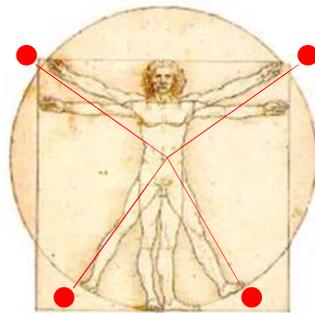


# TECNOLOGÍ@ y *DESARROLLO*

*Revista de Ciencia, Tecnología y Medio Ambiente*

VOLUMEN XIII. AÑO 2015

SEPARATA



**UNA MIRADA AL MUNDO ARDUINO**

**José Carlos Herrero Herranz, Jesús Sánchez Allende**



UNIVERSIDAD ALFONSO X EL SABIO  
Escuela Politécnica Superior  
Villanueva de la Cañada (Madrid)

© Del texto: José Carlos Herrero Herranz , Jesús Sánchez Allende  
Mayo, 2015.

<http://www.uax.es/publicacion/una-mirada-al-mundo-arduino.pdf>

© De la edición: *Revista Tecnología y desarrollo*

Escuela Politécnica Superior.

Universidad Alfonso X el Sabio.

28691, Villanueva de la Cañada (Madrid).

ISSN: 1696-8085

Editor: Javier Morales Pérez – [tecnologia@uax.es](mailto:tecnologia@uax.es)

No está permitida la reproducción total o parcial de este artículo, ni su almacenamiento o transmisión ya sea electrónico, químico, mecánico, por fotocopia u otros métodos, sin permiso previo por escrito de la revista.

## UNA MIRADA AL MUNDO ARDUINO

**José Carlos Herrero Herranz <sup>a)</sup>, Jesús Sánchez Allende <sup>b)</sup>**

<sup>a)</sup> Máster en Ingeniería Ambiental. Ingeniero Eléctrico. Profesor Sistemas Electrónicos, IES Palomares Vallecas [ctdoc2012@gmail.com](mailto:ctdoc2012@gmail.com)

<sup>b)</sup> Dr en Ingeniería de Telecomunicación. Ingenierías TIC. Escuela Politécnica Superior Universidad Alfonso X el Sabio [jallende@uax.es](mailto:jallende@uax.es)  
Avda de la Universidad nº 1, Villanueva de la Cañada 28691 Madrid,, España

RESUMEN: Este primer artículo sobre Arduino es eminentemente descriptivo y ofrece una visión global de los aspectos más significativos del mundo Arduino, el hardware, los distintos modelos de placas oficiales, deteniéndose con un mayor detalle en el modelo UNO R3, las placas compatibles, el entorno y lenguaje de programación de Arduino, así como otros entornos gráficos de programación, las *shields* (escudos) que conectan las plataformas con el mundo exterior y permiten que se puedan comunicar, conectarles sensores o controlar actuadores.

PALABRAS CLAVE: Arduino, hardware libre, microcontrolador, robótica, electrónica.

ABSTRACT: This first article about Arduino is eminently descriptive and provides an overview of the most significant aspects of the Arduino world, hardware, official models of board, stopping in greater detail in the model UNO R3, compatible boards, IDE and Arduino programming language and other graphical programming environments, the shields that connect the platforms with the outside world and allow them to communicate, connect them to sensors and driver actuators.

KEY-WORDS: Arduino, open hardware, microcontroller, robotics, electronic.

SUMARIO: 1. Arduino 2. Arduino UNO 3. Entorno de desarrollo integrado para Arduino 4. Familia Arduino 5. Shields 6. Otros entornos de programación de Arduino 6.1. Ardublock 6.2. Miniblock 6.3. Modkit 6.4 Scratch 6.5 Physical Etoys 6.6. Ardulab 7. Clones y compatibles Arduino 8. Arduino en el espacio, ArduSat 9. Fritzing 10. Conclusiones 11. Webgrafía. Introducción 2. Texto dividido en apartados 3. Conclusiones 4. Bibliografía

## 1. Arduino

Es un sistema microcontrolador monoplaca, de hardware libre, de fácil uso y bajo coste, desarrollado inicialmente para facilitar el uso de electrónica en diseños artísticos e interactivos y la aplicación de esta por personas no expertas.<sup>1</sup>

Actualmente hay más de 20 modelos de plataformas Arduino con diferencias en cuanto a características y posibilidades, número de entradas/salidas, microcontrolador, etc, pero compatibles entre sí, manteniendo una compatibilidad de abajo arriba, esto es, una aplicación que funciona en una plataforma, funcionará en otra más compleja, (superior), y que por tanto incluya las características de la primera.

Los microcontroladores más habituales en la plataforma son los de la familia AVR de ATMEL, aunque algunas plataformas utilizan otros microcontroladores, ejemplo Cortex M3 de ARM, de 32 bits.

