

# B o l e t í n UNAMente Robótica



Año 2 N° 6 Publicación Bimestral Octubre de 2013

## El pasado de Arduino

Arduino es una tarjeta de desarrollo de software y hardware libre que ha tenido mucho éxito en el ámbito mundial y que contempla aplicaciones en diferentes plataformas para poder seguir vigente, inclusive Google ha lanzado un kit de desarrollo basado en Arduino para teléfonos inteligentes con Andriod.

La idea de la tarjeta surge de un grupo de profesores en Ivrea, Italia en 2005, quienes se conocen como el núcleo del equipo: David Cuartielles, Gianluca Martino, Tom Igoe, David Mellis, y Massimo Banzi, quienes necesitaban un micro-controlador en una tarjeta con características específicas, pero lo que en ese momento había en el mercado, no satisfacía sus necesidades y era caro para sus estudiantes, 100 dólares aproximadamente, por lo que pensaron en desarrollar una tarjeta propia más barata. En ese entonces los creadores de Processing (programa en el que está basado el lenguaje de programación de Arduino) Ben Fry y Casey Reas, conocieron a Massimo; las piezas se fueron armando. Por otro lado, lo que ofrecían los micro-controladores en esa época eran pocas entradas, por lo que decidieron que no iban a escatimar en ellas.

Al trabajar con estudiantes, 100 dólares se les hacía demasiado costoso para una tarjeta, por lo que decidieron fijarse como reto un precio máximo de 30 dólares, que era el equivalente de salir a cenar pizza en algún lugar. Para el nombre se inspiraron en un bar al que asistían y que tenía por nombre “Bar di Re Ardiuno”, rey que un milenio antes había reinado en el lugar. Inclusive decidieron cambiar el color, del tradicional verde de otras tarjetas a un azul que los distinguiera.

Por otro lado, tuvieron la idea de presentar un producto listo para desempaquetar, conectar y usar, que fuera accesible, desde el punto de vista operativo, para gente que no tuviera muchos conocimientos de electrónica y/o programación (Massimo Banzi era arquitecto), y vaya que lo lograron, al menos se puede conectar y usar sin mayores problemas; la verdad es que algunos ingenieros no lo ven con buenos ojos, pues piensan que hay otros micro-controladores con mayor potencial, y existen, pero son de más difícil asimilación por los usuarios sin conocimientos en electrónica, por lo que, si éstos necesitan un prototipo más elaborado, tendrían la posibilidad de explicarlo con Arduino y presentarlo al grupo de ingenieros quienes les desarrollarán un trabajo con mayores alcances.

En verdad la tarjeta Arduino tiene beneficios, pues además de emplearse en diferentes sistemas operativos, mediante la elaboración de sencillas prácticas, se pueden aplicar conceptos de diferentes asignaturas en el control de prototipos mecanizados, hacer más atractivo el aprendizaje de las estructuras de programación, y tener un acercamiento a la tecnología controlada por la electrónica, no solamente a las jóvenes generaciones, sino también a diferentes disciplinas, tanto afines a la ingeniería como la de minas y metalurgia o geomática, o aparentemente desligadas de ella, como la psicología o el diseño gráfico.

*Alfredo Arenas González*  
*Profesor de la Facultad de Ingeniería*

## **Robótica en la nube**

En los últimos años, el concepto de computación en la nube ha ganado popularidad. Dicho concepto se refiere a un nuevo modelo de aprovisionamiento de servicios, basado en Internet, que permite a los proveedores y a los consumidores gestionar eficientemente recursos informáticos tales como almacenamiento, procesamiento y memoria. *Google Maps*, *Google Docs*, *Dropbox*, *Facebook* son claros ejemplos de servicios informáticos localizados en la nube. Nosotros como consumidores o clientes de esos servicios sólo tenemos que acceder a un navegador Web y una conexión a Internet para poder utilizarlos. De esta manera, en cuestión de segundos o minutos podemos conocer la ubicación de un sitio, crear documentos y compartirlos, almacenar imágenes y vídeos en la red, compartir información o contenido con nuestros amigos, o hacer búsquedas sobre cualquier tipo de tema.

