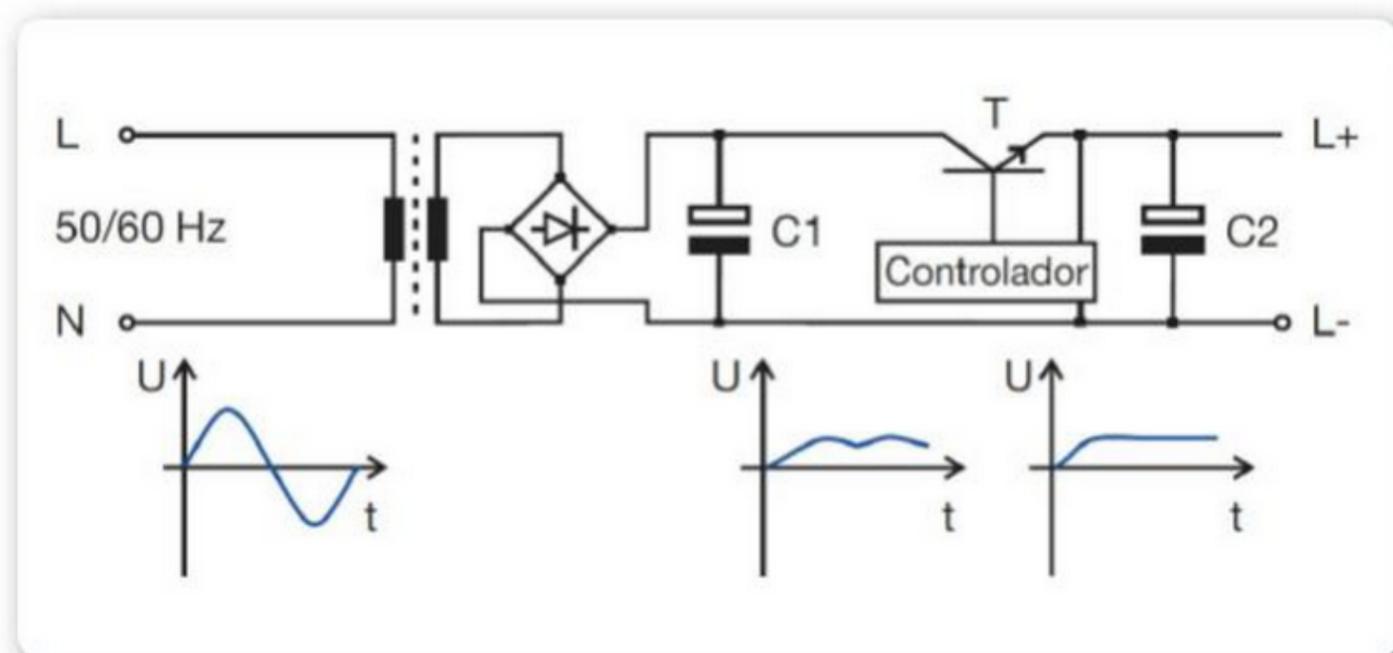


Figura 5. En una fuente de alimentación regulada en serie, un transistor conectado ejerce la acción de regular ajustando su resistencia interna.

Mientras la corriente de colector, o corriente de carga, no sea muy grande, tampoco lo será la potencia disipada en el transistor de salida. Pero cuando la corriente de carga es grande en el transistor de salida, tenemos que utilizar disipadores grandes y voluminosos. De acuerdo con la aplicación, podría ser más eficiente valernos de un regulador conmutado si la corriente en la carga fuera muy grande.

En una fuente de tensión lineal, la tensión de línea se aísla eléctricamente de la propia fuente mediante un transformador e ingresa a una etapa rectificadora cuya finalidad es transformar la onda alterna en una onda continua pulsante. Luego se produce el filtrado que transforma la onda pulsante en onda continua, disminuyendo los valores de ripple a un valor pequeño. Si el filtrado se realiza mediante condensadores, elegimos capacitores electrolíticos (microfaradios).

Para regular la tensión de salida de la fuente, incorporamos una etapa de regulación entre el filtrado y la carga. La etapa reguladora es un circuito electrónico que incorpora realimentación y permite comparar el valor de la tensión de salida con una tensión de referencia. En función de la diferencia entre ambas tensiones, se ejerce la acción de regulación, para que la tensión de salida permanezca dentro de los límites prefijados. Una fuente de alimentación podría requerir múltiples salidas con diferentes valores de tensión (fija o variable) y corriente.

La **etapa de rectificación** está constituida por uno o más diodos rectificadores, cuya función es rectificar y, así, transformar la tensión alterna senoidal (valor medio igual a cero) que proviene del secundario del transformador en una tensión continua pulsante (valor medio distinto de cero).

Se utilizan diodos, ya que tienen la característica de ofrecer una resistencia baja en un sentido de paso de la corriente y una resistencia grande en sentido contrario.

UN PUENTE
RECTIFICADOR DE
DIODOS INTEGRADO
TIENE CUATRO
DIODOS DENTRO DE ÉL



ENERGÍA ININTERRUMPIBLE



Se trata de la cantidad de minutos que el UPS será capaz de brindar energía de la batería para un nivel dado de carga. Este tiempo es muy importante porque nos dice cuánto tiempo continuará operando el equipo durante un apagón. El agregar más equipo al UPS resultará en un tiempo de respaldo más corto.

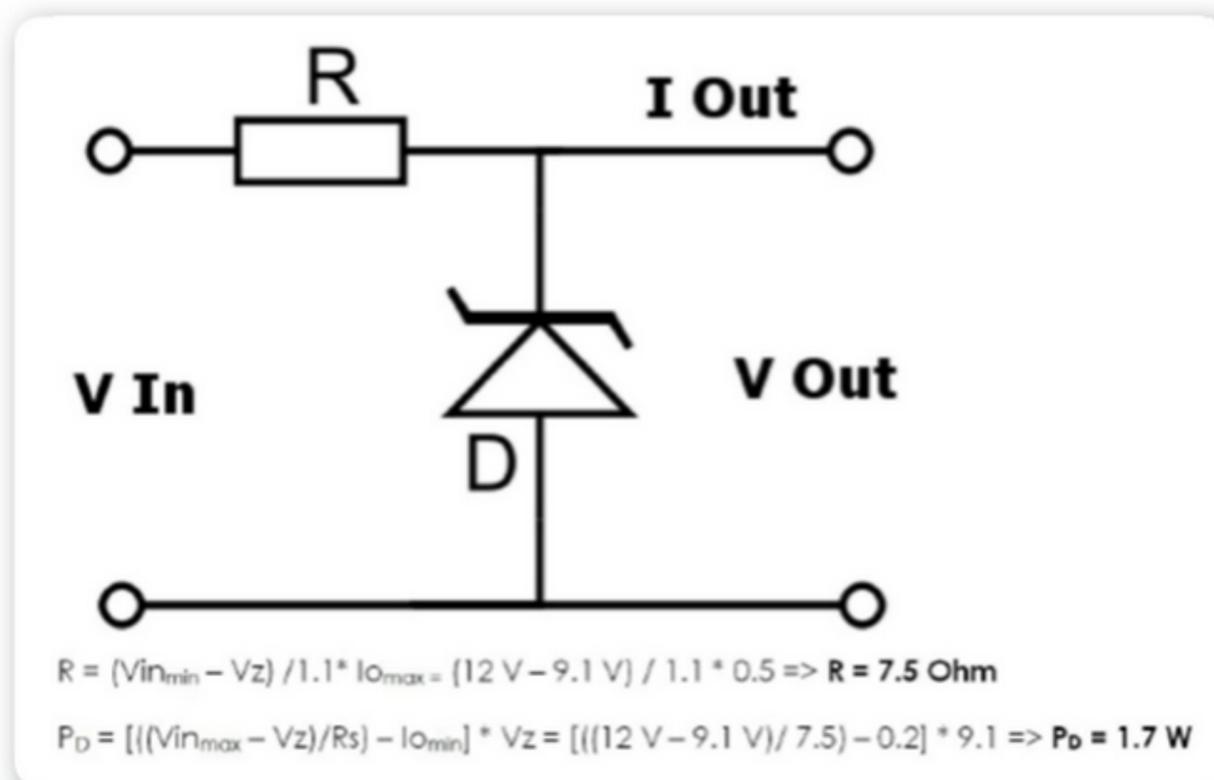


Figura 6. Los diodos rectificadores y los diodos de pequeña señal nunca se emplean en la zona de ruptura. En ella utilizamos diodos Zener.

Existen dos tipos de configuraciones: rectificación de **media onda** y rectificación de **onda completa**, así como una variante de este, el rectificador de **onda completa en puente**. En un rectificador de media onda, conectamos un diodo entre el secundario del transformador y el circuito que deseamos alimentar o cargar. Durante el semiciclo positivo de la tensión en el primario, el secundario tiene una tensión positiva entre sus extremos. Si el diodo se encuentra polarizado directamente, la tensión de pico en la salida de la fuente será igual al valor pico V_P (o valor máximo, V_M) de la tensión en el secundario ($V_p = \sqrt{2} * V_{ef}$). Durante el semiciclo negativo de la tensión en el primario, en el secundario tenemos una media onda negativa de tensión que polariza en inverso al diodo y, por lo tanto, la tensión de salida es igual a cero.

En la salida de la fuente no regulada tenemos un tipo de onda que llamamos **señal de media onda** porque los semiciclos negativos se han eliminado. La corriente en la carga es directa pulsante, comienza en cero en el inicio del ciclo, aumenta hasta el valor máximo en el pico positivo para disminuir hasta llegar a cero, donde permanece todo el semiciclo negativo. Un rectificador de media onda (**RMO**) es sencillo, pero como el diodo polarizado en inverso bloquea los semiciclos negativos de la señal rectificada, no utilizamos toda la energía disponible y el núcleo del transformador está magnetizado permanentemente (saturado) y deforma la onda rectificada.

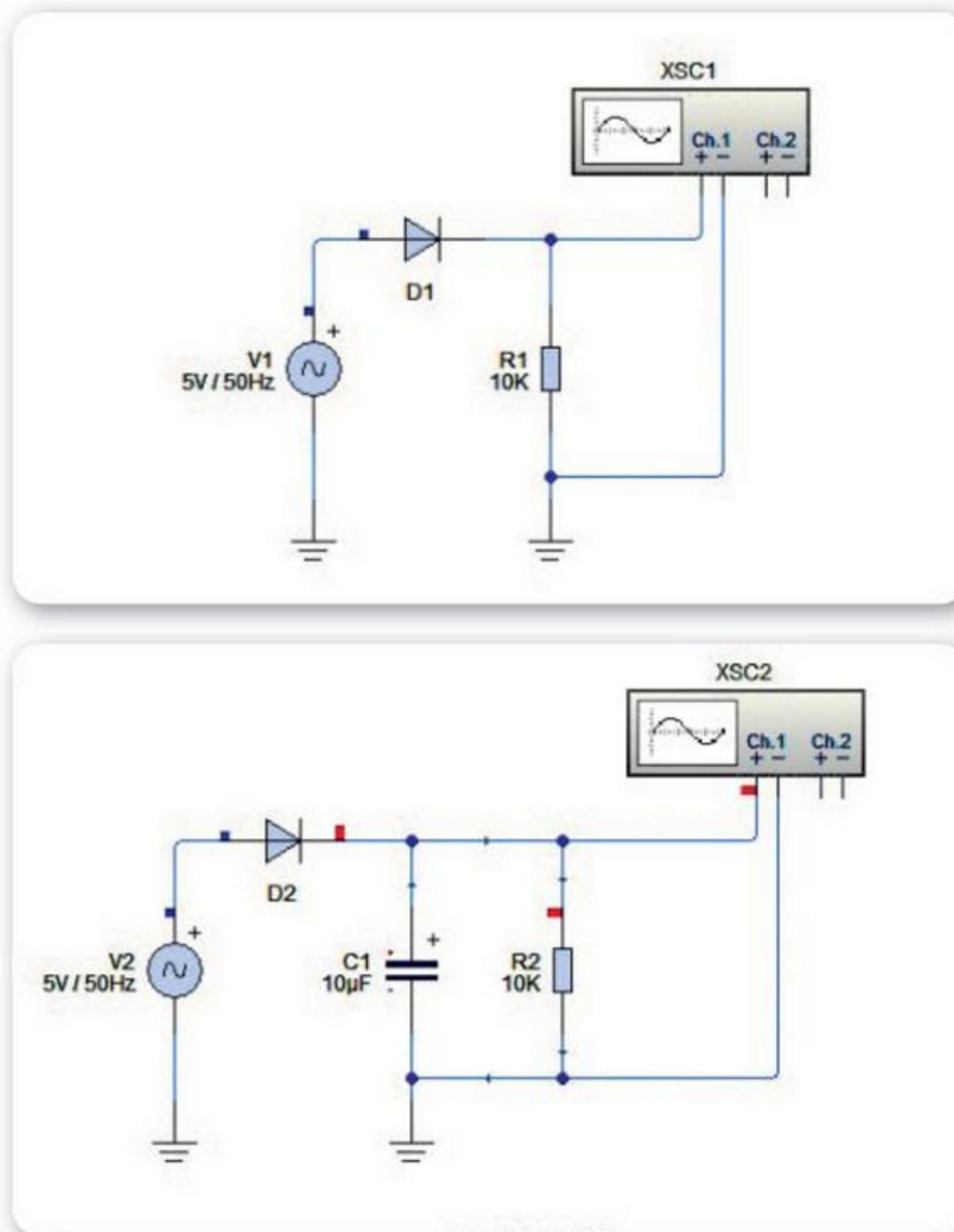


Figura 7. En la simulación de un rectificador de media onda (**RMO**), vemos la forma de onda en el diodo y cómo varía la tensión en el condensador.

El rectificador de onda completa (**ROC**) supera las desventajas del RMO y utiliza un transformador con punto medio en el secundario con diodos D1 y D2 en cada una de las ramas. Dada la conexión central en el secundario, el ROC equivale a dos RMO: D1 está polarizado en directo en el semiciclo positivo de la tensión del secundario y se bloquea durante el semiciclo negativo de ella (momento en que conduce el diodo polarizado en directo durante este semiciclo, D2, bloqueado durante el semiciclo positivo de la tensión del secundario). Obtenemos una corriente continua unidireccional sobre la carga. Cada diodo soporta una tensión inversa de pico igual a dos veces el valor máximo de la tensión en cada mitad del bobinado secundario.

Un rectificador de onda completa en **punto de Graetz (ROCG)** utiliza cuatro diodos (D1, D2, D3 y D4) y un transformador sin punto medio, más económico. En cada semiciclo de la tensión del secundario conducen simultáneamente dos diodos y cada diodo soporta una tensión inversa de pico (TPI) igual a la tensión máxima en el secundario. La tensión sobre la carga es el doble de la obtenida con un ROC. Durante el semiciclo positivo de la tensión alterna en el secundario conducen dos diodos: D3 y D2, pero durante el semiciclo negativo conducen los diodos D1 y D4.

Para este diodo, la **corriente de pico** está limitada a 30 A cuando se enciende la fuente, y su valor depende del número de ciclos necesarios para cargar el condensador de filtro: 30 A para un ciclo, 24 A para dos ciclos, 18 A para cuatro ciclos, y así sucesivamente. La capacidad del condensador de filtro es un valor central para determinar si el diodo soportará la corriente inicial. Con valores de capacidad de filtro superiores a 1000 uF, la constante de tiempo de carga es tan grande que el condensador necesita varios ciclos de la tensión alterna de entrada para cargarse por completo. Inicialmente, el condensador de filtro está descargado, y, al encender la fuente, la corriente inicial es muy elevada. Pueden dañarse el diodo rectificador y el condensador electrolítico de filtro como consecuencia del calentamiento y la formación de gases en el electrolito.

La **etapa de filtrado** convierte la tensión continua pulsante a la salida del rectificador en una tensión continua constante similar a la de una batería. La función del **filtro** ideal es reducir la componente alterna en la salida del rectificador, impedir que llegue a la carga y permitir el paso de la componente continua hacia la carga. Se comporta como un filtro pasa bajos que elimina el ripple en la salida de la fuente. El filtro más sencillo es un condensador electrolítico en paralelo que se



REGULADORES CON ZENER



Para 9 V, el valor Zener más cercano es 9.1 V. Además, definimos que la corriente que circula por el diodo Zener sea 1/10 de la máxima corriente de carga, de modo de asegurar el diodo bajo la carga máxima. Aplicando las fórmulas, obtenemos una resistencia serie de 7.5 Ohm y una potencia disipada de 1.7 W (seleccionamos 5 W para asegurarnos el funcionamiento aun sin corriente de carga).

carga a la tensión máxima (tensión de pico, VP) y suministra corriente a la carga cuando el/los diodo/s no conduce/n.

En un RMO el condensador se carga a la tensión máxima, VP, durante el primer cuarto de ciclo positivo de la tensión del secundario.

Cuando el semiciclo positivo de la tensión cae por debajo del máximo, el diodo se polariza en inverso y el condensador se descarga a través de la resistencia de carga. La constante **tiempo de descarga** (producto de la resistencia de carga y el valor de la capacidad) debería ser mucho mayor que el período **T** de la tensión alterna de entrada. El condensador perderá un pequeño valor de la tensión de carga durante el tiempo de polarización inversa del diodo. En el segundo semiciclo positivo, el condensador se carga nuevamente hasta VP y la tensión de carga se aproxima a una tensión continua constante con una pequeña desviación, ripple, debido a la carga y descarga del condensador.

Para reducir el ripple, aumentamos la constante tiempo de descarga. La tensión de ripple se mide en valor pico a pico (VRpp) y es el cociente entre la corriente continua en la carga dividida y el producto de la frecuencia de la tensión de ripple multiplicado por el valor de la capacidad. En un ROC, la frecuencia de entrada es el doble de un RMO, el condensador se carga el doble de veces en el mismo intervalo de tiempo y se descarga durante la mitad del tiempo. Así, el ripple es menor y la tensión continua de salida se acerca más a la tensión máxima (VP).