Para facilitar su uso y programación se desarrolló simultáneamente y conjuntamente con la plataforma Arduino un IDE (entorno de desarrollo integrado), en el que se usa un lenguaje de programación parecido a C++, basado en el lenguaje Wiring<sup>2</sup>, el entorno de desarrollo está basado en Processing<sup>3</sup>. El IDE permite editar, compilar y enviar el programa a la plataforma Arduino que se está utilizando, así como comunicarse vía serie y mostrar los datos en una ventana terminal. La plataforma Arduino se comunica con el IDE mediante un programa cargador (bootloader), precargado en el microcontrolador de la plataforma Arduino. EL IDE es software libre y se puede descargar gratuitamente desde el sitio web oficial de Arduino<sup>4</sup>.

Arduino se puede utilizar para desarrollar objetos interactivos, que pueden funcionar de forma autónoma, sin necesidad de estar conectados a un ordenador, o puede conectarse con otro software que se esté ejecutando en un ordenador, como por ejemplo Flash, Processing, Max/MSP, LabView, MatLab, entre otros<sup>5</sup>.

Con Arduino se puede tomar información del entorno a través de sensores conectados a sus entradas analógicas y digitales, puede controlar luces, motores y otros actuadores directamente o partir de las señales de control generadas en sus salidas. Hay modelos de Arduino específicos desarrollados para facilitar llevar tecnología puesta (weareables), o

<sup>1</sup> <http://arduino.cc/>

<sup>2</sup> <http://wiring.org.co/>

<sup>3</sup> <http://www.processing.org/>

<sup>4</sup> <http://arduino.cc/en/Main/Software>

<sup>5</sup> <http://arduino.cc/en/Guide/Introduction>

en la ropa, e-textiles <sup>6</sup>. Puede comunicarse con otras placas Arduino o con otros sistemas, mediante Wifi, Ethernet, Bluetooth, etc., esto permite también la interacción a distancia y el Internet de las cosas (*IoT*).

Arduino encuentra un uso amplio en la enseñanza: en materias relacionadas con la robótica, el control, la adquisición de datos, los diseños interactivos, etc., la sencillez de uso de esta plataforma permite a personas no expertas en electrónica, utilizar en sus creaciones dispositivos electrónicos y controlarlos, de una manera sencilla o asequible para ellos.

Las posibilidades de realizar desarrollos basados en Arduino tienen como límite la imaginación.

Arduino nació con el objeto de proporcionar una plataforma económica y fácil uso, para enseñar electrónica a los alumnos del Instituto de Diseño Interactivo de Ivrea, (Italia). Sus creadores y socios fundadores son los italianos Massimo Banzi y Gianluca Martino, los estadounidenses Tom Igoe y, David Mellis y el español David Cuartielles, quienes decidieron que su diseño fuera abierto licenciándolo en Creative Commons <sup>7</sup>, protegiendo la marca, pero el hardware, el software y la documentación son libres, se pueden descargar los archivos de diseño de un circuito electrónico, los fuentes del software y se pueden modificar, compartir etc. <sup>8</sup> En la actualidad hay más de 1.200.000 placas Arduino oficiales distribuidas por todo el mundo, y entre 3 y 4 millones de usuarios visitan de forma regular el sitio web cada trimestre <sup>9</sup>

## 2. Arduino UNO

De todas las plataformas Arduino, la más representativa y extendida entre los usuarios, es Arduino UNO R3 <sup>10</sup>, que es la versión actual de las primeras versiones de Arduino, suele tomarse como referente y es con la que se suele identificar al sistema Arduino al referirse a él sin más. La sencillez de uso, versatilidad y bajo coste hacen de Arduino UNO R3, una plataforma muy adecuada tanto como sistema final o como sistema de desarrollo para de proyectos a implementar con otras plataformas Arduino. Por este motivo es habitual tomar la plataforma Arduino UNO como punto de partida para adentrarse en el mundo Arduino y conocer sus características, elementos, etc.

---

<sup>6</sup> <http://www.quees.info/que-es-wearable.html>

<sup>7</sup> <http://es.creativecommons.org/blog/>

<sup>8</sup> <http://vimeo.com/18390711>

<sup>9</sup> <http://readwrite.com/2014/05/12/arduino-massimo-banzy-diy-electronics-hardware-hacking-builders>

<sup>10</sup> <http://arduino.cc/en/Main/ArduinoBoardUno>

Arduino UNO es un sistema basado en el microcontrolador de 8 bits ATmega328, de Atmel ©, un chip sencillo y de bajo coste. Arduino UNO, tiene 14 pines (0 a 13), cada uno de los cuales se puede configurar como entrada o salida digital. Los pines configurados como salida pueden proporcionar o absorber una corriente de hasta 40 mA, suficiente para excitar multitud de circuitos, sensores, etc. aunque insuficiente para otros, tales como algunos relés, solenoides, motores, etc., para los que es preciso utilizar algún circuito excitador. Seis de estos pines se pueden configurar también como salidas PWM, lo que permite variando el ciclo de trabajo de la señal cuadrada generada en el pin, obtener una tensión cuyo valor medio puede variar entre 0 y 5 V y así simular una salida analógica sobre una salida digital.

La placa incluye un LED conectado al pin 13, lo que permite entre otras cosas utilizarlo como dispositivo de salida en la verificación y depuración de programas. Los pines 0 y 1 de E/S digital se pueden configurar para utilizarse como un puerto serie.