¿Qué tiene que ver el cómputo en la nube con la robótica? Imaginemos que le damos al robot la siguiente instrucción ¡Trae una cerveza! Una de las tareas que el robot necesitaría ejecutar correctamente sería identificar la figura de lo que es una cerveza. Una posible aproximación a la resolución del problema sería reunir gran cantidad de imágenes de lo que es y no es una cerveza y entrenar un reconocedor de cervezas con dichas imágenes. Ahora imaginemos que le decimos al robot ¡Trae una manzana! En este caso tendríamos que realizar el mismo proceso de entrenamiento de un reconocedor, pero ahora de manzanas. La Robótica en la nube permite a los robots utilizar servicios localizados en Internet para ejecutar sus tareas y compartir información y recursos. De esta manera, otra posible solución al problema antes planteado consiste en realizar una búsqueda automática en *Google Images* para recabar suficientes imágenes de cualquier objeto y entrenar un reconocedor de dicho objeto. Mejor aún, sería tener bases de datos públicas con los parámetros de los reconocedores y tener acceso a ellas a través de servicios Web. De esta manera, si la tarea es reconocer cervezas, el robot se comunicaría con el servicio Web apropiado, el servicio analizaría la imagen de entrada y respondería si la imagen de entrada

es o no una cerveza. Se puede notar que la solución que ofrece el cómputo en la nube permite: 1) delegar la pesada función de reconocimiento de objetos a otra computadora o red de computadoras, que posiblemente cuenten con mejores características de memoria y procesamiento, comparadas con las de nuestro robot; y, 2) hacer pública la funcionalidad del reconocedor para que pueda ser utilizada por otros robots.

Por lo tanto, la Robótica en la nube tiene consecuencias positivas entre las que destacan incrementar el procesamiento, economizar recursos y compartir el conocimiento. En primer lugar, consumir servicios alojados en Internet implica utilizar la memoria y el procesamiento de uno o más servidores en Internet para realizar operaciones con gran cantidad de datos; en segundo lugar, implica la reducción del costo, consumo de energía y peso del robot; y finalmente, implica compartir el conocimiento o funcionalidades con otros robots.



*Figura 1 RoboEarth permite compartir y reutilizar el conocimiento entre diferentes tipos de robots. Es posible consultar su página web en <http://www.robearth.org>.*

Navegación, reconocimiento de objetos, y reconocimiento de voz y texto son algunos de los servicios que actualmente podrían ser utilizados por nuestros robots. De manera que sería posible crear un robot, que indique en qué lugar se encuentre, cuáles son los objetos que están a su alrededor y que responda según comandos de voz; todo ello, utilizando servicios en la nube. Sin embargo, es necesario recalcar que sin conexión a Internet, ninguno de estos servicios estaría disponible. Además, la comunicación entre el robot y los servicios está limitada por la conexión misma.

Actualmente, diversos grupos de investigadores se encuentran trabajando en la creación de cerebros para robots en la nube. Uno de los proyectos más atractivo y ambicioso es el llamado RoboEarth, que según su página Web, se define como la World Wide Web para robots. Se trata de una plataforma robótica en la nube que permite a los robots compartir y reutilizar el conocimiento. Por otra parte, Google, empresa proveedora de diversos servicios en la nube, también trabaja en este campo.

En conclusión, la robótica en la nube permitirá crear robots con cerebros que se alimentarán de la generación de conocimiento por parte de grupos de investigación en robótica de todo el mundo; robots que serán capaces de responder eficientemente a las demandas de escalabilidad y disponibilidad de sus tareas.

*Luis Gilberto Mateos Ortiz*  
*Egresado de la carrera de Ingeniería Mecatrónica*

## **El *Blinki Wonder*: un led lleno de encanto y simpatía didáctica**

El **IDE** de **Arduino** y la importancia del *Blinki Wonder*

Trabajar de manera creativa y entusiasta con la tarjeta **Arduino** es combinar un circuito, lo más sencillo posible, con una idea dispuesta a materializarse en un programa; este programa deberá editarse y grabarse en el microcontrolador. Este proceso no es, ni más ni menos, que la formulación del **IDE** o entorno de desarrollo integrado de **Arduino**.

Todo chamaco (o ñoño) que se inicia con esta tarjeta, tiene la obligación moral de perder su inocencia electrónica con el montaje del tierno y fresísimo led parpadeante<sup>2</sup> al que en el Taller de Robotica del Anexo se le ha bautizado con el simpático mote de *Blinki Wonder*.