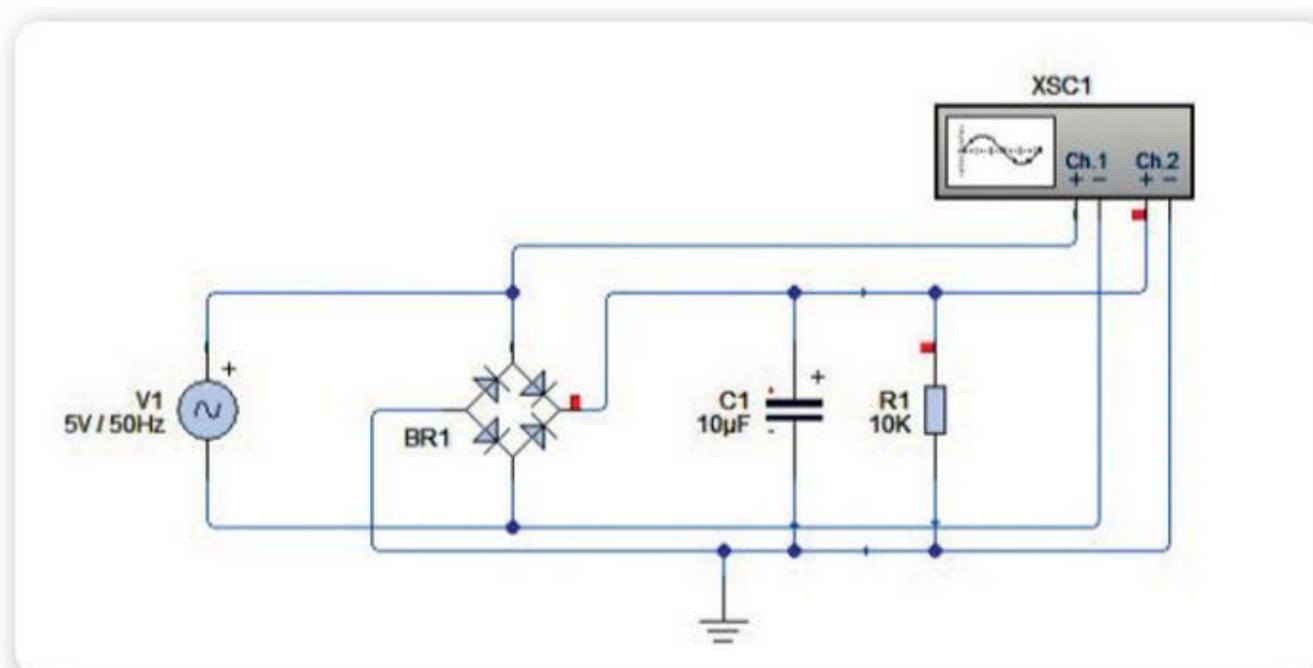


Figura 8. Un rectificador de onda completa (**ROC**) en puente de Graetz utiliza cuatro diodos y un transformador con punto medio, más económico.

REGULAMOS LA
TENSIÓN DE SALIDA
DE UNA FUENTE
LINEAL CON UN
REGULADOR



Para mantener constante la tensión de salida de una fuente, independientemente de la corriente de la carga y de la variación en la tensión alterna de entrada, utilizamos un **regulador de tensión**. Este se clasifica en fijo y variable y el primero, además, en serie y paralelo.

Un **regulador fijo** se construye con un diodo Zener, un diodo con tensión constante en la zona de ruptura. Por el diodo Zener pasa una corriente que debemos limitar para no destruir el diodo. La potencia máxima en un Zener es el producto de la tensión entre sus terminales y la corriente que pasa por él.

Del mismo modo, la máxima corriente que puede conducir resulta del cociente entre la potencia nominal y la tensión nominal. Así, un diodo de 10 V y 50 W no podría conducir más de 5 A. La resistencia limitadora en serie con el Zener restringe la corriente que circula por ella (suma de la corriente en la carga y la corriente en el diodo Zener) a un valor seguro.

Para regular mayores corrientes, incorporamos transistores en serie con la tensión de alimentación y utilizamos un diodo Zener como referencia de tensión fija para la base del transistor. Podríamos acudir a reguladores integrados del tipo 78XX –como el 7812 para 12 V– que contienen un regulador serie integrado a compensaciones de temperatura y protección contra cortocircuitos. Con un regulador variable obtenemos distintos valores de tensión de forma continua y lo podemos construir mediante componentes discretos (transistores, condensadores y resistencias) o con circuitos integrados especializados como el típico LM317.



FUENTE COMPLETA



El **LM317** es un regulador variable que entrega una corriente de 1.5 A con tensiones ajustables de 1.25 a 37 V. Incrementamos la corriente de salida mediante un disipador externo o recurriendo a un transistor de potencia exterior. Limitamos la corriente máxima con un transistor adicional que evita dañar los semiconductores. En www.electronics-lab.com/articles/LM317 hay una calculadora para LM317.

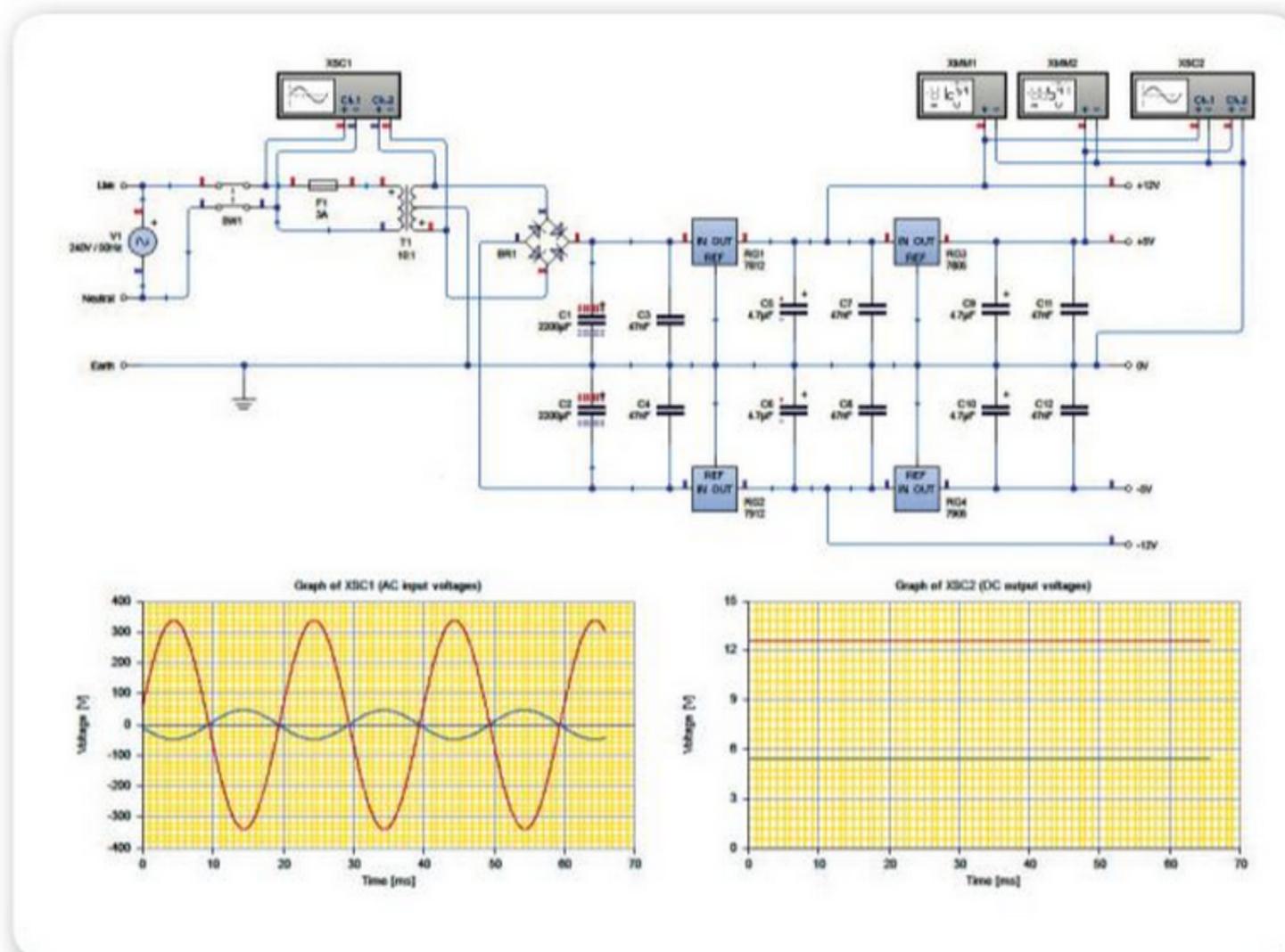


Figura 9. Fuente construida por medio de reguladores de tensión fijos del tipo 7812-7912-7805 y 7905, con cuatro salidas de ± 5 V y ± 12 V.



Fuentes conmutadas

Una fuente conmutada es una buena alternativa cuando los requerimientos se relacionan con la eficiencia y el espacio reducido. Tenemos dos tipos de organización para una fuente conmutada: conmutada en el primario o conmutada en el secundario. Una **fente de alimentación conmutada en el primario** responde al esquema: rectificador, conmutador, transformador, segundo rectificador, filtro, regulador y salida; y se caracteriza porque la tensión de red se rectifica, se suaviza y luego se interrumpe (se conmuta, de allí el nombre de fuente conmutada).

EL REGULADOR
CONMUTADO SIRVE
PARA CORRIENTES
DE GRAN CARGA
A BAJA TENSIÓN



La **interrupción** significa que la tensión de continua se conmuta periódicamente a una frecuencia entre 40 a 200 kHz por medio de un transistor de potencia (bipolar, MOSFET, entre otros).

Si comparamos este tipo de fuente con una fuente de alimentación regulada linealmente, la diferencia fundamental radica en que el transistor de potencia no actúa como una resistencia variable sino como interruptor. De ese modo, se genera una tensión alterna cuadrada que se transforma hacia el circuito secundario por medio de un transformador de alta frecuencia.

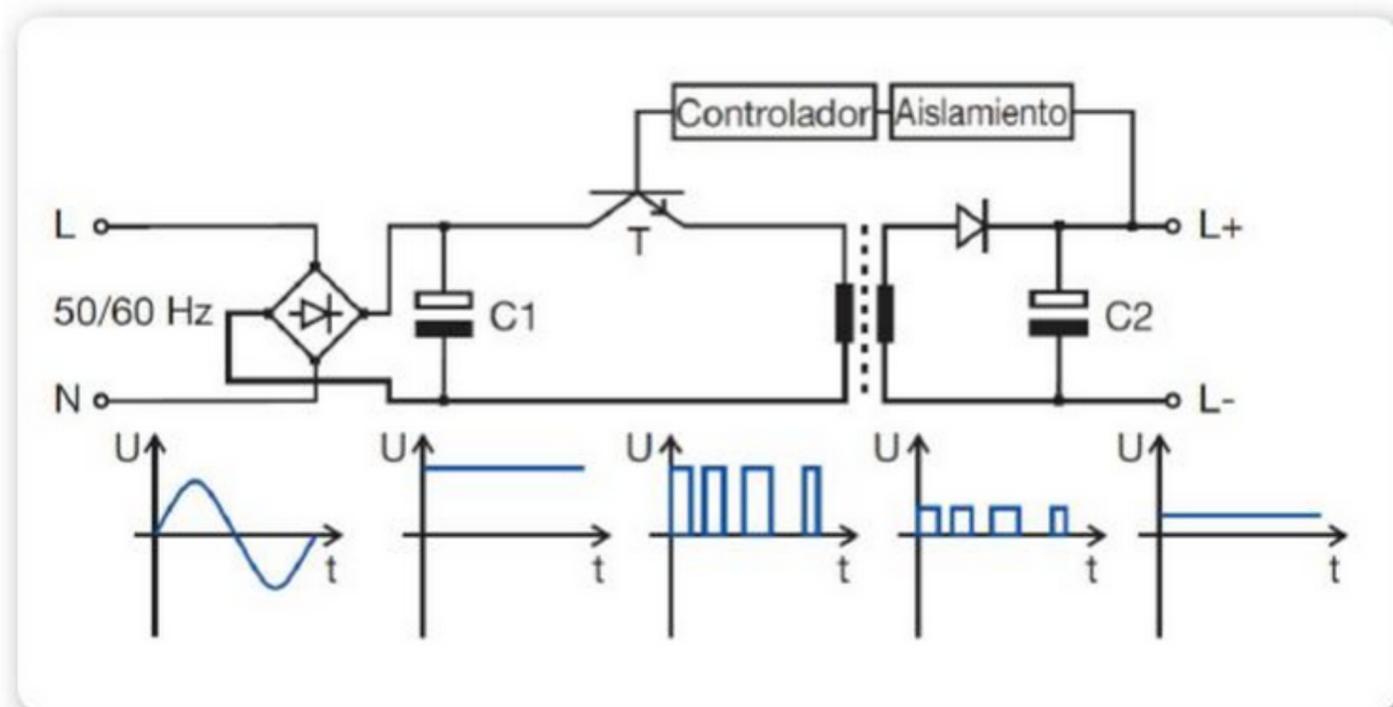


Figura 10. La fuente conmutada en primario se utiliza para energizar circuitos electrónicos y se vale de un transformador de alta frecuencia.

Por último, en el circuito secundario se rectifica y suaviza la tensión. Controlamos la cantidad de energía que se transforma hacia el circuito secundario en función de la carga, variando la **tasa de interrupción**. A mayor cantidad de tiempo en conducción para el transistor, mayor será la cantidad de energía transformada hacia el circuito secundario (esquema de modulación por ancho de pulsos, PWM).

Como consecuencia de utilizar una tensión alterna de alta frecuencia, las fuentes de alimentación conmutadas en primario tienen una gran ventaja relacionada con el tamaño de su transformador, que es mucho menor que el tamaño requerido para transformar bajas frecuencias (50/60 Hz). Así, se reduce el peso y la disipación en el interior de la unidad, con el consiguiente aumento en el rendimiento: de 70 % a 90 %.

Una ventaja del esquema conmutado en el primario es que la tensión de salida no depende directamente de la tensión de entrada, por lo que podemos emplearlo dentro de un amplio rango de tensiones de entrada. Mediante el condensador conectado a la salida del rectificador podemos compensar interrupciones en la tensión de red de varios milisegundos, y es el tamaño de este condensador el que limita las posibilidades de la compensación, en especial, en las fuentes de tamaño pequeño.

Entonces, es fundamental encontrar un equilibrio práctico entre el tamaño de la fuente de alimentación y el tiempo de compensación. Empleamos las fuentes de alimentación conmutadas en primario en una multitud de aplicaciones de alimentación de dispositivos electrónicos.

El diseño de las **fuentes de alimentación conmutadas en secundario** es similar al diseño de las fuentes de alimentación conmutadas en primario, excepto por el hecho de que la interrupción se efectúa en el secundario. Entonces, tenemos que utilizar un transformador mucho mayor, dado que tiene que transformar una tensión de red de frecuencia de 50/60 Hz. No obstante, el transformador también actúa como un filtro y, por lo tanto, minimiza el ruido incorporado a la red de distribución eléctrica.

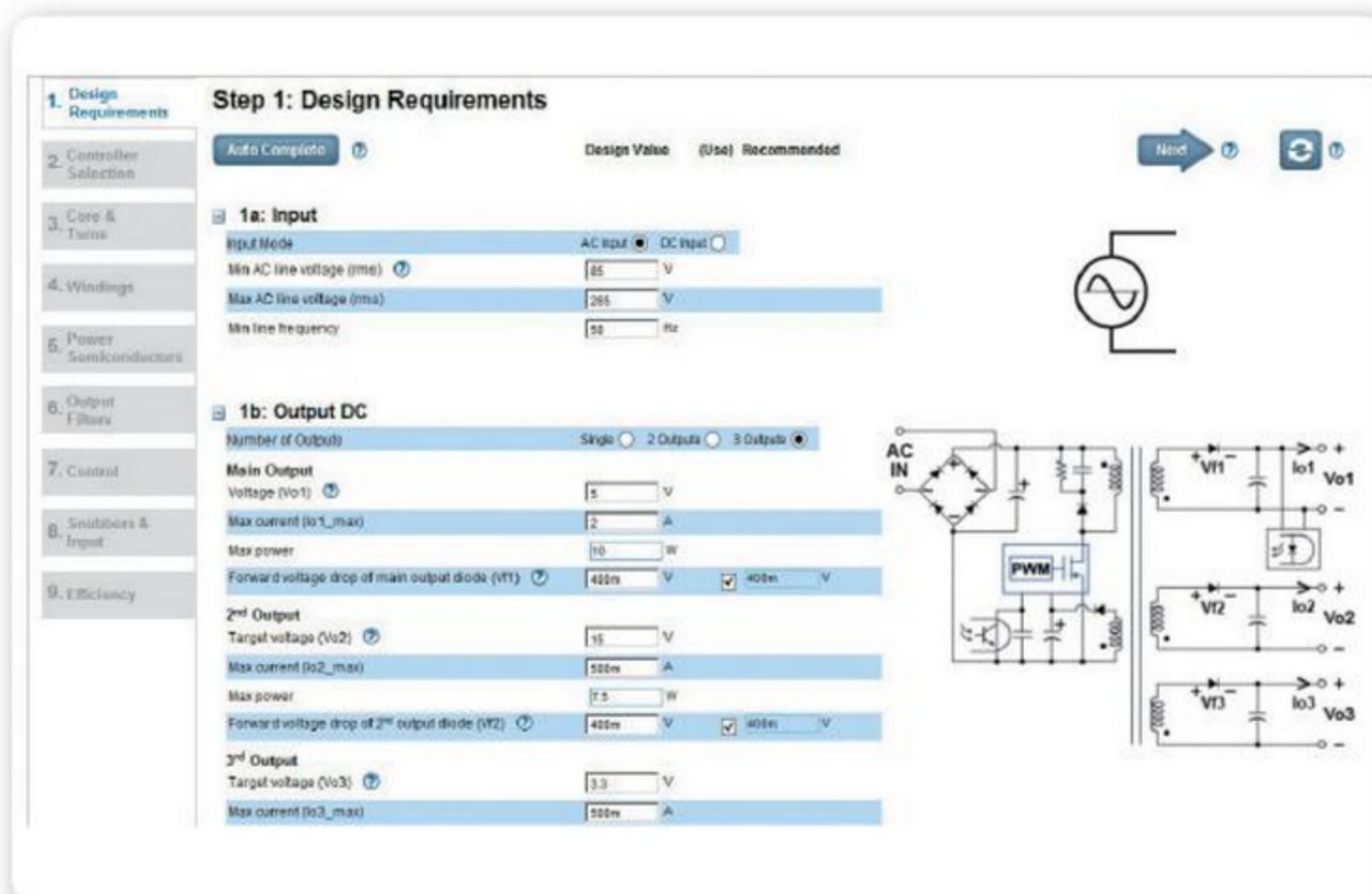


Figura 11. Al utilizar PSW para el diseño de la fuente, el primer paso es ingresar los requerimientos.

Las tres configuraciones básicas

Los reguladores conmutados vienen en tres configuraciones básicas: reductor, elevador e inversor.

En la **versión reductora** del conmutador, los pulsos rectangulares de la base saturan y cortan alternativamente el transistor de salida durante cada ciclo. Así, se produce una onda cuadrada en la entrada del filtro (LC). La función del filtro LC es bloquear la componente alterna de la tensión de salida y permitir el pasaje de la corriente continua hacia la carga. Como consecuencia de la conmutación (saturación-corte), el valor medio obtenido siempre es menor que la tensión continua de entrada, característica que da nombre a esta versión del conmutador. Esta configuración es la mejor elección cuando se necesita una tensión baja y una corriente alta, como en el caso de las PC, que funcionan con tensiones de + 5 V y una corriente alta, por lo que los conmutadores reductores se utilizan ampliamente en las computadoras personales.