El microcontrolador Atmega328 que usa Arduino UNO, incluye un convertor analógico-digital (A/D) de 6 canales, con una resolución de 10 bits, retornando un valor entero entre 0 y 1023. El uso principal de estos pines es para la lectura de sensores analógicos. Estos pines tienen también toda la funcionalidad de los pines de entrada-salida digitales, como los pines 0 - 13. Consecuentemente, si para alguna aplicación se precisan más pines de entrada-salida digital, y no se está usando ningún pin analógico, pueden usarse estos pines para ello.

La placa contiene todo lo necesario para utilizar el microcontrolador, cristal del oscilador de 16 MHz, regulador de tensión lineal, etc., es suficiente con conectarlo al ordenador a través del cable USB o alimentarlo desde una fuente de alimentación o una batería externa, con un voltaje recomendado de entre 7 a 9 V, para empezar a trabajar con él.

Tiene 32 KB de memoria Flash, 2 KB de memoria SRAM y 1KB de memoria EEPROM

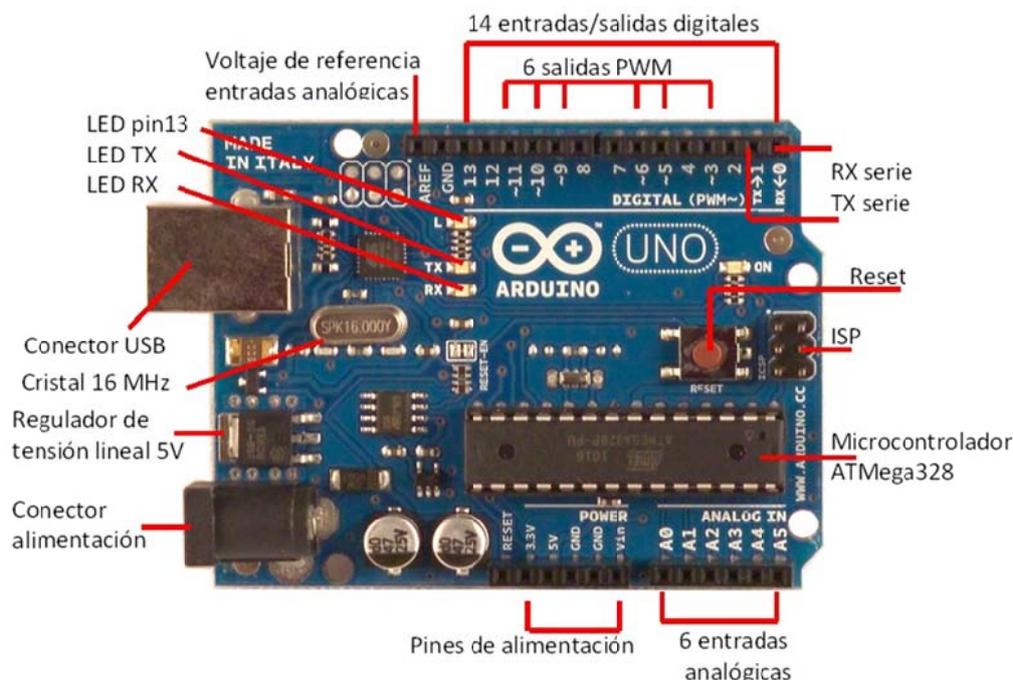


Fig. 2.1 Arduino Uno

### 3. Entorno de desarrollo integrado para Arduino

El entorno de desarrollo integrado, comúnmente IDE (*integrated development environment*) para Arduino, es una herramienta desarrollada en Java, por lo que puede utilizarse con distintos sistemas operativos, Windows, Mac OS X y Linux, es software abierto que se puede descargar desde la página principal de Arduino<sup>11</sup>. El IDE permite la edición y compilación de programas en el lenguaje de Arduino, y cargarlos en la plataforma para que se ejecuten. El entorno incluye multitud de ficheros de programas de ejemplo de utilización.

El IDE dispone también de una ventana tipo Terminal, para la comunicación por el puerto serie de las placas Arduino.<sup>12</sup>

<sup>11</sup> <http://arduino.cc/en/Main/Software>

<sup>12</sup> <http://arduino.cc/en/Guide/Environment>

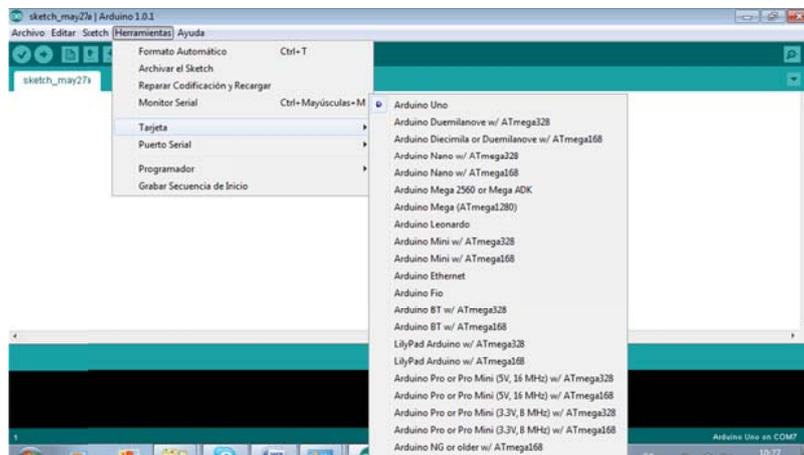


Fig. 3.1 Entorno de programación de Arduino.