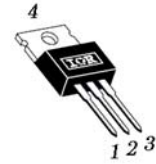
En este artículo se pretende persuadir a que toda persona que haya realizado y comprendido esta graciosa y elemental idea electrónica, que está lo suficientemente capacitada para mover motores, encender lámparas inteligentes de alta potencia, controlar semáforos a partir de relés entre otras ocurrencias más ingenieriles. Lo único que necesita conocer de manera muy rudimentaria, es la idea de cómo funciona un transistor de efecto de campo tipo *MOSFET* en su esquema de operación como conmutador.

**Arduino** padece de disfunción y no brinda potencia, pero el *MOSFET* le remedia esta incompetencia. Un microcontrolador es un invento genial casi perfecto, y por lo tanto debe tener al menos un defecto: no puede manejar potencia para poder mover motores, prender semáforos, o bien, cualquier carga que demande altas intensidades de corriente, por lo que se debe emplear una fuente externa y un circuito de potencia para lograr este propósito. Es aquí donde interviene el empleo de un *MOSFET* como conmutador encendido-apagado como una posible solución a este problema.

Funcionamiento del *MOSFET* como conmutador para entregar potencia

En la Figura 1 se muestra el *patigrama* del *MOSFET* Buz11 y algunas propiedades eléctricas; en la Figura 2 la manera en cómo se conecta este dispositivo con una fuente de voltaje y dos resistencias para que funcione como conmutador. Al considerar la señal lógica alta  $V_{in} = 5\text{ V}$ , se abre la compuerta (*Gate*) y permite el paso de una corriente  $I_G$  muy pequeña, pero que permite que el *MOSFET* tenga una señal de salida  $I_D$  que va a depender de los valores  $V_{CC}$  y  $R_L$ . En contrapartida una señal de nivel lógico bajo, cierra la compuerta e impide el paso de corriente  $I_G$ , luego  $I_D = 0\text{ A}$  y  $V_0 = 0\text{ V}$ .

Esta corriente  $I_G$ , que bien puede operar como una señal de control a la entrada del *MOSFET*, la puede generar la tarjeta **Arduino**, con la gran ventaja de proporcionar una intensidad de corriente del orden de microamperios, cantidad de corriente suficientemente pequeña que puede ser suministrada por este microcontrolador y operar satisfactoriamente con muy baja potencia.



- 1 *Gate* (G)
- 2 *Drain* (D)
- 3 *Source* (S)
- 4 *Drain* (D)

$V_{DSS} = 150\text{ V}$   
 $R_{DS(on)} = 0.042\Omega$   
 $I_D = 43\text{ A}$

Figura 1 *Patigrama* y características.

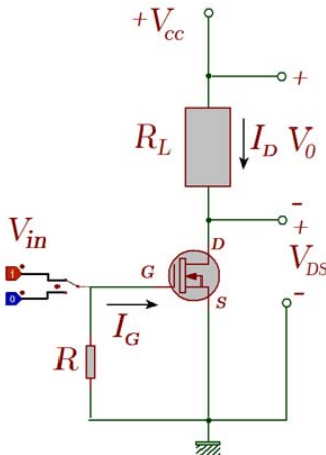


Figura 2 *MOSFET* como conmutador.

Propiedades *MOSFET* de conmutación

Si  $V_{in} = 5\text{ V}$ , se abre la compuerta

$I_G$  del orden de  $\mu\text{A}$  y  $P_{in} \ll 10\text{ mW}$

$V_0 \approx V_{CC}$   $V_{DS} \approx 0\text{ V}$  e  $I_D \approx V_{CC}/R_L$

$P_0 \approx V_{CC} I_D = R_L I_D^2 = V_{CC}^2/R_L$

Si  $V_{in} = 0\text{ V}$

$I_G = 0\text{ A}$  y  $P_{in} = 0\text{ W}$

$I_D = 0\text{ A}$  y  $V_0 = 0\text{ V}$ , luego  $P_0 = 0\text{ W}$

( $V_{DS} = V_{CC}$ ! ¿porqué?)

También  $P_0 \neq P_{in}$  ¿porqué?

<sup>1</sup>Por las siglas en inglés de Integrated Development Environment.

<sup>2</sup>El ejemplo de inicio llamado Blink. Ver en el IDE de **Arduino** en Archivo/Ejemplos/01 Basics/Blink.