En la **versión elevadora** del regulador conmutado, también el transistor funciona alternativamente en corte y saturación. Cuando el transistor se satura, la corriente circula por la inductancia L; mientras que, cuando el transistor se corta, el campo magnético almacenado en la bobina induce una tensión grande con polaridad opuesta, lo que mantiene circulando la corriente en la misma dirección. Además, la tensión inductiva es mayor que la tensión de entrada, por lo que el circuito funciona como elevador de tensión.

Por último, en la **versión inversora** del regulador, cuando el transistor está saturado, la corriente circula por la bobina; en cambio, cuando el transistor está cortado, el campo magnético en la bobina no puede sostenerse, y el inductor L se descarga a través del condensador desde masa. Así, la tensión en bornes del condensador es negativa.



MÁS SOFTWARE DE DISEÑO



Presentamos dos sitios web para el diseño de fuentes conmutadas. La Web www.poweresim.com/?ApplicationIdx=8&ChangeApplication=8 ofrece cálculos para potencias mayores y una amplia variedad de modelos. Por otro lado, el sitio web www.ti.com/ltds/ti/analog/powermanagement/power_portal.page, de la empresa Texas Instruments, tiene la herramienta Power Management, que también facilita el diseño de fuentes conmutadas.

Diseño online de una fuente conmutada

El diseño de fuentes con reguladores lineales utiliza transformadores que funcionan con 50/60 Hz, lo que produce soluciones ineficientes, ya sea por el costo elevado, el peso excesivo y el volumen, además de su bajo rendimiento con la consiguiente generación de calor. Por el contrario, podemos construir una fuente conmutada recurriendo a soluciones “en un chip” que facilitan el diseño de fuentes conmutadas capaces de trabajar con tensiones de entrada entre unos 80 y 230 V con rendimiento elevado (80 % - 90 %), bajo costo y volumen, y utilizando muy pocos componentes.

Si bien el diseño de una fuente conmutada es una actividad compleja, vamos a utilizar una de las herramientas disponibles en internet dedicadas a facilitarnos el proceso de diseño. Se trata de la herramienta online **Power Supply WebDesigner (PSW)**, de Fairchild Semiconductor, disponible en el sitio web www.fairchildsemi.com/support/design-tools/power-supply-webdesigner/, para diseñar **fuentes conmutadas del tipo Flyback**. En esta Web, también encontramos guías y ejemplos de diseño que utilizan los componentes de esta empresa.

Si bien las fuentes conmutadas tipo Flyback tienen el inconveniente de la bobina –que para potencias superiores a 150 W es demasiado grande–, se utilizan ampliamente en aplicaciones de baja potencia como SmartTVs, PCs y cargadores, entre otras. Aun así, la herramienta de Fairchild es interesante, ya que recurre a un controlador y a un transistor MOSFET en un mismo integrado, reduciendo el número de elementos a utilizar.

Las etapas que tenemos que recorrer para diseñar una fuente conmutada con PSW se relacionan con el diseño propiamente dicho, el análisis, la lista de materiales (**BOM**) y el reporte final. Como primer paso, en el diseño de una fuente de potencia CA a CC, tenemos que definir las especificaciones de entrada y salida de la fuente. Vamos a suponer que las especificaciones de entrada son las siguientes: 85 a 265 VAC, frecuencia de línea de 50 Hz y 80 % de eficiencia. Para las

CON UNA FUENTE
CONMUTADA
LOGRAMOS
EFICIENCIA Y
ESPACIO REDUCIDO



especificaciones de salida, proponemos alimentar cargas múltiples; en este caso, tres cargas: un display LCD (15 V / 1 A), un cargador de baterías (5 V / 2 A), que será la principal salida regulada, y un microcontrolador (3.3 V / 0.5 A).

	A	B	C	D	E
1	Ref	Qty	Part Number	Manufacturer	Description
2	U1	1	FSGM0465RLDTU	Fairchild Semiconductor	Power Switch ICs - Power Distribution SMPS Power Switch 4A,
3	Cb	1	C0603C151J4GACTU	Kemet	Cap Ceramic 150pF 16VDC C0G 5% SMD 0603 Paper T/R
4	Cdc	1	TC80	Cornell Dublier	CAP ALUM 80UF 450V AXIAL
5	Cf	1	GRM31C5C1H753JA01L	Murata	Cap Ceramic 0.075uF 50VDC C0G 5% SMD 1206 Embossed T/R
6	Cin1	1	158X224	Cornell Dublier	Cap Film 0.22uF 275VAC PET 20% 25 X 8 X 17.5mm RDL 22.5r
7	Cin2	1	158X154	Cornell Dublier	Cap Film 0.15uF 275VAC PET 20% 25 X 6.5 X 16mm RDL 22.5r
8	Co1	3	EEF-UE0K161ER	Panasonic	CAP 150UF 8V SPEC POLYMER SMD
9	Co2	2	94SL226X0025FBP	Vishay	Aluminum Organic Polymer Capacitors 22uf 25volts 20%
10	Co3	1	APXC6R3ARA181MF60G	United Chemi-con	CAP 180UF 6.3V ELECT POLY SMD
11	Co1_3	1	0603ZC104JAT2A	AVX	Cap Ceramic 0.1uF 10VDC X7R 5% SMD 0603 Paper T/R
12	Co2_3	1	06033C104JAT4A	AVX	Cap Ceramic 0.1uF 25VDC X7R 5% SMD 0603 Paper T/R
13	Co3_3	1	06036C104JAT2A	AVX	Cap Ceramic 0.1uF 6.3VDC X7R 5% SMD 0603 T/R
14	Copto	1	12063G104ZAT2A	AVX	Multilayer Ceramic Capacitors (MLCC) - SMD/SMT 25V 0.1uF
15	Cps	1	564R30GAD47	Vishay	Cap Ceramic 0.0047uF 3000VDC Z5U 20% 15.7 X 5mm RDL 9.6
16	Csn	1	BFC241919102	Vishay	CAP FILM MKP .0091uF 400VDC 2%
17	Cvcc_bk	1	EEU-EB1H470	Panasonic	Cap Aluminum 47uF 50VDC 20% 6.3 X 15mm RDL 2.5mm 87mV
18	Cvcc_hf	1	08055C224KAT2A	AVX	Cap Ceramic 0.22uF 50VDC X7R 10% SMD 0805 Embossed T/R
19	D1	1	SB380	Fairchild Semiconductor	DIODE SCHOTTKY 3A 80V DO-41
20	D2	1	S1D	Fairchild Semiconductor	DIODE GPP 1A 200V GEN PURP SMA
21	D3	1	SS15	Fairchild Semiconductor	DIODE SCHOTTKY 1A 50V SMA
22	Din1	1	MB6S	Fairchild Semiconductor	Diode Rectifier Bridge Single 600V 0.5A 4-Pin SOIC N T/R
23	Din2	1	MB6S	Fairchild Semiconductor	Diode Rectifier Bridge Single 600V 0.5A 4-Pin SOIC N T/R
24	Din3	1	MB6S	Fairchild Semiconductor	Diode Rectifier Bridge Single 600V 0.5A 4-Pin SOIC N T/R
25	Din4	1	MB6S	Fairchild Semiconductor	Diode Rectifier Bridge Single 600V 0.5A 4-Pin SOIC N T/R
26	Dsn	1	EGP10K	Fairchild Semiconductor	Diode Switching 800V 1A 2-Pin DO-41 T/R
27	Dvcc	1	FMKA130	Fairchild Semiconductor	Diode Schottky 130V 1A 2-Pin DO-214AC T/R
28	Dzvcc	1	MMS25250B	Fairchild Semiconductor	DIODE ZENER 20V 500MW SOD-123
29	Fuse	1			Resistor 1Q
30	Lin	1			Inductor 20mH

Figura 12. Lista de materiales (BOM) de la fuente conmutada, obtenida mediante uno de los reportes específicos.

A continuación, el software nos propone definir la topología de la fuente conmutada, en base a dos configuraciones: **convertidor tipo Flyback conmutado en el primario** (para una carga) o **convertidor tipo Flyback conmutado en el secundario** (soporta varias cargas). Seleccionamos el último, ya que nos permite tener hasta tres cargas y potencias de salida mayores a 100 W (la otra configuración solamente permite una carga y hasta 30 W de potencia).



FUENTE CONMUTADA DE TRES SALIDAS

PSW permite diseñar fuentes conmutadas tipo Flyback seleccionando un controlador Fairchild y el MOSFET asociado para hacer un diseño asistido eficaz, obtener una lista de materiales, seleccionar el proveedor de componentes que optimice la relación beneficio-coste, simular el funcionamiento beneficioso para ajustar el diseño preprototipo y diseñar el transformador de alta frecuencia.

A partir de este punto, podemos seleccionar entre dos alternativas para el diseño: la opción **Auto Diseño**, donde solo debemos introducir algunos datos básicos y elementales para que el simulador nos informe respecto de cada componente necesario para la construcción de la fuente de alimentación (aunque utilizando productos de Fairchild Semiconductors); y la opción **Diseño Avanzado**, en la que podemos elegir todos los componentes para utilizar en el diseño: el circuito integrado conmutador + MOSFET, diodos, transformador de conmutación (núcleo de ferrite), circuito amortiguador, filtros de salidas, el circuito empleado como filtro de línea de corriente alterna de entrada y las resistencias y capacitores.

Así, podemos obtener estimaciones rápidas y precisas sobre el rendimiento de nuestro diseño, al mismo tiempo que trabajamos en ajustar los modelos presentados por el simulador; todo durante el mismo proceso de diseño. La herramienta online, además, nos permite utilizar el tiempo de trabajo para realizar un análisis detallado y adquirir conocimientos acerca del modo en que nuestro diseño funcionará en el mundo real. La herramienta simula los diagramas de Bode, ofreciendo las formas de onda y los gráficos en estado estacionario (Stand-By) y en pleno funcionamiento. Como diseñadores, nos permite analizar cómo funciona el diseño y asegurarnos de que va a operar en forma satisfactoria desde el inicio.

Entonces, la herramienta PSW nos ayuda en el diseño óptimo de fuentes de alimentación conmutadas, sin necesidad de tener que construir un prototipo ni conocer detalladamente las complejas expresiones matemáticas que serían necesarias utilizando el camino tradicional. De este modo, si colocamos valores no compatibles con nuestro diseño, PSW se encarga de sugerirnos valores coherentes con un funcionamiento estable y equilibrado.

Otra de las opciones que nos brinda el sistema es la selección del tipo de fuente Flyback que realizaremos: con realimentación directa desde uno de los bobinados del transformador de conmutación o mediante un optoacoplador encargado de monitorear la tensión de salida (PSR o SSR).



EL DISEÑO
DE FUENTES
LINEALES UTILIZA
TRANSFORMADORES
DE 50/60 HZ



Cuando el diseño está completo, *Power Supply WebDesigner* (PSW) crea una **lista de materiales** (en inglés, *Bill of Materials*, **BOM**), que puede ser enviada al departamento de compras de una empresa para adquirir los componentes online con el consecuente ahorro de tiempo y recursos en cuanto a la documentación o la búsqueda de proveedores. También podemos almacenar los diseños para futuras consultas o cederlos a otros proyectistas.

El sistema adopta los datos que le suministramos con las dimensiones del transformador de ferrite por utilizar o nos recomendará –si no seleccionamos uno adecuado– las medidas y formas por utilizar, en función de la potencia que entregue la fuente a la salida; gráficos y estadísticas completas respecto del funcionamiento y eficiencia de la fuente, según las potencias de entrada y salida solicitadas, y los oscilogramas completos de funcionamiento, de acuerdo con los requerimientos indicados en la simulación.

PSW nos sugiere utilizar el circuito integrado conmutador de potencia **FSGM0465RLDTU** perteneciente a la serie Green FPS de Fairchild, una nueva familia de chips *Fairchild Power Switch* (**FPS™**) que ofrece convertidores de conmutación de alta eficiencia y bajo nivel de interferencias electromagnéticas (EMI). Son especialmente adecuados para conseguir alta fiabilidad en los sistemas reproductores de DVD, en los decodificadores, en los monitores LCD, en otros diseños de potencias de 25 W e inferiores y en fuentes de alimentación.

Se trata de un controlador integrado por pulsos modulados en ancho (*Pulse Width Modulation*, **PWM**) y un transistor de efecto de campo (*Field Effect Transistor*, **FET**), específicamente diseñados para el suministro de potencia en modos de conmutación offline (*Switch-Mode Power Supplies*, **SMPS**), con un mínimo de componentes externos. En base a la técnica de conmutación **valley** por Fairchild, sus



EL CONTROL DE LAS FUENTES



En el circuito de control, observamos las diferencias entre la fuente convencional y la conmutada. En la primera, el regulador es lineal, opera en el secundario y controla la tensión de salida absorbiendo la diferencia de tensión entrada-salida y disipando la potencia en forma de calor. Este es uno de sus inconvenientes.

En una fuente Flyback, el controlador es del tipo **SÍ/NO** e interviene sobre la tensión de salida.

conmutadores incrementan la eficiencia de conversión de potencia en un 1 % y reducen el nivel de EMI hasta en un 5 dB, en comparación con las topologías de conmutación de los convertidores convencionales.

El controlador de PWM incluye un oscilador integrado de frecuencia fija **UVLO** (*UnderVoltage LockOut*), un **LEB** (*Leading Edge Blanking*), un controlador de puerta optimizado, un arranque suave interno, fuentes de corriente precisas compensadas en temperatura para la compensación del lazo y autoprotección para los circuitos eléctricos. Esta combinación nos ofrece ventajas en cuanto al costo de los sistemas, el número de componentes, la fiabilidad, el tamaño y el peso de la integración del IC de PWM y del MOSFET, junto con la alta fiabilidad y eficiencia energética. Es una solución todo en uno.

Los convertidores conmutados de tipo Flyback funcionando en modo valley son adecuados para los diseños con alta relación prestaciones-precio.

Estos convertidores de potencia funcionan con una estrecha variación de frecuencia, al mismo tiempo que mantienen el funcionamiento en conmutación en el modo valley. Este modo de funcionamiento permite que el MOSFET encienda con un mínimo drenaje de tensión y ofrece modulación de frecuencia inherente que produce una reducción significativa del ruido de tipo EMI. Este modo de funcionamiento permite también la conmutación suave con las consiguientes mejoras de la eficiencia.

FAIRCHILD SEMICONDUCTOR®		Flyback/Forward PWM - FSGM0465R
Design Requirements		
Requirements		
Input Mode		AC
Min AC line voltage (rms)		85V
Max AC line voltage (rms)		265V
Min line frequency		50Hz
Number of Outputs		3
Main Output		
Voltage (Vo1)		5V
Max current (Io1_max)		2A
Max power		10W
Forward voltage drop of main output diode (VF1)		400mV
2nd Output		
Target voltage (Vo2)		15V

Figura 13. Informe final que contiene los requerimientos de diseño de la fuente conmutada, obtenido mediante uno de los reportes específicos.

La fábrica lo suministra en un encapsulado DIP o LSOP de ocho contactos que ahorra espacio. Los dispositivos tienen protección contra sobretensiones (**OVP**, *OverVoltage Protection*), protección contra sobrecargas (**OLP**, *OverLoad Protection*), protección contra excesos anormales de corriente (**AOCP**, *Abnormal OverCurrent Protection*) y apagado por exceso de temperatura (**TSD**, *Thermal ShutDown*).

Entre las características básicas, podemos citar: convertidores conmutados integrados; baja emisión de interferencias electromagnéticas (EMI): 5 dB por debajo de las soluciones convencionales; elevada eficiencia: un 1 % mejor que las soluciones convencionales; consumo en reposo de 0.2 W (sin carga); limitación de corriente pulso a pulso; protección integrada (OLP, OVP, AOCP, TSD, UVLO).

Las aplicaciones más comunes son en fuentes de alimentación para televisores LCD, monitores, combinación STB-DVD y PCs, entre otros. Entre los valores típicos del chip FSGM0465RLDTU, tenemos una temperatura de operación para la juntura comprendida entre $-40\text{ }^{\circ}\text{C}$ a $+125\text{ }^{\circ}\text{C}$ y $R_{DS(ON)}$ máxima de 2.6 Ohm.

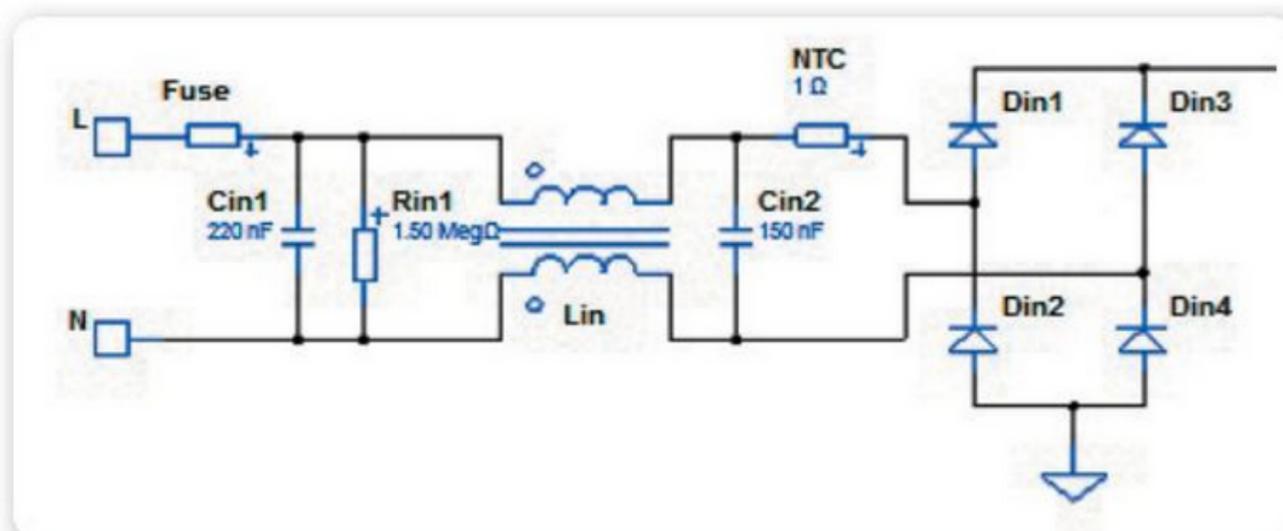
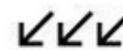


Figura 14. Sección rectificadora perteneciente a la fuente de alimentación conmutada diseñada mediante la aplicación online PSW de Fairchild.



DISEÑO DEL TRANSFORMADOR



Los conocimientos requeridos, la gran cantidad de núcleos de ferrite disponibles y la dispersión de especificaciones entre los fabricantes obstaculizan el proceso de diseño de un transformador. En los sitios de las empresas Epcos, www.epcos.com, TDK, www.tdk.co.jp/tefe02/ferrite.htm y Magnetics, www.mag-inc.com, se encuentra software para asistir la fabricación de inductores.