### 3.1 Lenguaje de programación del IDE de Arduino

El lenguaje que utiliza este entorno para escribir los programas es parecido al C++. Todos los programas escritos para Arduino en el entorno IDE (sketches) contienen las funciones *setup()* y *loop()*. La función *setup()* se sitúa en primer lugar y es también la primera función en ejecutarse, habitualmente una sola vez. Esta función contiene la declaración de las variables, inicializaciones, los parámetros de configuración etc.

La función *loop()* contiene la parte del programa que se ejecutará cíclicamente, esto es, el bucle infinito. Es típico de esta función, la lectura de entradas, la activación de salidas, comprobaciones, etc.

Además de las funciones *setup()* y *loop()* citadas, un programa para Arduino puede incluir otras funciones creadas por el usuario, en el sitio web oficial de Arduino<sup>13</sup>, se proporciona información suficiente para crearlas.

En el mismo portal también hay disponible una serie de bibliotecas estándar para Arduino, para manejar hardware o datos<sup>14</sup>, para manejo de EEPROM, Ethernet, visualizadores LCD, tarjeta SD, servos, Wifi, entre otras y también es posible realizar otras bibliotecas que necesite el usuario, para lo que puede seguirse las recomendaciones dadas en el portal<sup>15</sup>.

<sup>13</sup> <http://playground.arduino.cc/Code/Function>

<sup>14</sup> <http://arduino.cc/en/Reference/Libraries>

<sup>15</sup> <http://arduino.cc/en/Hacking/LibraryTutorial>



Fig 3.2 Edición de un programa en el entorno de programación

A modo de resumen y para tener una visión global, en el cuadro siguiente se resumen las instrucciones, variables etc del lenguaje de programación Arduino, para una referencia completa de este se puede encontrar en [1], [2].

<p><b><u>Estructura básica de un programa</u></b></p> <pre>void setup() void loop()</pre> <p><b><u>Estructuras de control</u></b></p> <pre>if() if()...else for() switch()...case while() do()... while break continue return goto</pre> <p><b><u>Sintaxis</u></b></p> <pre>;(punto y coma) {} (llaves) // (comentario línea)</pre>	<p><b><u>Operadores compuestos</u></b></p> <pre>++ (incremento) -- (decremento) += (suma compuesta) -= (resta compuesta) *= (multiplicación compuesta) /= (división compuesta) &amp;= (AND bit a bit compuesto)  = (OR bit a bit compuesto)</pre> <p><b><u>Constantes</u></b></p> <pre>HIGH   LOW INPUT   OUTPUT true   false</pre> <p><b><u>Tipo de datos</u></b></p> <pre>void boolean char</pre>	<p><b><u>Funciones E/S digitales</u></b></p> <pre>pinMode(pin, [INPUT, OUTPUT]) digitalWrite(pin, valor) int digitalRead(pin)</pre> <p><b><u>Funciones E/S analógicas</u></b></p> <pre>int analogRead(pin) analogWrite(pin, value) analogReference([DEFAULT, INTERNAL, EXTERNAL])</pre> <p><b><u>Funciones E/S avanzadas</u></b></p> <pre>tone(pin, freqhz) tone(pin, freqhz, duración_ms) noTone(pin) shiftOut(pinDatos, pinRelej, [MSBFIRST,LSBFIRST], valor) shiftIn() unsigned long pulseIn(pin, [HIGH,LOW])</pre> <p><b><u>Funciones de tiempo</u></b></p>
---	---	---