Dado que  $R_L$  representa la resistencia de la carga final que debe controlarse y cuyo consumo de potencia lo debe suministrar la fuente de voltaje  $V_{CC}$ , esta resistencia bien puede representar un motor, una lámpara o cualquier dispositivo que demande una potencia considerable. Suponga que la carga aludida corresponde a un motor; si el pin 13 de la tarjeta **Arduino** se conecta a la entrada  $V_{in}$  del *MOSFET* de la Figura 2, tal como se muestra en el montaje de la Figura 3, y a continuación se escribe en el IDE de **Arduino** de la computadora el código correspondiente al Blink, que se muestra en el listado de la Figura 4 después de guardar, compilar, correr el programa y observar lo que pasa en la tarjeta **Arduino** y demás chunches que lo acompañan; se dará cuenta que . . . el motor gira, cada vez que el led parpadea. ¿Cómo la ves Andrés?

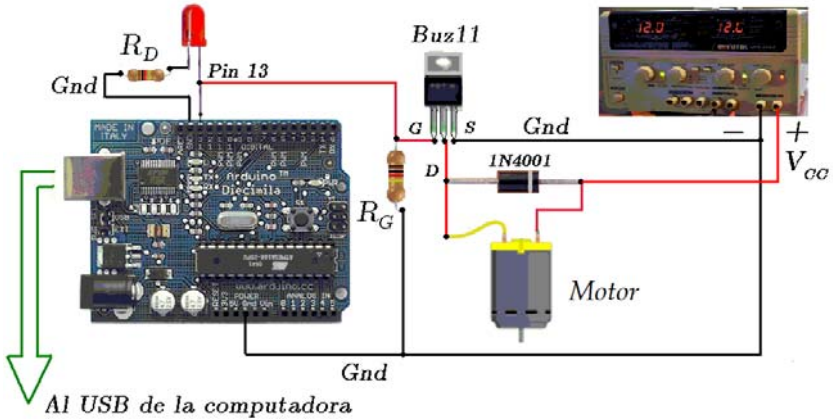
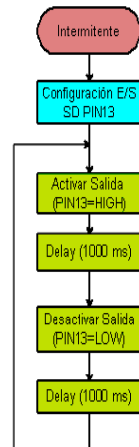
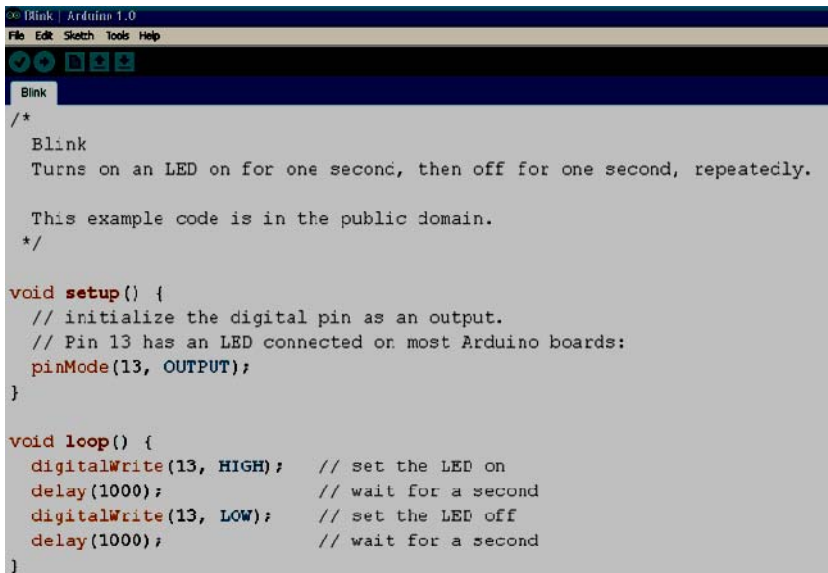


Figura 3 Dos cargas controladas por el **Arduino**, con la mismísima idea pero con diferente propósito.

En el siguiente número se presentará la continuación de este entorno de desarrollo consistente en cómo pasar de un motor parpadeante al control de su velocidad angular.



The image shows a screenshot of the Arduino IDE interface. At the top, the title bar reads "Blink | Arduino 1.0". Below it is a menu bar with "File", "Edit", "Sketch", "Tools", and "Help". A toolbar with icons for saving, running, and other functions is visible. The main window displays the code for the "Blink" program. The code includes a multi-line comment describing the program's function: "Blink Turns on an LED on for one second, then off for one second, repeatedly. This example code is in the public domain." The code defines a setup function to initialize pin 13 as an output and a loop function that toggles the LED on and off with 1000ms delays.