Diseño y reguladores

Al momento de **diseñar una fuente**, son varias las necesidades que debemos tener en cuenta. Esto dependerá, claro está, del circuito o aplicación que vayamos a alimentar.

Si bien la **fuentes de alimentación** será la que le entregue la energía al circuito, esta no será la que la genere. La fuente también necesita ser alimentada. Esta alimentación puede provenir de la red eléctrica hogareña, de una batería o de otro medio. Al diseñar la fuente de alimentación, debemos tener en cuenta con qué la alimentaremos.

La fuente de alimentación nos entregará la tensión continua necesaria para alimentar los circuitos lógicos y digitales. También podemos diseñar fuentes de alimentación que nos provean voltajes alternos, como **senoidal**, para probar circuitos electrónicos –como, por ejemplo, un amplificador o conversor analógico digital– en nuestro laboratorio.

Las tensiones más utilizadas por los circuitos digitales son 5 y 12 V. Por esto, cualquier fuente que diseñemos contará con estas salidas fijas. También, por cuestiones prácticas, será conveniente incluir una salida de amplitud variable, generalmente entre 2 y 30 V. Con esta, podremos alimentar una amplia gama de circuitos. Además, podemos diseñar una fuente de alimentación que entregue tanto voltaje positivo como negativo.

Asimismo, al diseñar la fuente de alimentación debemos tener en cuenta la corriente que consumirá el circuito alimentado. O bien, cuando compramos una fuente de alimentación en el mercado, cuál es la corriente máxima que entrega. Si la corriente que consume es mayor que la prevista en el diseño, esta última se dañará y quedará inservible. Los componentes de la fuente de alimentación soportan una cierta cantidad de corriente circulando por ellos.



DISEÑO PERSONALIZADO

Construir una fuente de alimentación no solo nos permite definir las amplitudes de las tensiones y las corrientes máximas que entregará. Podemos construir el gabinete de la fuente de alimentación con los materiales y los colores que más nos agraden. También podemos agregarle luces y accesorios cromados para darle estilo, y que la fuente de alimentación se diferencie de las demás.



Figura 15. Las fuentes de alimentación comerciales nos permiten regular el voltaje y la corriente máxima que entregamos al circuito en un amplio rango.

Si circula por ellos una corriente mayor que la máxima soportada, el circuito se dañará. También puede suceder que, por alguna falla del circuito que estamos alimentando, como un cortocircuito, este consuma mucha más corriente que la esperada. Si no prevemos en

la fuente una forma de detectar esto y desconectar la alimentación, la fuente también resultará dañada. Varias fuentes de alimentación comerciales hoy vienen con salida de corriente regulada.



Figura 16. Esta fuente regulada posee varios tipos de conectores de salida y una variedad de voltajes de salida ya determinados entre los que podemos elegir.

Puede suceder que falle la fuente de alimentación. Si esto sucede, y hay un cortocircuito en la fuente de alimentación, esta comenzará a consumir una cantidad desmedida de corriente de la línea de alimentación. Esto nos traerá una serie de problemas. Si el cortocircuito interno a la fuente de alimentación se da en las etapas más cercanas a la salida, el exceso de corriente circulará por gran parte de ella, dañando el resto de los componentes que aún están sanos. Esto causará un aumento de la temperatura interna de la fuente de alimentación, y puede, incluso, generar un incendio.

AL DISEÑAR
LA FUENTE,
NO OLVIDEMOS
LA CORRIENTE
MÁXIMA EXIGIDA



Por otra parte, como ya mencionamos, ese exceso de corriente será consumido de la red de energía eléctrica de nuestro taller o laboratorio. Si no lo controlamos, corremos el riesgo de dañar la instalación eléctrica, o que finalmente se active la llave de protección térmica cortando la energía de la instalación. Para evitar esto, lo que debemos hacer es colocar un fusible a la entrada de la fuente de alimentación.

Los componentes electrónicos de la fuente disiparán energía en forma de calor. Para asegurarnos que la disipación de esta energía sea suficiente, debemos proveer a la fuente de disipadores o coolers que fuercen la circulación de aire por ella.

Por último, también debemos tener en cuenta la presentación. Esta dependerá de nuestro gusto, de nuestra necesidad y de la disponibilidad del mercado local. Podemos encontrar una variada oferta de gabinetes de diversos materiales, tamaños y formas. También hallaremos una oferta variada en lo que refiere a los conectores de salida y a las perillas de regulación, en caso de que decidamos que los parámetros de tensión y corriente entregados sean variables.



DISIPAR LA ENERGÍA



La fuente de alimentación genera energía. Una parte será entregada al circuito, y otra, disipada en forma de calor. Si ese calor se mantiene dentro del gabinete de la fuente, irá aumentando la temperatura hasta romper los componentes. Para evitar esto, colocamos disipadores de material que conduzcan muy bien el calor, o bien, elementos que fuercen la circulación de aire.



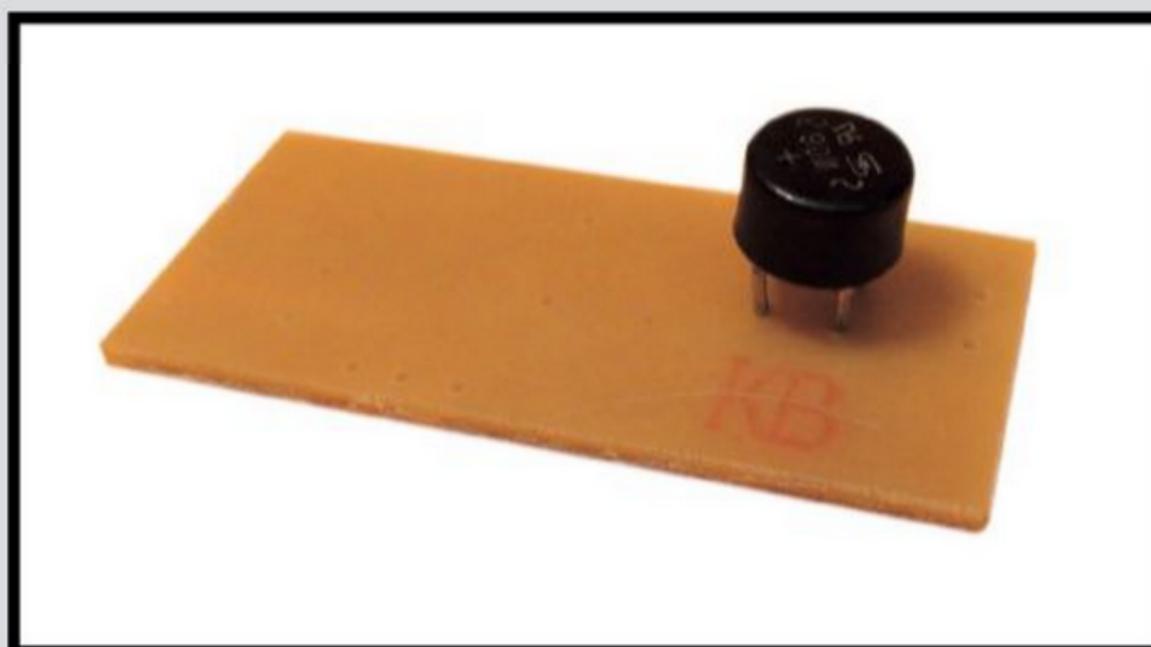
Figura 17. El cargador que usa nuestro teléfono móvil también es una fuente de alimentación que le otorga energía a la batería.

En el siguiente Paso a paso, construiremos una pequeña fuente de alimentación regulada que nos entregará 9 V y 1 A. Los componentes son: 1 transformador de 120/220 V a 18 V, 1 puente diodo de 1,5 A, 1 capacitor de 4700 μF , 1 integrado LM7809, 1 capacitor de 1 μF , un par de conectores y un placa PCB.

PAP. FUENTE REGULADA



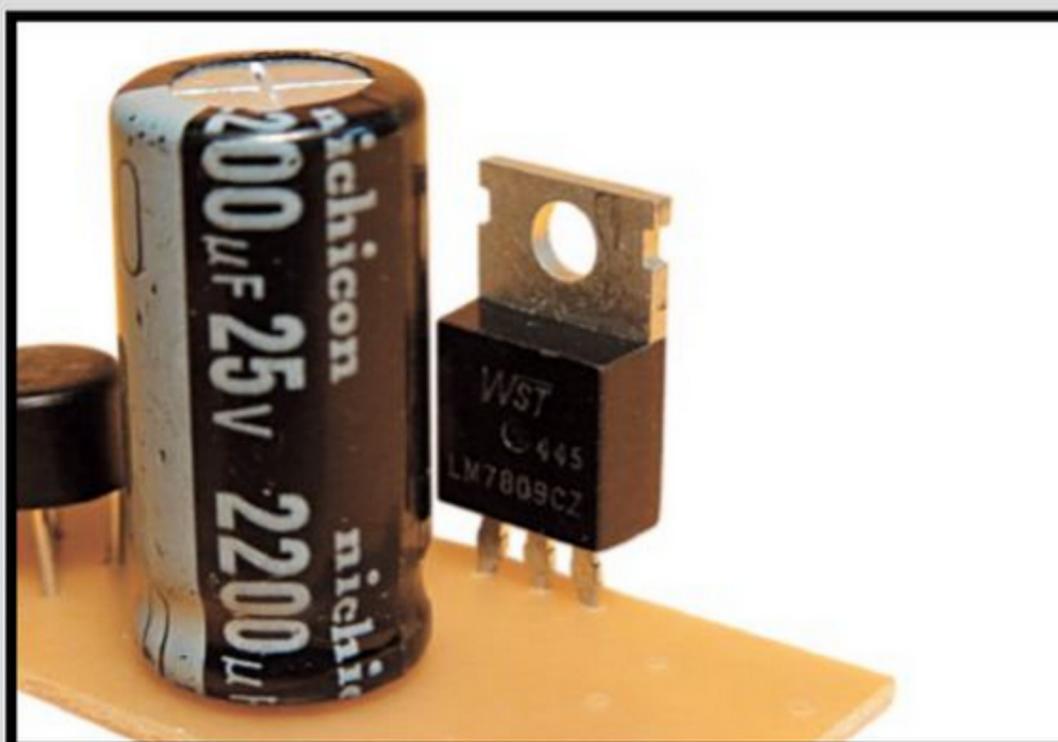
01 En el circuito que realice, plasmee en la placa PCB e interconecte los dispositivos entre sí. Una vez que realice el circuito en la placa PCB, comience a soldar los componentes. El primero es el puente diodo.



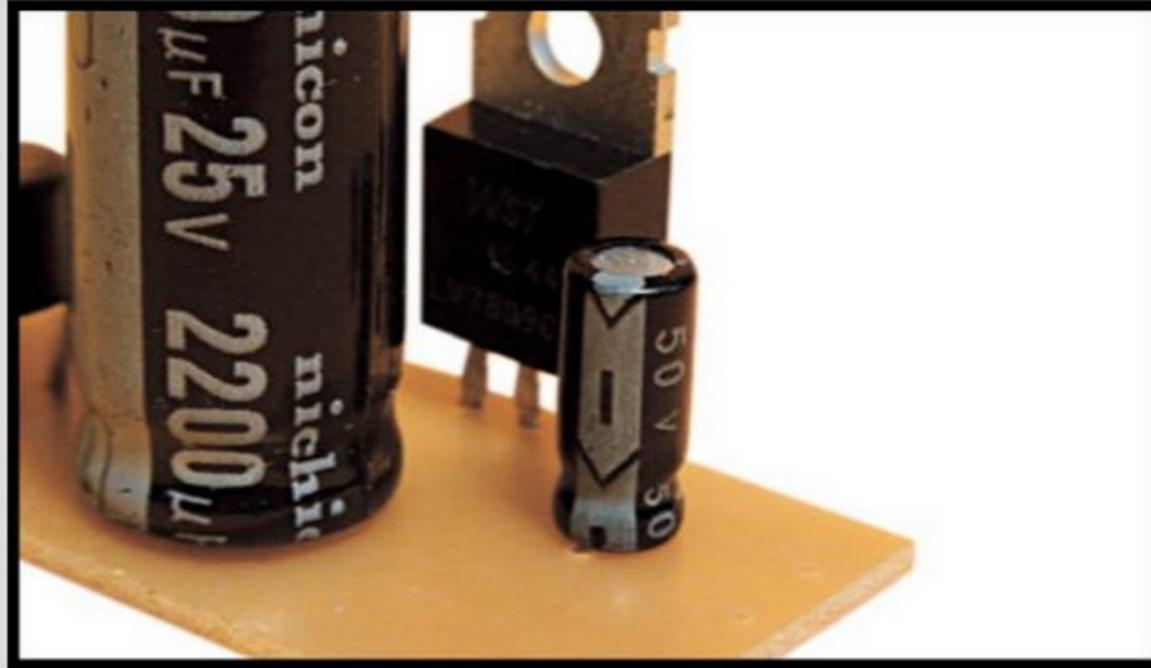
02 Junto al puente diodo, suelde el capacitor que será el encargado de rectificar la corriente alterna positiva que este entrega.



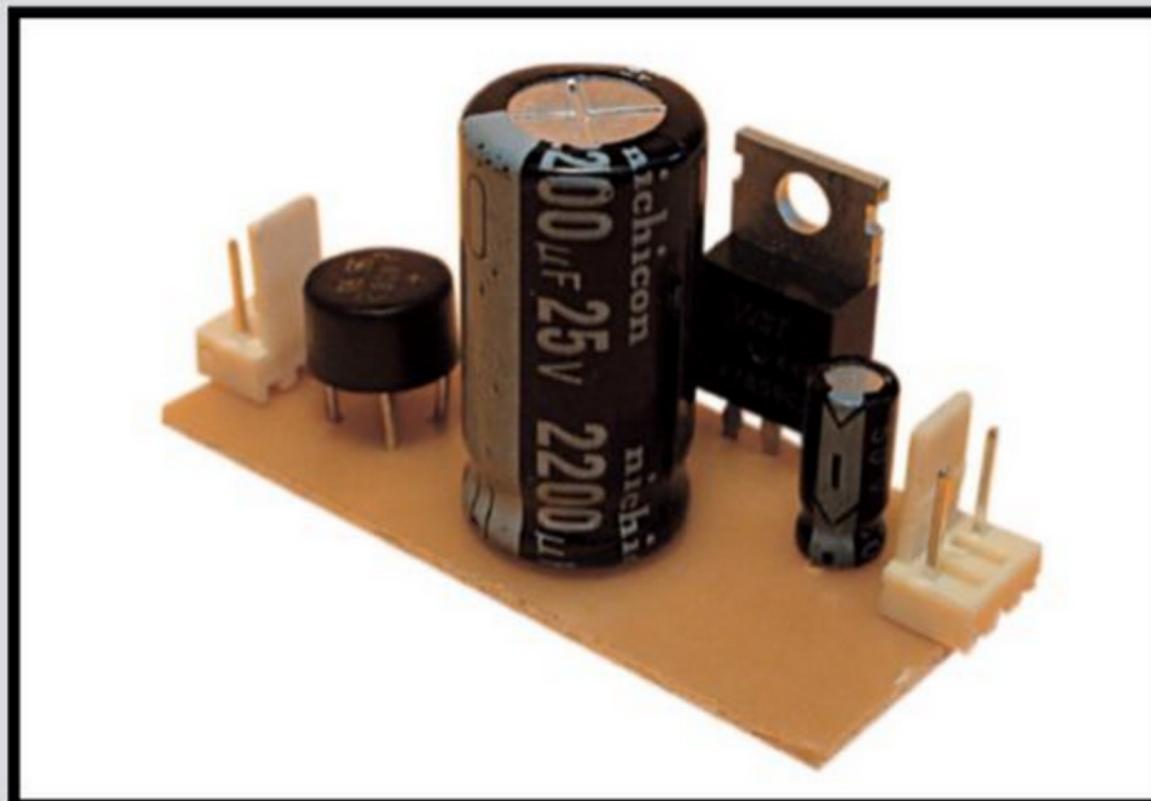
03 Luego, proceda a soldar el integrado LM7809 que será el que mantenga la salida de tensión constante a pesar de las variaciones del filtro.



- ▶
- 04** Luego, coloque el capacitor de salida que se encarga de evitar que se presenten rizados de corriente a la salida.



- 05** Por último, coloque los conectores que permiten conectar la fuente al transformador y al circuito que se alimentará.



Si bien en el Paso a paso anterior hemos realizado el diseño de una fuente de alimentación para 9 V y 1 A, el diseño se puede adaptar a otros valores. Para ello, basta con leer la hoja de datos del regulador de tensión que vayamos a utilizar, realizar los cálculos y obtener el valor de los capacitores y demás componentes, incluido el transformador de la entrada.

La corriente máxima que entrega esta fuente es de 1 A, pero, debido a que no tiene disipador, no recomendamos llevarla a ese nivel de exigencia. Si se necesita que la fuente entregue toda esa corriente, debemos dotar al regulador de un disipador. Los disipadores comerciales vienen tipificados por su resistencia térmica con un valor dado en °C/W. Si no conseguimos disipadores en el mercado, siempre podemos realizarlos reciclando una chapa de aluminio. Es imprescindible que el disipador esté en contacto con el aire libre para poder realizar su tarea. Si, por algún motivo, el disipador debe quedar dentro del gabinete de la fuente de alimentación, o si el disipador no es lo suficientemente grande e igual se eleva su temperatura, esta solución no basta.

En tal caso, debemos recurrir a otra solución. Lo más correcto y práctico es dotar a la fuente de alimentación de un forzador de aire. Así, al forzar la circulación de aire por el disipador de calor, el intercambio de energía se hará más rápido y eficiente. El ventilador puede ser conectado a la misma salida de la fuente, debido a su bajo consumo, y esto también nos permite asegurarnos que se encenderá en el momento en que se encienda la fuente. Al momento de adquirir el ventilador, debemos asegurarnos de que sea del voltaje adecuado.



Reguladores de voltaje

Las fuentes de alimentación sencillas suelen estar conformadas por cuatro etapas: **transformación**, **rectificación**, **filtrado** y **regulación**. Esta última etapa es imprescindible, ya que la salida del filtrado no suele ser una tensión continua perfecta y varía en función de la corriente de la carga y de la alimentación de la red. También podemos encontrar a la salida de la etapa de filtrado que la tensión posee un rizado. Para eliminar todos estos inconvenientes, utilizamos los reguladores de voltaje.

La función que le daremos al regulador de tensión será la de entregar una tensión fija y bien definida para así poder alimentar a nuestros circuitos de la forma correcta. Los reguladores de voltaje son también los que entregarán la corriente de salida, por lo que deben ser capaces de soportar desde algunas decenas de miliamperios hasta decenas de amperios.