<p>única)</p> <p>/* */ (comentario multilínea)</p> <p>#define</p> <p>#include</p> <p><b><u>Operadores aritméticos</u></b></p> <p>= (asignación)</p> <p>+ (suma)</p> <p>- (resta)</p> <p>* (multiplicación)</p> <p>/ (división)</p> <p>% (módulo)</p> <p><b><u>Operadores de comparación</u></b></p> <p>== (igual que)</p> <p>!= (no igual que)</p> <p>&lt; (menor que)</p> <p>&gt; (mayor que)</p> <p>&lt;= (menor o igual que)</p> <p>&gt;= (mayor o igual que)</p> <p><b><u>Operadores booleanos</u></b></p> <p>&amp;&amp; (AND)</p> <p>   (OR)</p> <p>! (NOT)</p> <p><b><u>Punteros</u></b></p> <p>* (valor: seguir puntero)</p> <p>&amp; (referencia: obtener puntero)</p> <p><b><u>Operadores a nivel de bit</u></b></p> <p>&amp; (AND bit a bit)   (OR bit a bit)</p> <p>^ (XOR bit a bit)</p> <p>~ (NOT bit a bit)</p> <p>&lt;&lt; (desplazamiento a la izquierda)</p> <p>&gt;&gt; (desplazamiento a la derecha)</p>	<p>unsigned char</p> <p>byte</p> <p>int</p> <p>unsigned int</p> <p>word</p> <p>long</p> <p>unsigned long</p> <p>short</p> <p>float</p> <p>double</p> <p>sizeof</p> <p>string</p> <p>array</p> <p><b><u>Conversiones</u></b></p> <p>char() byte() int()</p> <p>word() long() float()</p> <p><b><u>Calificadores</u></b></p> <p>static //persiste entre llamadas</p> <p>volatile //usa la RAM</p> <p>const //solo lectura</p> <p>PROGMEM //usar la flash</p> <p><b><u>Bits y Bytes</u></b></p> <p>lowByte(x)</p> <p>highByte(x)</p> <p>bitRead(x, bitn)</p> <p>bitWrite(x, bitn, bit)</p> <p>bitSet(x, bitn)</p> <p>bitClear(x, bitn)</p> <p>bit(bitn)</p>	<p>unsigned long millis()</p> <p>delay(miliseundos)</p> <p>delayMicroseconds(microsegundos)</p> <p><b><u>Funciones matemáticas</u></b></p> <p>min(x, y) max(x, y)</p> <p>abs(x) constrain(x, a, b)</p> <p>map(val, deBAJO, deALTO, aBAJO, aALTO)</p> <p>pow(base, exponente)</p> <p>sq(x) sqrt(x)</p> <p>sin(rad) cos(rad)</p> <p>tan(rad)</p> <p><b><u>Funciones números aleatorios</u></b></p> <p>randomSeed(semilla)</p> <p>long random(max)</p> <p>long random(min, max)</p> <p><b><u>Interrupciones Externas</u></b></p> <p>attachInterrupt(interrupt, func, [LOW, CHANGE, RISING, FALLING])</p> <p>detachInterrupt(interruption)</p> <p>interrupts()</p> <p>noInterrupts()</p> <p><b><u>Comunicaciones serie</u></b></p> <p>Serial.begin(baudios)</p> <p>int Serial.available()</p> <p>int Serial.read()</p> <p>Serial.flush()</p> <p>Serial.print(datos)</p> <p>Serial.println(datos)</p>
---	--	--

Tabla 3.1 Resumen lenguaje Arduino

#### 4. Familia Arduino

Hay más de 20 modelos de Arduino, con diferentes características en cuanto a tamaño, prestaciones, etc., esto permite utilizar la plataforma en una gran variedad de proyectos, más o menos complejos y utilizar el modelo Arduino más adecuado a cada necesidad. Así mismo el equipo Arduino sigue desarrollando plataformas, que permitan abordar otros proyectos o en otros campos, o para dar respuesta a demandas planteadas por los usuarios.

La tabla siguiente recoge un resumen de las características más significativas de los diferentes modelos de la plataforma Arduino.<sup>16, 17</sup>

	<b>Arduino UNO R3.</b> Microcontrolador ATmega328, a 16 MHz. Tiene 14 pines de entradas / salidas digitales, 6 de ellas también tienen función como salidas PWM, 6 entradas analógicas de 10 bits de resolución, 1 UART. Memoria flash 32 KB, SRAM 2 KB, EEPROM 1 KB. <sup>18</sup>
	<b>Arduino DUE.</b> Microcontrolador Atmel AT91SAM3X8E ARM Cortex-M3 de 32 bits, a 84 MHz. Tiene 54 entradas/salidas digitales, de las cuales 12 se pueden utilizar como salidas PWM, 12 entradas analógicas con resolución de 12 bits, 4 UARTs, dos salidas analógicas con DAC. Memoria flash 512 KB, SRAM 96 KB. <sup>19</sup>
	<b>Arduino Leonardo.</b> Microcontrolador ATmega32u4, a 16 MHz. Tiene 20 pines de entradas / salidas digitales, 7 de ellas también tienen función como salidas PWM, 12 entradas analógicas de 10 bits de resolución, 1 UART. Memoria flash 32 KB, SRAM 2,5 KB, EEPROM 1 KB. <sup>20</sup>
	<b>Arduino Mega2560 Rev3.</b> Microcontrolador ATmega2560, a 16 MHz. Tiene 54 pines de entradas / salidas digitales, 15 de ellas también tienen función como salidas PWM, 16 entradas analógicas de 10 bits de resolución, 4 UARTs. Memoria flash 256 KB, SRAM 8 KB, EEPROM 4 KB. <sup>21</sup>
	<b>Arduino ADK Rev3.</b> Microcontrolador ATmega2560, a 16 MHz. Tiene 54 pines de entradas / salidas digitales, 15 de ellas también tienen función como salidas PWM, 16 entradas analógicas de 10 bits de resolución, 4 UARTs. Memoria flash 256 KB, SRAM 8 KB, EEPROM 4 KB.