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

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}
```

Figura 4 Diagrama de flujo y código del programa Blink.

#### Observaciones

Los valores de las resistencias empleadas son  $R_D = 220 \Omega$ ,  $R_D = 22 \text{ k}\Omega$ , el valor de la fuente  $V_{CC} = 12 \text{ V}$ , el motor del mismo voltaje. Importante: el diodo 1N4001 colocado entre la fuente (*Source*) y el drenaje (*Drain*) del MOSFET, sirve para proteger a este dispositivo debido a posibles voltajes de la fuerza contra-electromotriz ocasionada por el motor (¿what? . . . pregúntaselo a tu *profe* de Electromagnetismo). El MOSFET Buz11 puede sustituirse por otro, como puede ser el IRFZ34N.

*Hugo Germán Serrano Miranda*  
*Profesor de carrera de la Facultad de Ingeniería*

#### Seminario de Robótica aplicada al aprendizaje de la ciencia y la tecnología

Con un nutrido grupo de académicos y alumnos interesados, se realizó este evento académico en el Auditorio Raúl J. Marsal el pasado 16 de mayo del año en curso. Fue una actividad que integró, tanto en su organización como en su participación y realización, a docentes del Colegio de Ciencias y Humanidades, CCH, la Escuela Nacional Preparatoria, ENP, y la Facultad de Ingeniería, FI. A lo largo del día se presentaron diecisiete ponencias, seis del CCH, tres de la ENP y ocho de la FI; además, se impartió una conferencia magistral a cargo de

la maestra Josefina Bárcenas López y el maestro José Antonio Domínguez Hernández del CCADET sobre el tema central de este Seminario.

Las conclusiones más importantes que se obtuvieron fueron las siguientes:

- 1 La robótica provee situaciones didácticas que pueden lograr en los alumnos su desarrollo cognitivo y el aprendizaje de aspectos científicos y tecnológicos, dado que generan conexiones significativas entre sus estructuras mentales y sus conocimientos previos con las experiencias vividas durante la resolución de los problemas inherentes a dicha área de conocimiento.
- 2 Es fundamental la divulgación de las experiencias entre los profesores, con el fin de perfeccionar la aplicación de las estrategias de aprendizaje que promuevan el aprendizaje de los alumnos en las ciencias básicas por medio de la robótica, así como el desarrollo de distintas habilidades y actitudes deseables en todo profesionista.
- 3 Una necesidad actual cada vez mayor es la automatización de procesos y servicios, por consiguiente los profesionales tendrán que interactuar con este tipo de sistemas, ya sea en su desarrollo o mantenimiento. Para afrontar este reto, es necesario que los alumnos tengan buenos antecedentes en ciencias básicas, sobre todo en álgebra lineal, cálculo vectorial, ecuaciones diferenciales, cinemática y dinámica así como electricidad y magnetismo.
- 4 La investigación básica es una excelente estrategia de aprendizaje de la inteligencia artificial y la robótica; a través de los años, se ha transitado por diferentes paradigmas de aprendizaje, desde el pasivo (tradicional) pasando por el activo (constructivista) y el modelo piagetiano (epistemología genética), hasta llegar al modelo actual, todavía en desarrollo correspondiente al que se le puede denominar Robótica y Educación.
- 5 Tanto en el CCH como en la ENP se ha buscado trabajar con jóvenes a través de los Clubes de Robótica e Informática y se ha fomentado la participación en concursos y eventos de robótica, nacionales e internacionales como la RoboCup, lo que ha permitido desarrollar en los alumnos habilidades manuales y de diseño así como la integración de conocimientos de materias de matemáticas y física.

En este encuentro académico fue posible favorecer el intercambio de experiencias sobre la implementación de la robótica en el proceso enseñanza aprendizaje de la ciencia y la tecnología, de docentes del CCH, la ENP y la FI, por tanto los objetivos del Seminario se cumplieron a cabalidad.

*Yukihiko Minami Koyama*

*Profesor de carrera de la Facultad de Ingeniería*

**Responsable: Alfredo Arenas G. [unamente.robotica@gmail.com](mailto:unamente.robotica@gmail.com)  
<http://dcb.fi-c.unam.mx/Publicaciones/UNAMenteRobotica>**