Encontraremos muchos circuitos integrados que funcionan como reguladores de tensión. En el integrado, dan la referencia de tensión, un dispositivo de control, circuito de protección y amplificación. La tensión de salida puede ser fija o variable mediante potenciómetros. Los voltajes fijos pueden, a su vez, ser positivos o negativos.

En los reguladores de tensión fijos típicos tenemos 3 terminales. Un terminal tiene la función de entrada (IN); otro, la salida regulada (OUT); otro es el terminal de tierra (COMMON). La tensión de salida constante que nos proporciona está fijada en +5 V, +15 V o -15 V. La línea más difundida es la LM78XX para las tensiones positivas, y la línea LM79XX para voltajes negativos. En ambos casos, las últimas cifras marcadas con XX indican el voltaje regulado de salida. Así, encontraremos dentro de esta línea reguladores para 05, 06, 08, 09, 12, 15, 18, 24 y 30 V. Los encapsulados que podemos encontrar dependen de la corriente que otorguen: plásticos para las más bajas y completamente metálicos para los que superan 1 A.



Figura 18. Cuando la potencia que debe disipar el regulador es muy grande, este viene en un encapsulado que permite montarse directamente sobre los disipadores.

Los reguladores de tensión variables nos permiten modificar el valor de la tensión de salida. Estos cambian la función del terminal de tierra, manteniendo los tres terminales. En estos reguladores, tenemos el terminal de entrada, el de salida y el de regulación (**ADJ**). Para modificar la tensión de salida se conecta un potenciómetro al regulador. Se coloca el punto medio de este en el terminal ADJ; uno de los extremos, a la salida regulada de tensión; y el otro, a la masa del circuito. Los más populares de estos dispositivos son el integrado LM317 y el LM337. El primero entrega tensiones positivas y el segundo, negativas, con una corriente máxima de 1,5 A. La tensión de salida la podemos hacer variar entre 1,25 y 30 V.

UN REGULADOR
DE TENSIÓN
ASEGURA UNA
ALIMENTACIÓN
CONSTANTE

El mercado también nos ofrece reguladores de tensión duales. Esto es: tenemos, en un mismo circuito integrado, dos tensiones idénticas, pero de polaridad opuesta. Este integrado tiene dos terminales de entrada (uno para la tensión positiva que deseemos regular y otra para la negativa), un terminal de masa y los dos terminales de salida, uno para la tensión regulada positiva y otro para la negativa. Un ejemplo es el RC4195 que nos entrega ± 15 V y 100 mA. También podemos conseguir el RC4194, que es la versión variable de este regulador de voltaje dual.

Al momento de diseñar el circuito de una fuente de alimentación basada en un regulador de tensión específico, o al momento de comprar un regulador para la fuente de alimentación ya existente, debemos elegir el regulador basándonos, en primer lugar, en el voltaje por regular. Pero esto no es todo. Los fabricantes nos facilitan muchos otros parámetros



¿TE RESULTA ÚTIL?

Lo que estás leyendo es el fruto del trabajo de cientos de personas que ponen todo de sí para lograr un mejor producto. Utilizar versiones "pirata" desalienta la inversión y da lugar a publicaciones de menor calidad.

NO ATENTES CONTRA LA LECTURA. NO ATENTES CONTRA TI. COMPRA SÓLO PRODUCTOS ORIGINALES.

Nuestras publicaciones se comercializan en kioscos o puestos de voceadores; librerías; locales cerrados; supermercados e internet (usershop.redusers.com). Si tienes alguna duda, comentario o quieres saber más, puedes contactarnos por medio de usershop@redusers.com

HAY REGULADORES
DE VOLTAJE
PARA TENSIONES
POSITIVAS Y
NEGATIVAS

que nos ayudan a decidir por el regulador de tensión más adecuado para nuestra situación. Para esto, los fabricantes nos facilitan una serie de características presentes en la hoja de datos. A continuación, describiremos algunas de ellas.

Regulación de línea o *line regulation* es la medida de la capacidad que tiene el circuito para mantener la tensión de salida constante, aun cuando la tensión de entrada varíe.

Como la tensión de entrada de las fuentes de alimentación se obtiene a partir de la tensión de red alterna, esta tiene un rizado importante a la frecuencia de red, aun luego de la etapa de filtrado. Este rizado puede generar un mal funcionamiento o un daño irreparable en los dispositivos que conectemos a la fuente de alimentación, por lo que es imprescindible eliminarlo.

Regulación de carga o *load regulation*, nos indica la capacidad que tiene el circuito para mantener la tensión que entrega a la salida, aunque cambie la corriente que le demanda la carga. En una fuente de alimentación ideal, la tensión de salida debería ser independiente de esta corriente de carga. Esta característica está estrechamente relacionada con la resistencia equivalente de salida del circuito.

También podemos encontrar, en la hoja de datos, la **tensión de referencia** o *reference voltage*, que nos indica la tensión que utiliza el regulador para ajustar la tensión de salida.

En el caso de los reguladores de tensión variables también contamos con el dato de **corriente de ajuste** o *adjustment pin current*, que es la corriente que el circuito integrado presenta a la salida del pin *adjustment* y que debemos utilizar para calcular el rango de la resistencia variable que utilizaremos para el rango de tensiones de salida deseado.

La **corriente de salida mínima** o *minimum output current* indica la corriente mínima que debe estar presente en el terminal de salida para asegurar el correcto funcionamiento del regulador de voltaje.

Otra característica que debemos tener en cuenta es la **corriente máxima de salida** o *current limit* que nos proporciona el integrado, ya que, si sobrepasamos este valor, se activará el circuito de protección y el regulador dejará de entregar tensión. Estos tipos de circuitos integrados necesitan que la amplitud de la tensión presente en su terminal de entrada sea superior a la amplitud de la tensión de salida regulada.



Figura 19. Los disipadores de calor de distintas formas y colores nos permiten personalizar también este aspecto de nuestra fuente de alimentación.



Figura 20. Al colocarle un disipador de calor al regulador de tensión, permitimos que trabaje de forma adecuada y durante más tiempo.

El valor de cuánto mayor debe ser está indicado en la hoja de datos en la característica tensión de dropout. Si la diferencia de tensión entre la entrada y la salida es mayor que esta, se asegura la regulación de la tensión de salida. El regulador LM7805 tiene, por ejemplo, un dropout

LA SALIDA DE
LOS REGULADORES
DE TENSIÓN PUEDE
MANTENERSE FIJA
O VARIAR



de 2.5 V. Esto quiere decir que la tensión de entrada tiene que ser, como mínimo, de 7.5 V para asegurarnos una salida de 5 V.

En el mercado, encontramos otros reguladores que poseen voltajes de dropout de tan solo 0.6 V, como el LM2931. Como cualquier otro dispositivo electrónico, los reguladores de voltaje también tienen una tensión de entrada máxima, la cual no debemos superar para evitar daños o destrucción en el integrado. Este voltaje viene indicado en la

hoja de datos como **tensión máxima diferencial entrada salida** o *input-output voltage differential*. En el regulador LM7805, este valor es de 25 V, por lo que la tensión máxima de entrada no debe exceder los 30 V para que el dispositivo funcione correctamente durante mucho tiempo.

Como mencionamos anteriormente, dentro de la fuente de alimentación se producirá calor. El regulador de tensión es el mayor responsable de esa generación de calor. Es por eso que encontraremos que algunos reguladores de voltaje, si bien tienen su encapsulado en plástico, traen en la parte posterior una superficie metálica. Esta superficie está en contacto con el terminal de tierra y suele venir con una perforación.

La intención de esto es usar el orificio para que, mediante una tuerca y un tornillo, podamos adherir al regulador un cuerpo disipador. Los más comunes son aquellos realizados en aluminio. De esta forma, el disipador se mantiene siempre en contacto con el regulador de voltaje y ayuda a evitar un sobrecalentamiento del dispositivo, lo que genera un daño en él. El uso de estos disipadores es imprescindible para un correcto y duradero funcionamiento de la fuente de alimentación y nos permite prescindir de coolers, en algunos casos.



CÁLCULO DE COMPONENTES



Para diseñar una fuente regulada, debemos determinar el valor de los componentes que lleva. Podemos utilizar las fórmulas que el fabricante entrega en la hoja de datos o utilizar el software **Regulator Design** (www.diyaudio.com/forums/power-supplies/160699-regulator-design-v1-3-a.html), que, de manera rápida, calcula el valor de los componentes.

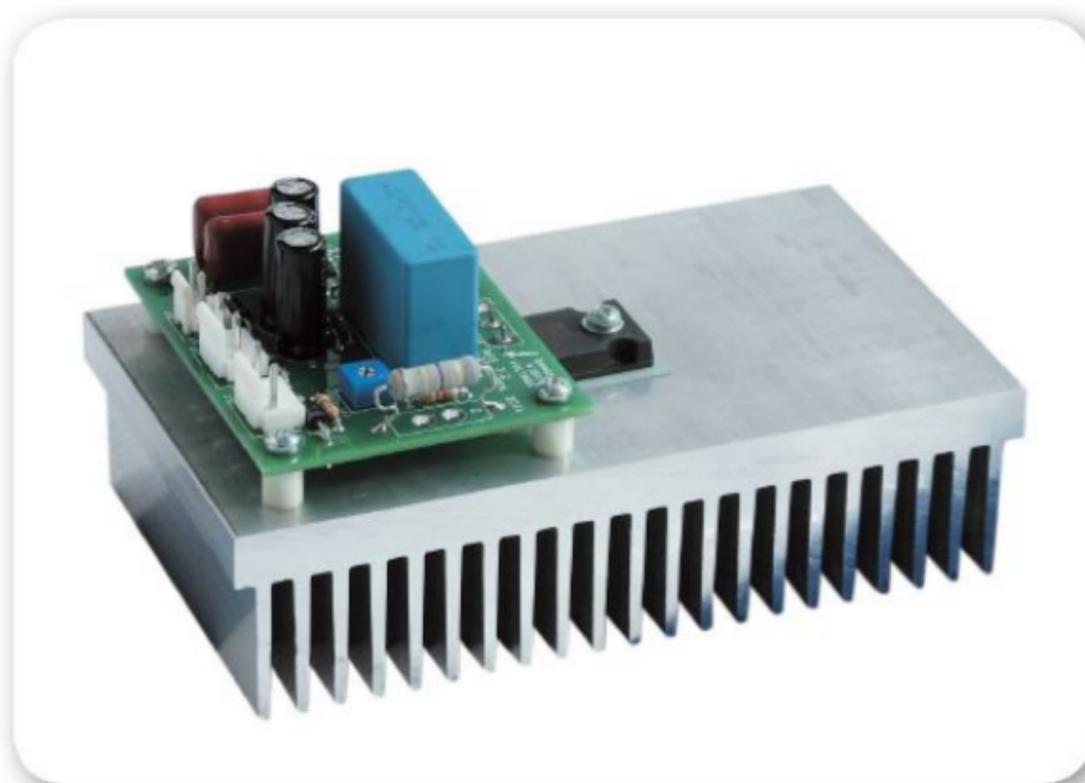


Figura 21. El tamaño del disipador de calor debe ser varias veces mayor que el del regulador de tensión para poder cumplir su tarea.

Generación alternativa de 220 V

Los **sistemas UPS** o **Sistema Ininterrumpible de Energía** (*Uninterruptible Power Supply*) se definen como un equipamiento o dispositivo encargado de proporcionar energía eléctrica por medio de baterías u otros almacenadores durante un período de tiempo y, durante un apagón, a todos los artefactos conectados a él. A estos artefactos los podemos denominar cargas críticas, por ser sensibles a fluctuaciones de tensión, de frecuencia o a la forma de onda del



TIEMPO DE RESPALDO

Se denomina tiempo de respaldo a los minutos durante los que el UPS brinda energía de la batería para un nivel dado de carga. Este tiempo es muy importante porque indica cuánto tiempo continuará operando el equipo durante un apagón. El agregar más equipo al UPS, disminuye el tiempo de respaldo. Se puede aumentar este tiempo comprando un UPS con una batería mayor o agregando módulos de batería externa.

suministro, como, por ejemplo, computadoras, equipamiento de comunicaciones y de laboratorio, entre otros. Los UPS actúan también como reguladores de tensión, mejorando la calidad de la energía eléctrica y filtrando subidas y bajadas de tensión.



Figura 22. UPS que ofrece un amplio margen de voltaje de entrada que reduce el número de descargas de la batería y prolonga su período de vida útil.

Los sistemas de informática toleran la mayoría de las interrupciones eléctricas; sin embargo, muchas veces pueden producir la pérdida de datos y daños materiales. El UPS contribuye a la disminución del voltaje, eliminando picos que sobrepasan ciertos niveles. Cuando se producen cortes de electricidad, la energía almacenada en la batería de emergencia mantiene la fuente de alimentación, suministrando electricidad a los equipos durante un corto período, de 5 a 10 minutos. Más allá de la autonomía que brinda el UPS, este tiempo ganado permite también que el equipo se conecte a otras fuentes de energía.

El UPS generalmente trabaja en las siguientes ocasiones:

- **Interferencias en la red eléctrica**, es decir, por un corte de electricidad de unos segundos que puede causar que el equipo informático se reinicie.
- **Sobrevoltaje**, se traduce en un valor nominal mayor que el máximo previsto para el funcionamiento normal de los aparatos eléctricos.

- **Picos de voltaje** o sobrevoltajes transitorios de amplitud alta que ocurren cuando se apagan y encienden dispositivos que demandan mucha alimentación y que, con el tiempo, pueden causar daños a los componentes eléctricos.
- **Descargas de rayos**, que constituyen una fuente de sobrevoltaje extremo que se produce durante las tormentas.



Figura 23. Los UPS pueden ser utilizados para asegurar la alimentación en líneas de copia de seguridad.

En cuanto a los tipos de UPS, podemos señalar varios, cada uno con distintas características de desempeño:

- **UPS standby**: es el tipo más usado para computadoras personales. El switch de transferencia está regulado para elegir entre la entrada *CA* (*Alternating Current* o corriente alterna) filtrada, que es la fuente de energía primaria, y alternar con la batería, siempre que la fuente de energía primaria falle. Los principales beneficios de este diseño son la alta eficiencia, el pequeño tamaño y el bajo costo. Con un filtro apropiado y un circuito eléctrico de sobretensión de corriente, estos sistemas pueden proveer una filtración de ruido adecuada y supresión de sobretensión.
- **UPS corriente alterna**: obtiene a su salida una señal alterna, por lo que necesita un inversor para transformar la señal continua que proviene de las baterías en una señal alterna.

- **UPS corriente continua:** las cargas conectadas al UPS requieren una alimentación de corriente continua; por lo tanto, este transformará la corriente alterna de la red en corriente continua y la usará para alimentar la carga y almacenarla en sus baterías. Por tanto, no requiere convertidores entre las baterías y las cargas.
- **UPS de línea interactiva:** el diseño más utilizado para los negocios pequeños. El convertidor de la batería a energía AC siempre está conectado a la salida del UPS. Cuando la entrada de energía falla, el interruptor se abre y la energía fluye desde la batería hacia la salida del UPS. Usualmente incorpora un transformador **tap-changing**, que agrega regulación de voltaje, ajustando las tomas de corriente de salida a medida que el voltaje de entrada varía. La regulación del voltaje es una característica importante cuando existen condiciones de bajo voltaje. Entre sus características principales, tenemos alta eficiencia, tamaño pequeño y alta confiabilidad.
- **UPS con volante inercial:** en este caso, la energía para mantener el suministro de alimentación eléctrica se conserva a través del denominado almacenamiento cinético. De esta manera, no se requiere el uso de baterías y, con ello, se evita la necesidad de mantenerlas o reemplazarlas.



Figura 24. Imagen de un modelo de UPS de alto rendimiento para redes de voz y de datos, laboratorios médicos y aplicaciones de industria.

Otro elemento para destacar son las baterías, de las cuales existen diversas variedades y se las llama, por lo general, de acuerdo con los materiales que se utilizan para su construcción. Existen de un solo uso, por ejemplo, de zinc, carbón, litio, alcalinas, etcétera. También las hay recargables, y, una vez que se descargan, pueden ser cargadas conectándoles un voltaje nominal de la batería. Ejemplo de ellas son las de níquel-cadmio, plomo ácido, alcalina recargable y de gel.

En general, las baterías de plomo ácido son una buena opción en términos de rentabilidad y confiabilidad, ya que las nuevas tecnologías aún no logran igualar su desempeño. Hoy en día, la tecnología más utilizada son los volantes de inercia que pueden proveer tiempo de respaldo suficiente a un UPS mientras arranca un generador. Una de las ventajas principales es que no se tiene que reemplazar cada tres años, como sucede con las baterías de plomo ácido.

En la actualidad, la industria de los UPS ha ido evolucionando de manera paulatina para seguir incluyendo diseño, aunque prácticamente están llegando al límite de la eficiencia. Esto hace que los diferentes tipos de dispositivos se adecúen a diferentes aplicaciones, de modo que estos se incorporarán al mercado cada vez con mayor eficiencia y capacidad, teniendo en cuenta sus beneficios y desventajas.



Figura 25. UPS que puede ser utilizado para computadoras personales. Se trata de un producto accesible y de tamaño reducido.



Figura 26. UPS de gran capacidad, que se utiliza para proteger al equipo durante cortes de corriente alterna y resguardar los datos de forma segura.

Generadores a combustible

Se trata de una máquina que, a través de un motor de combustión interna, mueve un generador eléctrico. Se utilizan, por lo general, cuando existe un déficit en la generación de energía eléctrica en algún lugar, o bien cuando son frecuentes los cortes en el suministro eléctrico.

Como dijimos, los generadores a combustible se usan comúnmente en aquellos lugares donde no hay suministro eléctrico, por lo general, en zonas aisladas, muy poco habitadas y con poca infraestructura.

La elección de un **generador** depende principalmente del gasto de energía de respaldo que necesitemos para una aplicación específica,



PARA TENER EN CUENTA

Muchas marcas ofrecen software para cerrar las aplicaciones ante un corte de energía, junto a servicios como supervisión y monitoreo de tensión de entrada, estado de la batería y envío de e-mails para informar lo que sucede ante un incidente. Estos datos, sumados a la calidad de onda senoidal a la salida del dispositivo, son de gran ayuda a la hora de elegir y adquirir un UPS.

o bien, de la energía de reserva para garantizar el funcionamiento ininterrumpido de los aparatos o equipos de misión crítica.



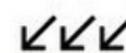
Figura 27. Un grupo electrógeno portátil es útil para llevarlo donde deseemos. Se ha vuelto una pieza muy importante en actividades recreativas.

Para profundizar un poco más, mencionaremos las partes más importantes de un generador a combustible:

- **Motor a combustión:** representa la fuente de energía mecánica para que el alternador gire y genere electricidad. Existen dos tipos de motores: motores a nafta y a gasoil (diésel). Generalmente, los motores diésel son los más utilizados en los grupos electrógenos por sus prestaciones mecánicas, ecológicas y económicas.