<sup>16</sup> <http://arduino.cc/en/Main/Products>

<sup>17</sup> <http://es.wikipedia.org/wiki/Arduino>

<sup>18</sup> <http://arduino.cc/en/Main/ArduinoBoardUno>

<sup>19</sup> <http://arduino.cc/en/Main/ArduinoBoardDue>

<sup>20</sup> <http://arduino.cc/en/Main/ArduinoBoardLeonardo>

<sup>21</sup> <http://arduino.cc/en/Main/ArduinoBoardMega2560>

Conector USB para comunicarse directamente con teléfonos móviles basados en Android. <sup>22</sup>	
	<b>Arduino Micro.</b> Similar a Arduino Leonardo. Microcontrolador ATmega32u4, a 16 MHz. Tiene 20 pines de entradas / salidas digitales, 7 de ellas también tienen función como salidas PWM, 12 entradas analógicas de 10 bits de resolución, 1 UART. Memoria flash 32 KB, SRAM 2,5 KB, EEPROM 1 KB. Su forma facilita su uso en placas de inserción o de prototipos. <sup>23</sup>
	<b>Arduino Mini.</b> Microcontrolador ATmega328, a 16 MHz. Tiene 14 pines de entradas / salidas digitales, 6 de ellas también tienen función como salidas PWM, 8 entradas analógicas de 10 bits de resolución. Memoria flash 32 KB, SRAM 2 KB, EEPROM 1 KB. Por su tamaño es apropiada para prototipos, uso en placas de inserción y para ocupar el mínimo espacio. <sup>24</sup>
	<b>Arduino Nano.</b> Microcontrolador ATmega328, a 16 MHz. Tiene 14 pines de entradas / salidas digitales, 6 de ellas también tienen función como salidas PWM, 8 entradas analógicas de 10 bits de resolución, 1 UART. Memoria flash 32 KB, SRAM 2 KB, EEPROM 1 KB. <sup>25</sup>
	<b>Arduino Ethernet Rev3 sin/con PoE.</b> Microcontrolador ATmega328 a 16 MHz. Tiene 9 pines de entradas / salidas digitales, 4 de ellas también tienen función como salidas PWM, 6 entradas analógicas de 10 bits de resolución. Memoria flash 32 KB, SRAM 2 KB, EEPROM 1 KB. Incorpora un controlador e interface Ethernet y conector para una tarjeta SD y se le puede añadir un módulo PoE. <sup>26</sup>
	<b>Arduino Esplora.</b> Plataforma similar a Arduino Leonardo, con la diferencia de que incorpora varios dispositivos sensores listos para usar: un sensor de luz, un joystick analógico, un potenciómetro lineal, un sensor de temperatura, un acelerómetro de tres ejes, un micrófono, cuatro pulsadores y salidas de sonido (buzzer), luz (LED RGB) y un zócalo para montar una pantalla TFT LCD en color. <sup>27</sup>
	<b>Arduino BT.</b> Microcontrolador ATmega328P a 8 MHz. Tiene 14 pines de entradas / salidas digitales, 6 de ellas también tienen función como salidas PWM, 6 entradas analógicas de 10 bits de resolución, 1 UART. Memoria flash 32 KB, SRAM 2 KB, EEPROM 1 KB. Incluye un módulo bluetooth para comunicación serie inalámbrica. <sup>28</sup>

<sup>22</sup> <http://arduino.cc/en/Main/ArduinoBoardADK>

<sup>23</sup> <http://arduino.cc/en/Main/ArduinoBoardMicro>

<sup>24</sup> <http://arduino.cc/en/Main/ArduinoBoardMini>

<sup>25</sup> <http://arduino.cc/en/Main/ArduinoBoardNano>

<sup>26</sup> <http://arduino.cc/en/Main/ArduinoBoardEthernet>

<sup>27</sup> <http://arduino.cc/en/Main/ArduinoBoardEsplora>

<sup>28</sup> <http://arduino.cc/en/Main/ArduinoBoardBT>

	<b>Arduino Fio.</b> Microcontrolador ATmega328 a 16 MHz. Tiene 14 pines de entradas / salidas digitales, 6 de ellas también tienen función como salidas PWM, 8 entradas analógicas de 10 bits de resolución, 1 UART. Memoria flash 32 KB, SRAM 2 KB, EEPROM 1 KB. Orientada a aplicaciones inalámbricas, incorpora un zócalo para montar un módulo XBee. <sup>29</sup>
	<b>Arduino Pro.</b> Microcontrolador ATmega328 a 8 MHz. Tiene 14 pines de entradas / salidas digitales, 6 de ellas también tienen función como salidas PWM, 6 entradas analógicas de 10 bits de resolución, 1 UART. Memoria flash 32 KB, SRAM 2 KB, EEPROM 1 KB. Está orientado para su uso en instalaciones semipermanentes. <sup>30</sup>
	<b>Arduino Pro Mini.</b> Microcontrolador ATmega168 a 8 MHz. Tiene 14 pines de entradas / salidas digitales, 6 de ellas también tiene función como salidas PWM, 8 entradas analógicas de 10 bits de resolución, 1 UART. Memoria flash 16 KB, SRAM 1 KB, EEPROM 512 B. Está orientado para su uso en instalaciones semipermanentes. <sup>31</sup>
	<b>Arduino LiLyPad.</b> Microcontrolador ATmega328 a 8 MHz. Tiene 14 pines de entradas / salidas digitales, 6 de ellas también tienen función como salidas PWM, 8 entradas analógicas de 10 bits de resolución. Memoria flash 16 KB, SRAM 1 KB, EEPROM 512 B. Diseñado para realizar dispositivos para llevar puestos, ( <i>wearables</i> ) ó montar electrónica en la ropa (e-textiles), se puede coser a la ropa con hilos conductores. <sup>32</sup>
	<b>Arduino LiLyPad USB.</b> Microcontrolador ATmega32u4 a 8 MHz. Tiene 9 pines de entradas / salidas digitales, 4 de ellas también tienen función como salidas PWM, y otras 4 tienen función de entradas analógicas de 10 bits de resolución. Memoria flash 32 KB, SRAM 2,5 KB, EEPROM 1 KB. Incluye lo necesario para su conexión USB. Diseñado para realizar dispositivos para llevar puestos o montar electrónica en la ropa de vestir, se puede coser a la ropa con hilos conductores. <sup>33</sup>
	<b>Arduino LiLyPad Simple.</b> Microcontrolador ATmega328, a 8 MHz. Tiene 9 pines de entradas / salidas digitales, 5 de ellas también tienen función como salidas PWM, y otras 4 tienen función de entradas analógicas de 10 bits de resolución. Memoria flash 32 KB, SRAM 2 KB, EEPROM 1 KB. Diseñado para realizar dispositivos para llevar puestos o montar electrónica en la ropa de vestir, se puede coser a la ropa con hilos conductores. <sup>34</sup>

<sup>29</sup> <http://arduino.cc/en/Main/ArduinoBoardFio>

<sup>30</sup> <http://arduino.cc/en/Main/ArduinoBoardPro>

<sup>31</sup> <http://arduino.cc/en/Main/ArduinoBoardProMini>

<sup>32</sup> <http://arduino.cc/en/Main/ArduinoBoardLilyPad>

<sup>33</sup> <http://arduino.cc/en/Main/ArduinoBoardLilyPadUSB>

<sup>34</sup> <http://arduino.cc/en/Main/ArduinoBoardLilyPadSimple>



**Arduino LiLyPad SimpleSnap.** Microcontrolador ATmega328, a 8 MHz. Tiene 9 pines de entradas / salidas digitales, 5 de ellas también tienen función como salidas PWM, y otras 4 tienen función de entradas analógicas de 10 bits de resolución. Memoria flash 32 KB, SRAM 2 KB, EEPROM 1 KB. Diseñado para realizar dispositivos para llevar puestos o montar electrónica en la ropa de vestir, incluye una batería y en lugar de cosido se sujeta mediante corchetes de presión por lo que se puede montar y desmontar fácilmente.<sup>35</sup>



**Arduino YUN.** Microcontrolador ATmega32u4, a 16 MHz. Tiene 20 pines de entrada/salida digitales, 7 de ellas también tienen función como salidas PWM, 6 canales analógicos, más otros 6 multiplexados sobre 6 pines digitales. Memoria flash 32 KB, SRAM 2 KB, EEPROM 1 KB. Incorpora también un Atheros AR9331, MIPS a 400MHz, con memoria externa RAM DDR2 de 64 MB y flash de 16 MB con una distribución Linux precargada y proporciona un interface WIFI otro Ethernet. Se programa vía WIFI, puede actuar como punto de acceso o como *router* e incluye zócalo para tarjeta SD.<sup>36</sup>



**Arduino Zero.** Microcontrolador ATSAM3U8, de 32 bits a 48 MHz. Tiene 14 entradas/salidas digitales, de las cuales 12 se pueden utilizar como salidas PWM y UART, 6 entradas analógicas con resolución de 12 bits, una salida analógica con DAC de 10 bit. Memoria flash 256 KB, SRAM 32 KB, EEPROM 16 KB. Se puede considerar como la evolución de Arduino UNO a una plataforma de 32 bits. Incluye un debugger.<sup>37</sup>



**Arduino robot.** Es un sistema completo en forma de robot, con dos placas Arduino: una de control y otra de motores, basadas en el procesador ATmega32u4, por lo que son similares a la placa Leonardo. Incluyen varios sensores y actuadores: dos potenciómetros, cinco pulsadores, un compás digital, cinco sensores de suelo, altavoz, dos motores, una pantalla LCD de color, lector de tarjetas SD.<sup>38, 39</sup>



**Intel Galileo Gen2.** Procesador Intel, X1000 Quark SoC, (todo el sistema en un chip), tipo Pentium, de 32 bits, a 400 MHz, 12 KB de SRAM incrustado y Memoria del sistema – 256 MB DDR3, 5, Hardware y software y conectores, compatible con Arduino UNO, lo que permite utilizar cualquier *shield* diseñada para Arduino y ejecutar los programas desarrollados para este. Como Arduino UNO y en el mismo lugar tiene 14 entradas/salidas digitales, 6 de ellas pueden funcionar como salidas PWM de 12 bits, 6 entradas analógicas.