TIPOS DE POTENCIA



Los diferentes tipos de potencia describen las condiciones máximas permisibles de carga de un generador. Existen tres tipos: la potencia **standby**, para la provisión de energía de emergencia; la potencia **prime**, que se aplica cuando se suministra energía en lugar de la energía comprada; y la potencia **carga base**, que opera en paralelo con la red, bajo cargas constantes por largos periodos de tiempo.

- **Regulación del motor**: es un dispositivo mecánico diseñado para mantener una velocidad constante del motor con relación a los requisitos de carga. La velocidad del motor está directamente relacionada con la frecuencia de salida del alternador, por lo que cualquier variación de la velocidad del motor afectará a la frecuencia de la potencia de salida. Esto permite que el motor a combustión no trabaje en exceso cuando la carga de salida no es alta.
- **Sistema de refrigeración**: consta de un sistema por medio de agua, aceite o aire. El sistema de refrigeración por aire consiste en un ventilador de gran capacidad que hace pasar aire frío a lo largo del motor para enfriarlo (tipo electroventilador de automóvil). El sistema de refrigeración por agua o aceite consta de un radiador y un ventilador interior para enfriar sus propios componentes. Este último se utiliza en generadores de muy alta potencia que requieren una refrigeración más eficiente.
- **Alternador (generador)**: es el generador de electricidad en sí, protegido contra salpicaduras, autoexcitado, autorregulado y sin escobillas, acoplado con precisión al motor. Se elige la opción sin escobillas para evitar el rápido desgaste del motor, ya que se supone que es un dispositivo de uso continuo y no se pueden permitir fallas.
- **Aislamiento de la vibración**: el grupo electrógeno está dotado de tacos antivibrantes, diseñados para reducir las vibraciones transmitidas por el grupo motor-alternador. Estos aisladores están colocados entre la base del motor, del alternador, del cuadro de mando y la bancada.
- **Interruptor automático de salida**: para proteger al alternador, se suministra un interruptor automático de salida adecuado para el modelo y régimen de salida del grupo electrógeno con control manual. Para grupos electrógenos con control automático, se protege el alternador mediante contactores adecuados para el modelo correspondiente y el régimen de salida.

La utilización de los generadores por combustión varía sustancialmente. Para cada situación es recomendable un cierto tipo de generador. Básicamente, utilizan grupos electrógenos desde personas que necesitan luz en una noche de pesca a la intemperie

hasta en un sistema de emergencia en un hospital de alta complejidad. Lo que más importa al momento de seleccionar un generador a combustible es su estabilidad y su potencia de salida. Es muy común ver grupos electrógenos acoplados a baterías; esto permite tener más respaldo eléctrico en caso de una emergencia, así como también más estabilidad. Como conclusión, podemos decir que los grupos electrógenos se han vuelto una pieza fundamental en lugares que requieren un suministro eléctrico ininterrumpido.



Figura 28. A la hora de elegir un generador, debemos tener en cuenta sus características eléctricas, como la potencia, el número de fases y la frecuencia.



RESUMEN

En este capítulo, estudiamos las fuentes de alimentación, realizando una primera diferenciación entre lineales y conmutadas. Vimos las distintas etapas que conforman el circuito, para lograr la entrega de energía de manera regulada y estabilizada respecto de los parámetros que pueden producir dispersión. Conocimos también los circuitos y los dispositivos reguladores de voltaje y encaramos un proyecto de armado de una fuente completa para que pueda ser aprovechada y utilizada en el laboratorio. Por último, conocimos la generación alternativa de energía eléctrica, que es una temática de gran actualidad, ya que prácticamente todo lo que utilizamos en la vida cotidiana y en los ambientes industriales y de empresas depende de la electricidad, especialmente lo relativo a las computadoras. Para esto, vimos los sistemas de alimentación ininterrumpida (UPS) y los generadores eléctricos que funcionan a combustible.

Actividades

TEST DE AUTOEVALUACIÓN

- 1 ¿Cuáles son las clasificaciones posibles de una **fente de alimentación**?
- 2 ¿De qué depende la **tensión de salida** en el diseño de un circuito?
- 3 ¿Qué es una **fente conmutada**, en qué tipo de circunstancias es recomendable utilizarla y qué tipo de configuraciones acepta?
- 4 ¿Qué cuestiones debemos tener en cuenta a la hora de **diseñar una fente**?
- 5 ¿Cómo se denomina el dispositivo encargado de **proporcionar energía eléctrica** mediante baterías u otros almacenadores durante un período de tiempo?



PROFESOR EN LÍNEA



Si tiene alguna consulta técnica relacionada con el contenido, puede contactarse con nuestros expertos: profesor@redusers.com



Las fuentes de alimentación y la robótica

En este último capítulo, explicaremos las nociones básicas de la robótica, las diferencias entre los distintos tipos de robots y las fuentes de alimentación que se le aplican.

▼ Anatomía de los robots302	▼ Dispositivos especiales311
Tipos y características	Grados de libertad 311
de los robots 304	Zona y volumen de trabajo..... 311
▼ Componentes de un robot.....305	Precisión de los movimientos 312
El controlador..... 306	Configuraciones y morfología 313
▼ Baterías308	▼ Fuentes de alimentación317
Plomo y ácido..... 308	Microcontroladores aplicados
Níquel-cadmio 310	a la robótica..... 318
Alcalinas..... 310	▼ Resumen.....319
Níquel-hidruro metálico..... 310	▼ Actividades.....320
Litio-ion..... 310	



Anatomía de los robots

Cuando escuchamos la palabra **robot**, muchas veces pensamos en esas películas que nos han sorprendido por ostentar máquinas que realizan acciones similares o superiores a las capacidades del ser humano. Desde hace muchos años, los humanos han tratado de reproducir el comportamiento y la forma de los seres vivos, creando distintos modelos de robots y dotándolos de sistemas cada vez más avanzados.

Cada robot presenta características únicas respecto a su forma, fuerza y tipos de desplazamiento, que le confieren la habilidad de desarrollar tareas en forma muy eficiente, superando ampliamente al hombre o reemplazándolo en lugares peligrosos. A lo largo de este capítulo, veremos los componentes esenciales de los robots, así como los parámetros que caracterizan su aplicación en diferentes áreas.

En 1923, Karel Capek escribió la obra *Rossum's Universal Robot* (R.U.R.), en donde surgió por primera vez la palabra robot, la cual deriva de **robotnik**, que en checo significa “siervo”. A través de los años, se han construido muchos aparatos mecánicos; Leonardo da Vinci dedicó parte de su vida a desarrollar un león animado, aves voladoras, vehículos autopropulsados y tantos otros. Los robots nacieron de la necesidad y del progreso tecnológico recién en 1960, cuando George Devol, considerado padre de la robótica, desarrolló el primer robot industrial.

Estos aparatos pueden variar en cuanto a su forma, pero todos tienen una característica en común: la inteligencia necesaria para realizar tareas en diversos entornos, entendiéndose por inteligencia a la habilidad que poseen para resolver un problema.



MÚLTIPLES DISCIPLINAS



Podemos decir que la robótica demanda a los diseñadores el conocimiento de diversas áreas para su desarrollo e implementación, como electrónica, sistemas computacionales, mecánica y análisis de la relación costo-beneficio. En la actualidad, se ha popularizado el empleo y la fabricación de microrobots, lo que permitirá la expansión de la robótica en hogares e industrias en el corto plazo.

El Instituto Norteamericano de Robótica define al robot industrial como un manipulador multifuncional y reprogramable, diseñado para desplazar materiales, piezas, herramientas o dispositivos especiales, mediante movimientos programables y variables que permiten llevar a cabo diversas tareas.

Los robots son muy utilizados en tareas en las que se requiere alta exactitud, abaratamiento de costos, ensamblaje, embalaje u otras que resultan peligrosas, repetitivas o, incluso, sucias para las personas. Por tal motivo, existe la tendencia a buscar formas cada vez más humanoides para sus estructuras y, así, llegar a reemplazar al ser humano.

Figura 1. Mano robot con músculos neumáticos emulando los movimientos humanos. Posee sensores de fuerza para tomar objetos delicados.



Otros impulsores importantes de los robots son las aplicaciones militares. Estas fomentan el desarrollo de robots cada día más complejos, que demandan un avance tecnológico a veces solo superado por la ficción.



Figura 2. Robot soldador de estructura vertical con seis ejes. Tiene extremidades que le otorgan un rango extendido. Utiliza servomotores de corriente alterna para mover sus extremidades.

Tipos y características de los robots

Enumeraremos, a continuación, los distintos tipos de robots y las características más sobresalientes que se han alcanzado en los últimos años. En la actualidad, existe una amplia variedad de clasificaciones de robots, entre las que podemos encontrar: aviones autónomos, segway, mayordomos, vehículos terrestres autónomos, humanoides, colaborativos, tipo Rover, bípedos, ápodos, polimórficos, con forma de animales, entre otros. Según sus diferentes capacidades, los robots se clasifican en:

- **Androides bípedos**: son robots que parecen y actúan como los humanos. Tratan de imitar el comportamiento del hombre. Actualmente, se utilizan solo para experimentación. En este modelo, la principal limitante es la implementación del equilibrio a la hora del desplazamiento.
- **Zoomórficos**: su sistema de locomoción imita al de los animales. Su aplicación está vinculada al estudio de terrenos accidentados, volcanes y exploración de planetas, ya que son hábiles para sortear obstáculos. Pueden subclasificarse en caminadores y no caminadores; estos últimos son muy poco evolucionados.
- **Móviles**: poseen una gran capacidad de desplazamiento y un alto nivel de inteligencia. El desplazamiento se logra mediante plataformas o carros con sistemas de locomoción rodantes. Tienen gran aplicación en líneas de montaje industriales, donde hay que asegurar el transporte de piezas de un punto a otro. Por lo general, son guiados por telemando o por la información recibida del entorno mediante sensores.
- **Poliarticulados**: se caracterizan por agrupar robots primordialmente sedentarios, que poseen un número reducido



LEYES DE LA ROBÓTICA



En 1942, **Isaac Asimov** enunció en sus novelas de ciencia ficción tres leyes aplicables a la robótica que, con el tiempo, se convirtieron en cuatro y representan el código moral que deberían seguir. En la actualidad, el proyecto Physical Human-Robot Interaction Dependability and Safety, financiado por la Unión Europea, pretende forzar a los robots a cumplirlas.

de articulaciones para mover sus elementos terminales. Su utilidad principal está relacionada con el área industrial, donde se encargan de desplazar elementos que requieren cuidados especiales.



Figura 3. Robot de juguete de 16 servomotores, controlado en forma remota. Puede caminar, girar el torso y la cabeza y tomar objetos con sus garras.



Componentes de un robot

El manipulador, mecánicamente, es el componente principal que dota al robot de sus movimientos y define lo que llamamos la **zona de trabajo**. Está formado por una serie de elementos estructurales sólidos, unidos mediante articulaciones que permiten un movimiento relativo entre dos piezas consecutivas. Cada articulación le da un mayor grado de libertad.

La mayoría de los robots guarda similitud con la anatomía de las extremidades superiores del hombre. Para hacer referencia a los elementos que componen al robot, muchas veces se usan términos como cintura, hombro, brazo, codo, muñeca y antebrazo.

Las articulaciones utilizadas en un manipulador pueden ser:

EL MANIPULADOR
ES EL COMPONENTE
MECÁNICO MÁS
IMPORTANTE
DE UN ROBOT

- **Lineales:** los elementos estructurales se apoyan sobre un deslizador lineal, y se desplazan a lo largo del recorrido mediante tornillos sin fin o cilindros.
- **Rotacionales:** conformadas por una unión giratoria que, a menudo, es manejada por motores eléctricos con transmisiones.
- **Por cilindros hidráulicos y uso de palancas** .



El controlador

Es el que maneja cada uno de los movimientos del manipulador, las acciones, cálculos de posición y procesamiento de la información. Está continuamente enviando y recibiendo información de otros componentes que interactúan con él o que trabajan conformando un equipo.

En función de los parámetros que se regulan, existen varios tipos de controladores:

- **De posición** : se encargan de controlar la posición del elemento terminal.
- **Cinemáticos:** actúan sobre la posición y la velocidad.
- **Dinámicos:** además de actuar sobre la regulación de la velocidad y la posición, controlan las propiedades dinámicas del manipulador y de los elementos asociados.
- **Adaptativos** : engloban todas las regulaciones anteriores y se ocupan de controlar la variación de las características del manipulador al variar la posición.



DISPOSITIVOS DE ENTRADA/SALIDA



Los periféricos E/S son un conjunto de dispositivos pertenecientes a un sistema de procesamiento, que se utilizan para comunicarse con otros sistemas. Una entrada es una señal recibida por la unidad, mientras que las salidas son las que esta envía. Los dispositivos de E/S pueden ser utilizados por otro sistema o por una persona para comunicarse con una computadora. Teclados y mouses son considerados dispositivos de entrada, mientras que monitores e impresoras son de salida.



Figura 4. Robot con orugas destinado a la educación y la exploración. Está controlado a través de internet; además, posee cámara web, medidor láser y oruga de cuatro motores.

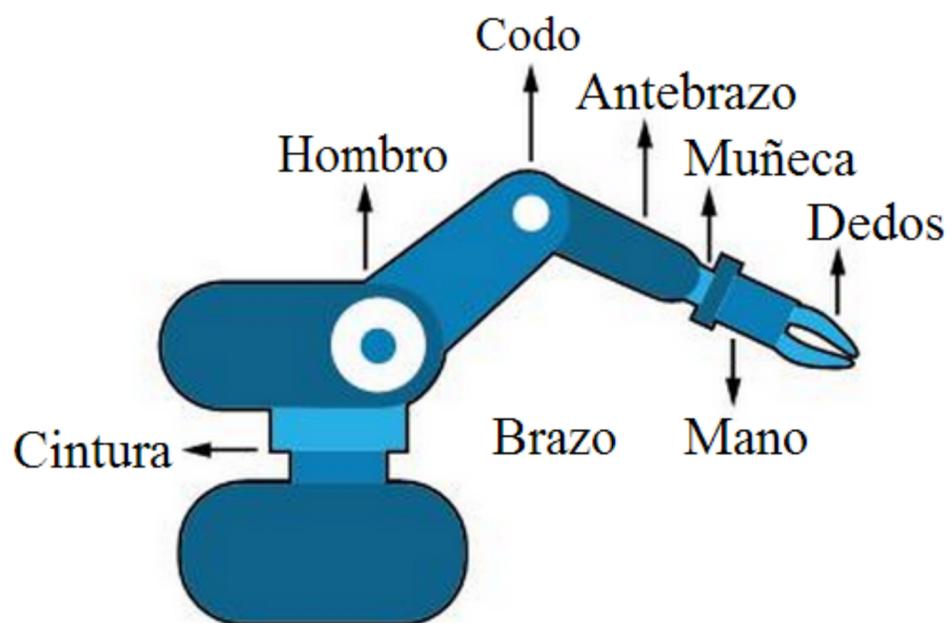


Figura 5. En esta imagen, podemos observar la similitud existente entre la estructura humana y la de un robot compuesto por seis ejes.

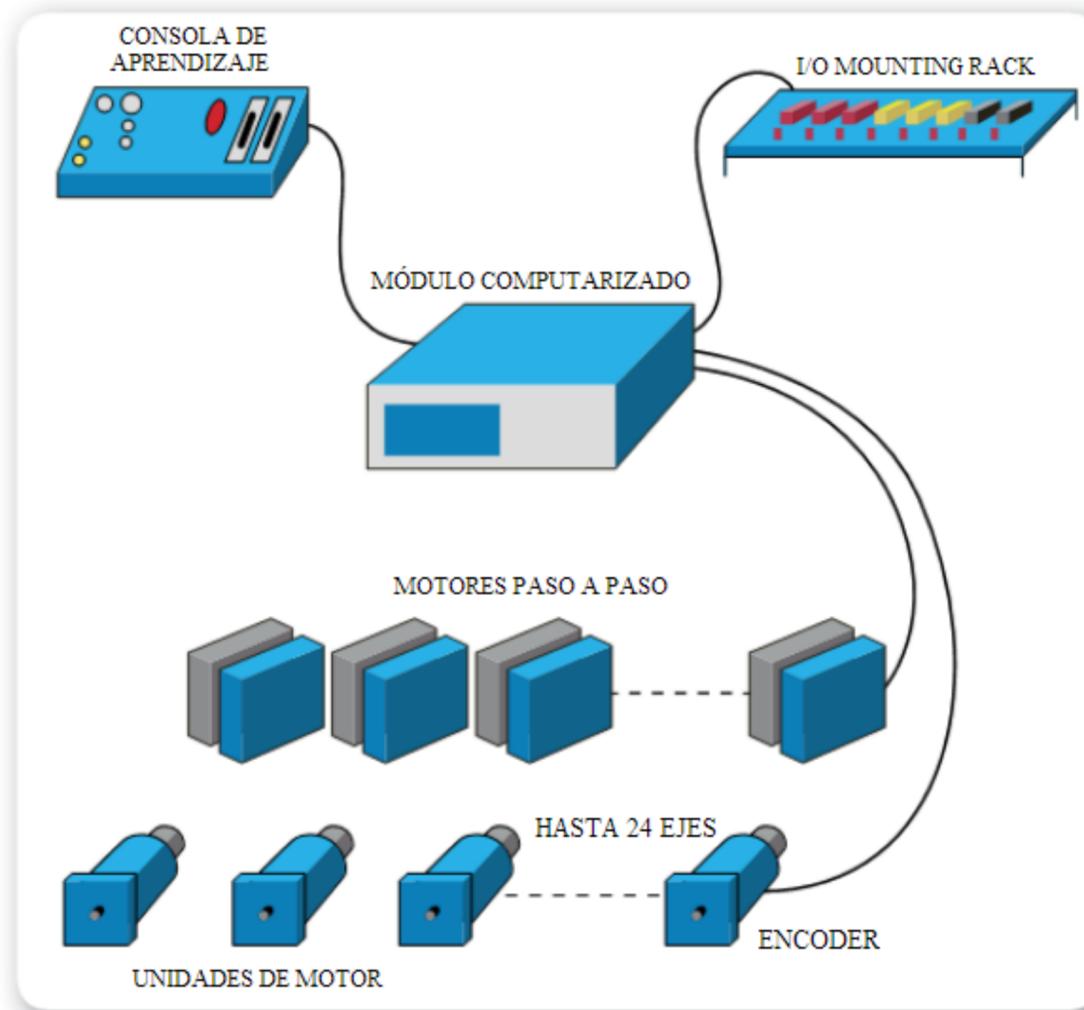


Figura 6. Esquema de las partes que hacen al control de un robot, como drivers, motores y placas de entrada-salida.

Baterías

Las baterías son dispositivos que emplean procedimientos electroquímicos para almacenar energía eléctrica en forma química. Cuando se conecta un circuito eléctrico, la energía química se convierte en eléctrica. Todas las baterías están compuestas por un número de celdas, constituidas en su interior por un electrodo positivo (ánodo) y uno negativo (cátodo). A continuación, enumeraremos las más utilizadas.

Plomo y ácido

Son las más comunes y producen energía a partir de que el plomo se sumerge en ácido sulfúrico (electrolito). Proveen un voltaje de **2 voltios** por elemento y pueden ser recargadas mediante un proceso lento. La característica más importante es su gran capacidad de entregar corriente, lo que las vuelve ideales para utilizar en motores de arranque.

Cada elemento contiene ácido sulfúrico y una serie de placas de plomo, dispuestas alternadamente entre positivas y negativas, separadas por un aislante resistente al ácido. El número de placas y su espesor determinan la cantidad de corriente que entrega cada elemento. Debajo de las placas hay un espacio en el cual se depositan los desprendimientos.

Existen otros tipos de baterías, como las de **gel**, donde al electrolito se le agrega un compuesto gelificante para reducir el movimiento dentro de ellas. Poseen una descarga muy baja durante su almacenamiento, y permiten un mayor número de ciclos de carga y descarga. Por su parte, las **baterías de ciclo profundo** cuentan con placas muy gruesas y poco porosas que otorgan una vida útil superior. Suelen usarse en instalaciones de energía solar como método de almacenamiento o como fuentes de energía para las UPS.



Figura 7. En esta imagen vemos un ejemplo de baterías recargables.



DOS TECNOLOGÍAS



Existen dos versiones conocidas en las baterías de plomo y ácido: las de **bajo** mantenimiento y las de **libre** mantenimiento. Las primeras necesitan un mínimo control (por ejemplo, el nivel de ácido), mientras que las otras no requieren ningún cuidado especial. Por lo general, son totalmente selladas, ya que reaprovechan el ácido que se evapora.

Níquel-cadmio

Son las pilas recargables más comunes y proporcionan 1,2 V por elemento. Contienen, en su interior, un cadmio negativo y el hidróxido de níquel positivo, separados por un electrolito de hidróxido de potasa. El cadmio es peligroso desde es punto de vista ecológico. Un inconveniente que presentan estas pilas es el efecto memoria, lo que hace necesario agotarlas en su totalidad antes de recargarlas.

Alcalinas

Utilizan mercurio y fueron muy populares en un momento. Se retiraron de la producción por ser muy contaminantes, ya que poseían un 30 % de mercurio y, una vez utilizadas, se desechaban sin tomar ninguna precaución ambiental. Fueron sustituidas por las pilas de **zinc-carbono**, llamadas “comunes”, que contienen menos del 0,01 % de mercurio.

Las pilas de **manganeso** son más recientes que las anteriores. Están compuestas por **dióxido de manganeso**, **hidróxido de potasio** y **pasta de zinc** amalgamada con mercurio, carbón o grafito.

Una sola pila alcalina puede contaminar **175.000 litros** de agua (más de lo que puede consumir un hombre en toda su vida). Las baterías alcalinas proveen un voltaje de 1,5 V por elemento, y no es posible recargarlas, por lo que es necesario disponer de ellas al final de su vida útil en forma especial, para evitar contaminaciones.

Níquel-hidruro metálico

Tienen un 40 % más de rendimiento que las de níquel-cadmio a igual cantidad de mA·H. No poseen efecto memoria y admiten alrededor de 700 ciclos de carga.

Litio-ion

Son muy caras de fabricar y proveen un ciclo de recarga de aproximadamente 600 ciclos, por lo que son utilizadas en aplicaciones de gran exigencia. Son más confiables y tienen mayor cantidad de energía que las de Ni/Cd o Ni/MH.

Dispositivos especiales

Al momento de seleccionar robots, existen características que los diferencian y que dependen estrechamente de la estructura seleccionada para la tarea en la que se los desea emplear.

Grados de libertad

Este parámetro se utiliza para determinar la posición y la orientación del elemento terminal del robot. Cada movimiento independiente que pueda realizar respecto del anterior se considera un grado de libertad. El número de grados de libertad que posee un robot está dado por la suma de todos los grados de libertad de las articulaciones que lo componen; por lo general, coincide con el número de articulaciones.

Para poder posicionar una pieza o herramienta en el espacio, se requieren de, al menos, **seis grados de libertad**: tres de posición y tres de orientación. Con un número más alto, se logra una mayor flexibilidad para posicionar el elemento terminal. Si la cantidad de grados de libertad es mayor a la requerida, decimos que el robot es **redundante**.

Zona y volumen de trabajo

La zona está definida por las dimensiones de los elementos del manipulador y los grados de libertad, y es fundamental para la selección del modelo de robot por utilizar. La zona de trabajo se subdivide en áreas diferenciadas entre sí por la accesibilidad específica del elemento terminal (aprehensor o herramienta). También queda restringida por los límites de giro y desplazamiento que existen en las articulaciones.



NUEVAS BATERÍAS DE LI-PO



Son baterías relativamente nuevas en el mercado. Su característica fundamental radica en que proveen una densidad de energía y una tasa de descarga muy altas. Se las emplea en aeromodelismo debido a su tamaño reducido y bajo peso: algunos modelos llegan a alcanzar relaciones de 7,2 kW/kg. El tiempo de recarga es muy bajo y brindan más de 5000 ciclos de uso.

Por otra parte, el **volumen de trabajo** del robot se refiere al espacio dentro del cual se puede desplazar el extremo de su muñeca. Para determinar este parámetro, no se debe tomar en cuenta el elemento terminal, ya que es posible adaptar **grippers** (pinzas) de distintos tamaños.



Figura 8. Vista de la zona vertical de trabajo de un robot donde podemos apreciarlos ángulos de rotación de cada articulación.

Precisión de los movimientos

Este importante concepto depende de varios factores: resolución, exactitud, repetibilidad, capacidad de carga, velocidad de movimiento y programabilidad. Veamos cada uno de ellos:

- **Resolución** : es el incremento de movimiento más pequeño en que el robot puede dividir su volumen de trabajo. Depende de los sistemas que controlan la resolución, ya que estos son los encargados de manejar los incrementos individuales de cada articulación y la inexactitud mecánica, ligada a la calidad de los componentes que conforman las uniones y las articulaciones.
- **Exactitud** : es la capacidad de un robot para situar su muñeca en un punto señalado dentro del volumen de trabajo. Mide la distancia entre la posición real del actuador y la especificada o deseada.

Un robot presenta una mayor exactitud cuando opera cerca de su base. A medida que el brazo se aleja, la exactitud irá decreciendo, debido a que las inexactitudes mecánicas se incrementan.

- **Peso de la carga** : al ser las cargas más pesadas, reducen la exactitud (incrementan las inexactitudes mecánicas). El peso también afecta la velocidad de los movimientos del brazo y la resistencia mecánica.
- **Repetibilidad** : es la capacidad de un robot de regresar a un punto programado todas las veces que sea necesario. Esta magnitud establece el grado de exactitud en la repetición de los movimientos de una tarea programada.
- **Capacidad de carga** : es el peso en kilogramos que puede transportar la garra de un robot. Este valor puede oscilar entre 1 kg y 200 kg para los robots industriales.
- **Velocidad de movimiento** : es el máximo valor que pueden alcanzar las articulaciones (rotación o desplazamiento lineal). Si utilizamos un valor alto, aumentamos el rendimiento, como en tareas de soldadura o manipulación de piezas.



Figura 9. Robot humanoide Robonaut 2, fabricado por la NASA y GM. Utiliza controles de vanguardia, sensores y tecnología de visión.

Configuraciones y morfología

Al hablar de configuración o morfología de un robot, hacemos referencia a la forma física que este posee.

El robot puede presentar cuatro configuraciones clásicas, que se relacionan con los modelos de coordenadas en el espacio: **cartesiana**, **cilíndrica**, **polar** o **angular**. También existe una configuración no clásica, conocida como **SCARA** (*Selective Compliance Assembly Robot Arm*):

Robot cartesiano : se caracteriza por tener movimientos lineales, pudiendo existir un **eje X**, dos ejes **XY** o tres ejes **XYZ**. Esto dota al robot de movimientos libres en el plano vertical y/u horizontal, o en el espacio.

Las herramientas de trabajo tales como las de agarre, soldadura o pegado (dependiendo de la aplicación o el proceso donde se utilicen) pueden adaptarse de manera sencilla a estos robots, y pueden gobernarse a través de los servoaccionamientos (drivers) de bajo costo.

Para realizar una trayectoria entre un punto y otro, el robot recurre a operaciones matemáticas. Calcula el trayecto punto a punto: el cálculo puede ser lineal, si se desplaza en línea recta, o por articulación, de acuerdo con el movimiento de la coyuntura.

A partir del acoplamiento de servomotores y controladores de posicionamiento con los sistemas de ejes, surgen robots cartesianos precisos y de gran rendimiento, utilizados para la realización de tareas de posicionamiento. Esto ofrece diferentes formas para la construcción de robots.

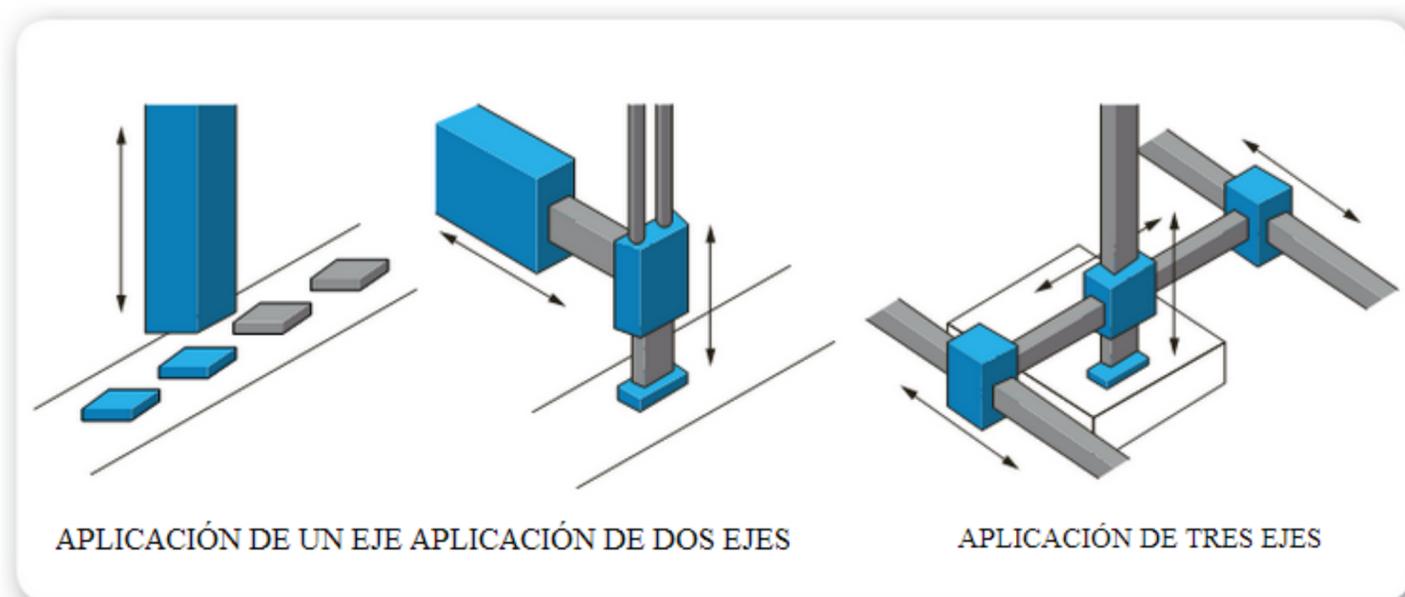


Figura 10. En la figura podemos ver que, para tareas más complejas, es necesario utilizar una mayor cantidad de ejes.

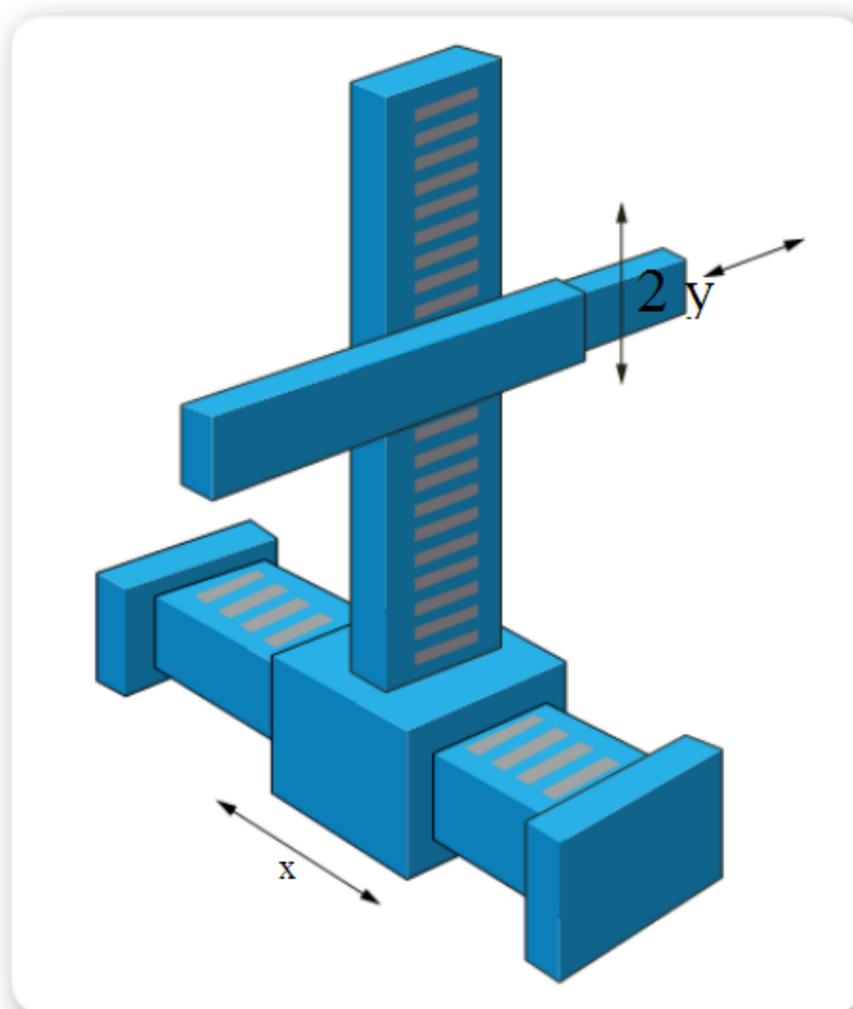


Figura 11. Robot cartesiano de tres ejes lineales de desplazamiento. Esta estructura puede alcanzar posiciones en el espacio.

Para la construcción de un sistema, existen módulos lineales **mono eje** que proveen movimientos en una dimensión. Hay tres modelos, en función del tipo de movimiento lineal que realizan:

- **Eje portal** : se mueve solo el carro. El cuerpo que actúa como guía no se desplaza; es el carro el que lo hace sobre esta guía, para cubrir el área de trabajo.
- **Eje de extensión**: se mueve solo el cuerpo del eje, el cual se introduce en el área de trabajo.



CONTRIBUCIÓN DE LA ROBÓTICA



La robótica colabora, en gran medida, con el incremento del empleo. Esto se logra automatizando los procesos, empleando máquinas más flexibles, reduciendo el costo de la maquinaria y produciendo una variedad de productos sin necesidad de realizar cambios importantes en la forma de fabricación. De esta manera, se provoca la creación de una gran cantidad de pymes y se contribuye a la descentralización de la industria.

- **Eje telescópico:** el carro y el cuerpo del eje se mueven fuera de la posición compacta de retracción y se introducen en el área de trabajo.

Los sistemas de dos y tres ejes pueden clasificarse, según el área de trabajo, en:

- **Posicionador lineal:** se encuentra encima o debajo del área de trabajo. Trabaja en **dirección X/Z** y está diseñado especialmente para el transporte dinámico de cargas en trayectos de recorrido corto en **dirección Z**.
- **Robot lineal:** está junto al área de trabajo, y fue diseñado para transportar cargas a gran velocidad en trayectos de recorrido corto.
- **Robot portal:** está situado encima del área de trabajo, para ahorrar espacio y transportar cargas en trayectos de recorrido largos.
- **Robot cilíndrico:** realiza dos movimientos lineales (de altura y de radio), combinados con un tercer movimiento rotacional en su base, por lo cual, presenta tres grados de libertad.
- **Robot esférico:** combina dos movimientos de rotación y uno de desplazamiento lineal que permiten al robot posicionarse en muchas direcciones y extender su garra o gripper.
- **Robot angular:** utiliza tres movimientos de rotación para posicionarse, presentando una articulación con movimiento rotacional y dos angulares. Estos robots se parecen al brazo humano, con cintura, hombro, codo y muñeca.
- **Robot SCARA:** es similar a la configuración cilíndrica, pero el radio y la rotación se obtienen por uno o dos tramos. Este brazo puede realizar movimientos horizontales de mayor alcance debido al empleo de dos articulaciones rotacionales.



ACTUADORES



Los elementos generadores de movimiento en las articulaciones pueden ser, según la energía que los impulsa, de tipo **neumático**, **hidráulico** o **eléctrico**. La impulsión neumática brinda gran velocidad de respuesta y un bajo costo. La hidráulica utiliza algún fluido de presión, como el aceite mineral. Por último, los motores eléctricos son los más utilizados en robótica por su sencillez, precisión y repetibilidad.

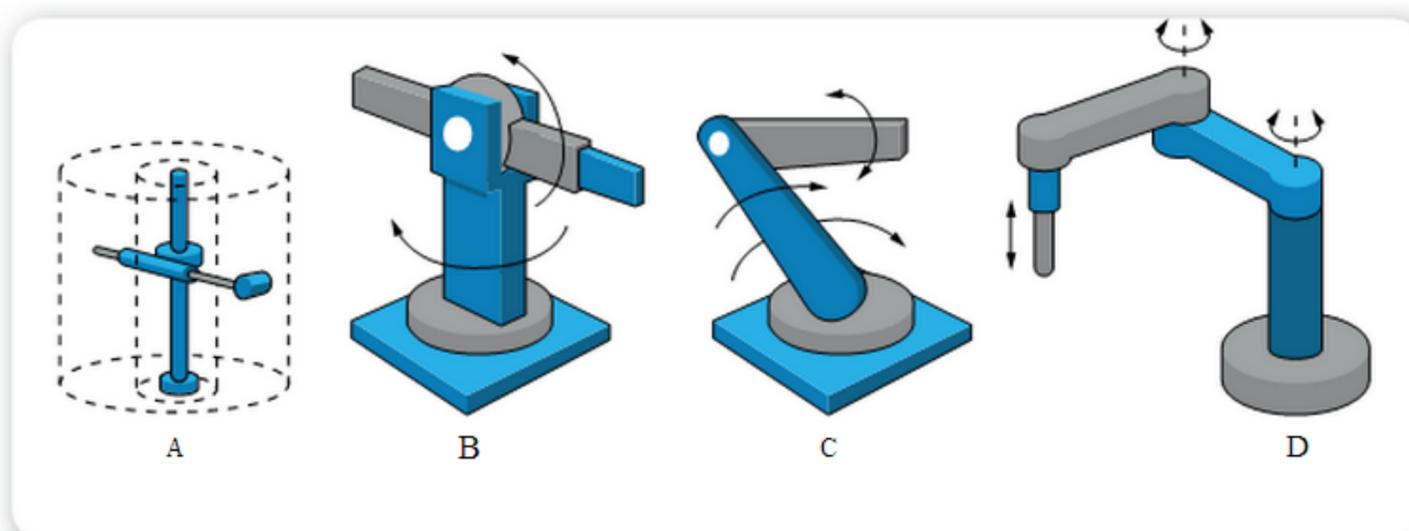


Figura 12. Aquí vemos las configuraciones más comunes y el tipo de movimiento de cada articulación: (A) Cartesiano, (B) Esférico, (C) Angular y (D) SCARA.



Fuentes de alimentación

La fuente de alimentación es un dispositivo que provee de energía a un robot. Existen fuentes que convierten la tensión alterna de la red en una o en varias tensiones continuas. Las hay de tipo lineal, conmutadas y especiales.

Otro tipo de fuente de energía es la batería, que entrega una corriente continua estable. Luego de usarla, tiene que ser recargada, según una tabla, en función de sus características.

La elección de la fuente de alimentación depende de la aplicación. Es decir, si el robot tiene que desplazarse de manera autónoma, se alimentará, seguramente, con baterías eléctricas recargables; en cambio, si no necesita desplazarse, o solo debe hacerlo mínimamente, será posible alimentarlo mediante una fuente de corriente alterna a través de un convertidor. Cuando utilizamos baterías para nuestro robot, es muy probable que necesitemos obtener varias tensiones de distinto valor. Para lograrlo, podemos emplear una fuente de alimentación conmutada, que permite aprovechar al máximo la energía de las baterías.

En general, se emplean alimentaciones separadas para la electrónica y para los servomotores, ya que estos últimos requieren más intensidad. Al momento de ser activados, estos dispositivos producen picos de corriente que podrían afectar gravemente al microcontrolador. Si empleáramos un regulador de tensión, se perdería mucha energía debido a la disipación de calor.

Tenemos que considerar que tanto las baterías como las fuentes de alimentación conectadas a CA deben suministrar la corriente necesaria para el funcionamiento de un robot y ser capaces de afrontar los picos de corriente que producen los actuadores al ser activados.

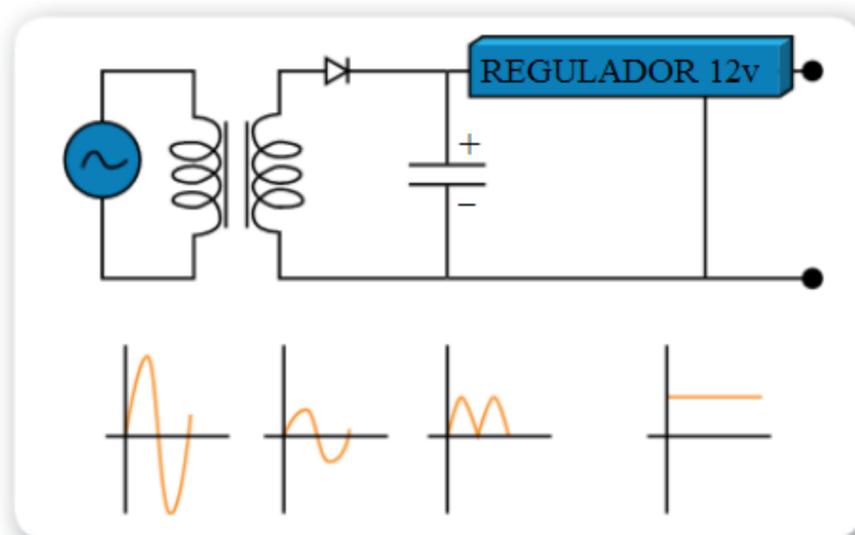


Figura 13. Fuente de poder lineal. Las etapas: reducción, rectificación, filtrado y regulación. Abajo, la onda del voltaje en cada una de las etapas.

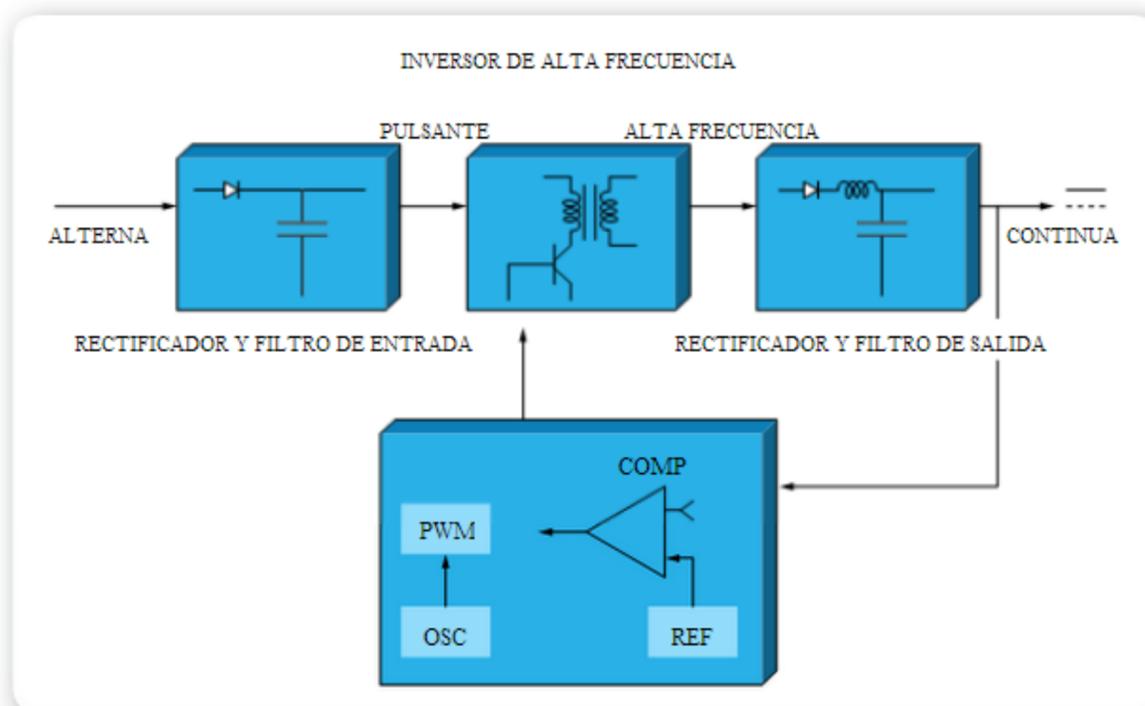


Figura 14. Esquema de una fuente conmutada estándar, donde se puede apreciar el lazo de realimentación conectado a la salida de tensión.

Microcontroladores aplicados a la robótica

Los **microcontroladores** son la herramienta fundamental en la elaboración de robots. Según el desempeño en la respuesta del

robot, debemos recurrir a modernos equipos de control, como los **DSP** (*Digital Signal Processor*).

La mayoría de las señales procedentes del mundo real son analógicas, como temperatura, velocidad del viento y sonido. En los procesos industriales, existen parámetros analógicos, como el espesor de una chapa, la distancia desde un robot al objeto, la fuerza, etcétera.

A partir de la aparición de los microprocesadores, se simplificó la construcción de circuitos integrados que contenían procesadores digitales para el tratamiento matemático de las señales analógicas. Podemos definir a los DSP como dispositivos especializados en el procesamiento de la información proveniente de una secuencia de muestras de una señal analógica.

Las señales que se desean procesar, pertenecientes a una serie de muestras, deben ser convertidas a formato digital por medio de los **convertidores A/D**. Por lo general, estas vienen de un sensor o transductor, por lo que la frecuencia de captura de las muestras deberá ser, como mínimo, el doble de la frecuencia máxima contenida en la señal analógica (teorema de Sanan-Nyquist).

Una vez obtenido el valor, según la finalidad que se desee, habrá que someter a las muestras a una o a varias operaciones matemáticas. La velocidad de procesamiento dependerá de las operaciones matemáticas que se le realicen a la muestra, antes de que la próxima medición ingrese para ser procesada.

Una aplicación práctica de los DSP –donde se requiere una gran capacidad de procesamiento de la información que llega de los sensores– son los robots que tienen control del equilibrio; en especial, si nos referimos al andar de los robots bípedos y a los de dos ruedas que funcionan como un péndulo invertido para mantener el equilibrio balanceándose sobre su eje vertical.



RESUMEN

A lo largo de este capítulo, detallamos los conceptos más relevantes de la robótica, analizamos las distintas morfologías, sistemas de locomoción y parámetros propios que definen a un robot. También pudimos conocer las distintas fuentes de energía que se utilizan en los circuitos electrónicos y, finalmente, estudiamos los métodos de control más difundidos y la aplicación de los DSP a la robótica.

Actividades

TEST DE AUTOEVALUACIÓN

- 1 ¿Cuáles son las características principales de un robot y cómo se puede definir a un robot industrial?
- 2 ¿Cómo se pueden clasificar los diferentes tipos de robots, según sus características y capacidades?
- 3 ¿Cuáles son los tipos de articulaciones que se utilizan en un manipulador?
- 4 ¿Cuál es el dispositivo encargado de manejar cada uno de los movimientos de un manipulador?
- 5 ¿Qué tipos de controladores existen?
- 6 ¿Cuáles son las características diferenciales que debemos tener en cuenta al momento de seleccionar un robot?
- 7 ¿Qué características tienen las cuatro configuraciones clásicas relacionadas con los modelos de coordenadas de que dispone un robot?
- 8 ¿Qué debemos tener en cuenta para elegir una fuente de alimentación para nuestro robot?
- 9 ¿Qué papel desempeñan los microcontroladores en la construcción de un robot?
- 10 ¿Qué función cumplen los conversores A/D?

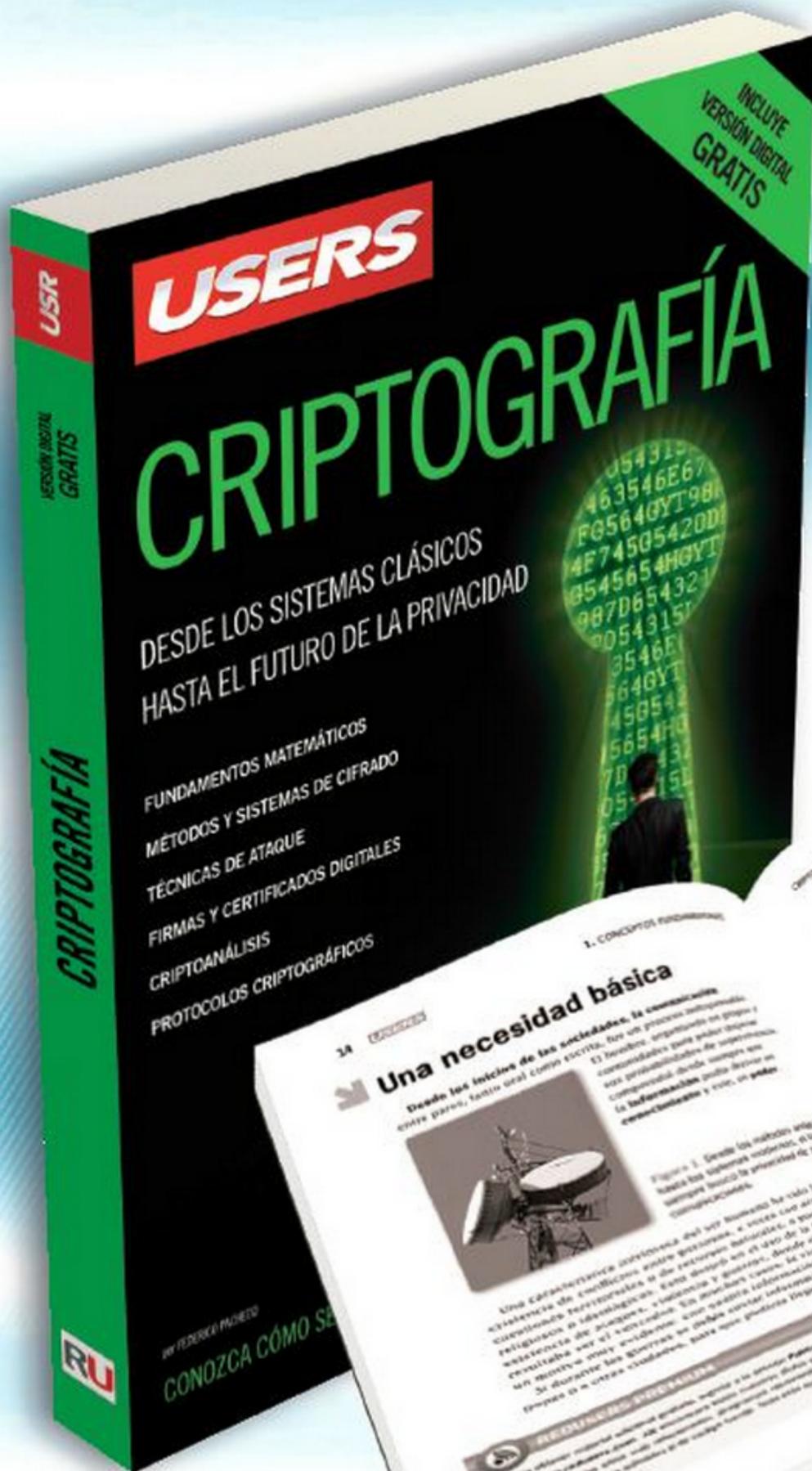


PROFESOR EN LÍNEA



Si tiene alguna consulta técnica relacionada con el contenido, puede contactarse con nuestros expertos: profesor@redusers.com

CONÉCTESE CON LOS MEJORES LIBROS DE COMPUTACIÓN



Una obra única que analiza la protección de datos y su evolución, desde la criptografía clásica a los algoritmos modernos.

- >> SEGURIDAD
- >> 208 PÁGINAS
- >> ISBN 978-987-1949-35-9



LLEGAMOS A TODO EL MUNDO VÍA  * Y  **

MÁS INFORMACIÓN / CONTÁCTENOS

 usershop.redusers.com  +54 (011) 4110-8700  usershop@redusers.com

* SÓLO VÁLIDO EN LA REPÚBLICA ARGENTINA // ** VÁLIDO EN TODO EL MUNDO EXCEPTO ARGENTINA



ELECTRÓNICA



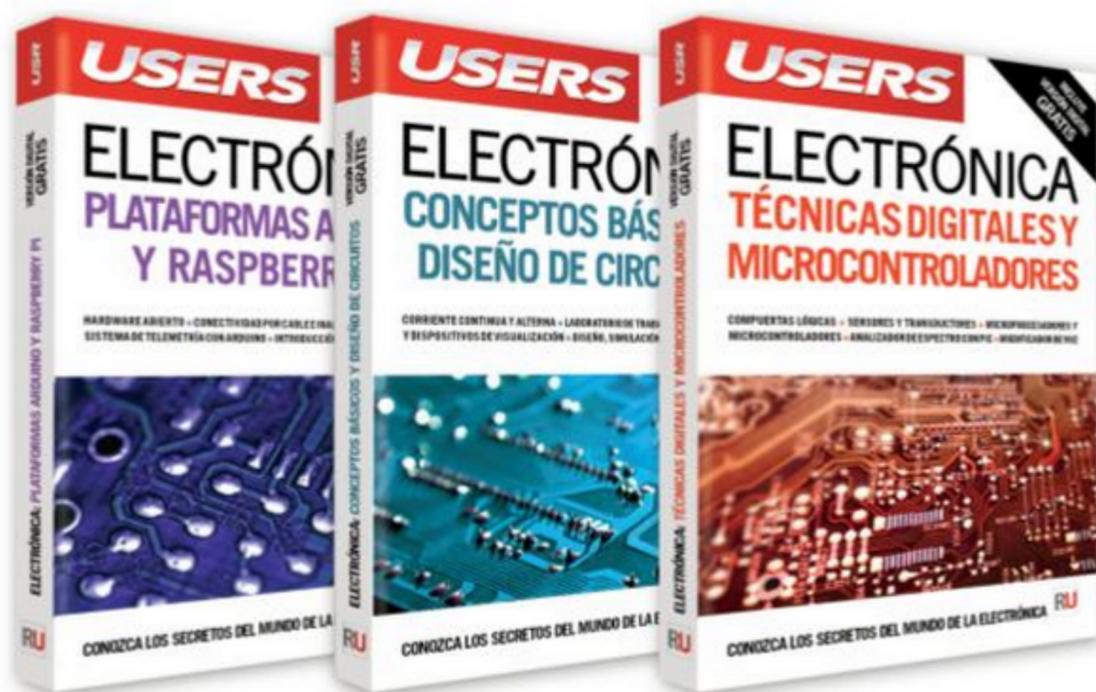
Plataformas Arduino y Raspberry Pi

Esta última entrega de la colección Electrónica propone un recorrido por la historia del movimiento del código abierto y del software libre, que servirá como introducción para conocer las plataformas abiertas más utilizadas en la actualidad: Arduino y Raspberry Pi.

Junto a sus entornos de desarrollo, estas dos herramientas son las que mejor se posicionaron en el mercado. Gracias a su facilidad de uso, su gran comunidad de usuarios, su flexibilidad y su eficiencia, permiten compartir el trabajo propio y realizar nuevas creaciones sobre la base de proyectos de otros técnicos.

* En este libro ENCONTRARÁ:

Plataformas abiertas: historia del software libre. Relación entre código fuente y circuitos. Plataformas de hardware libre más utilizadas. / **Plataforma Arduino:** historia y programación de Arduino. Aplicaciones en electrónica, automatización y telemetría. Entorno de desarrollo IDE. / **Sistema de telemetría con Arduino:** implementación del proyecto. Limitaciones y alternativas. / **Raspberry Pi:** características de la plataforma. Software necesario. Carga del sistema operativo. Overclocking. Integración con Arduino. / **Conectividad por cable:** protocolos TCP/IP. Modulación y estándares. USB. Software y utilidades de diagnóstico. / **Conectividad inalámbrica:** principios de funcionamiento. Comunicaciones a corta, media y larga distancia.



Colección electrónica

El contenido de esta colección fue publicado previamente en los fascículos del curso visual y práctico Técnico en electrónica.



REDUSERS.com

En nuestro sitio podrá encontrar noticias relacionadas y también participar de la comunidad de tecnología más importante de América Latina.

profesor en LÍNEA

Ante cualquier consulta técnica relacionada con el libro, puede contactarse con nuestros expertos: profesor@redusers.com.

ISBN 978-987-1949-56-4



9 789871 949564 >