<sup>35</sup> <http://arduino.cc/en/Main/ArduinoLilyPadSimpleSnap>

<sup>36</sup> <http://arduino.cc/en/Main/ArduinoBoardYun?from=Products.ArduinoYUN>

<sup>37</sup> [http://arduino.cc/en/uploads/Main/Arduino\\_Zero\\_front.png](http://arduino.cc/en/uploads/Main/Arduino_Zero_front.png)

<sup>38</sup> <http://arduino.cc/en/Main/Robot>

<sup>39</sup> [http://complubot.educa.madrid.org/proyectos/arduino/ArduinoRobot/es/ArduinoRobot\\_historia\\_es.php](http://complubot.educa.madrid.org/proyectos/arduino/ArduinoRobot/es/ArduinoRobot_historia_es.php)

Incluye varios puertos: mini-PCI Express, Ethernet, serie RS 232, USB raíz, USB cliente, conector para tarjeta Micro-SD y 8 MB de memoria flash NOR.<sup>40</sup>



**Arduino TRE.** Basado en el procesador Sitara AM3359AZCZ100 (ARM Cortex-A8) a **1 GHz**, 512 Mb de RAM DDR3L, es un pequeño ordenador que puede ejecutar Linux. Incluye 1 puerto Ethernet 10/100, 1 puerto USB 2.0 y 4 puertos USB Host, salida de vídeo HDMI y audio, zócalo para tarjetas Micro-SD. Incluye también un procesador ATmega32u4, el mismo que la placa Leonardo, por lo que también tiene las características y funcionalidad de esta, así pues son dos sistemas en uno. Desde la misma placa es posible editar, compilar y ejecutar los *sketches*.<sup>41</sup>

Tabla 4.1 Modelos Arduino

## 5. Shields

Las funcionalidades de Arduino se pueden extender conectándole a módulos para realizar distintas funciones: control de motores CC, conexión a redes Ethernet, conexión Bluetooth, etc., a los que en la jerga de este mundo se les conoce como *shields* (escudos), ya que se acoplan sobre la misma placa Arduino y permiten conectar varios módulos, unos sobre otros, formando una pila de ellos. Las más habituales son para Arduino UNO y en la tabla siguiente se recogen la mayoría de las *shields* oficiales, las que llevan el identificador Arduino.<sup>42</sup>



**Arduino Ethernet Shield.** Permite la conexión de Arduino a redes Ethernet 10/100 Mb, y por tanto a Internet, incluye un conector estándar RJ45. Se le puede añadir un módulo PoE, para que pueda suministrar alimentación a través del cable de red. Incluye un conector para una tarjeta microSD accesible a través de la red.<sup>43</sup>



**Arduino GSM Shield.** Basada en el modem radio GSM/GPRS M10, cuatribanda GSM850MHz, GSM900MHz, DCS1800MHz y PCS1900MHz, junto con una tarjeta SIM de cualquier operador de red de telefonía móvil, permite a Arduino conectarse a Internet a través de GPRS, a velocidades de hasta 85,6 Kbps, enviar y recibir SMS y hacer y recibir llamadas (hay que añadir micrófono y auricular).<sup>44</sup>

<sup>40</sup> <http://arduino.cc/en/ArduinoCertified/IntelGalileo>

<sup>41</sup> <http://arduino.cc/en/Main/ArduinoBoardTre>

<sup>42</sup> <http://arduino.cc/en/Main/Products>

<sup>43</sup> <http://arduino.cc/en/Main/ArduinoEthernetShield>

<sup>44</sup> <http://arduino.cc/en/Main/ArduinoGSMShield>

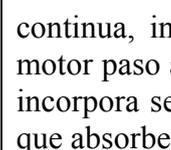
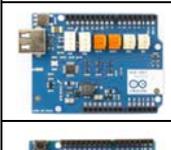
	<b>Arduino Wireless SD Shield.</b> Permite la comunicación inalámbrica de Arduino, utilizando módulos XBee ZigBee/802.15.4 de DIGI <sup>45</sup> , lo que permite alcances de hasta 30 m en interiores y 100 en exteriores. Incluye un conector para una tarjeta microSD. <sup>46</sup>
	<b>Arduino WiFi Shield.</b> Permite la conexión de Arduino a redes WiFi, con el estándar 802.11 b y g Ethernet 10/100 Mb, y por tanto a Internet. Permite la conexión a redes abiertas o encriptadas con WEP o WPA2. Incluye un conector para una tarjeta microSD, a la que se puede acceder a través de la red. <sup>47</sup>
	<b>Arduino Motor Shield.</b> La placa está basada en el circuito L298, que contiene dos excitadores en puente completo, lo que permite manejar cargas inductivas, como relés o solenoides, controlar la velocidad mediante PWM y el sentido de giro de hasta dos motores de corriente continua, independientes o mediante la combinación de ambos; puentes controlar un motor paso a paso bipolar. La corriente máxima por cada puente es de 2 A y así mismo incorpora sensores que permiten medir a través de las entradas analógicas, la corriente que absorbe cada motor. <sup>48</sup>
	<b>Arduino USB Host Shield.</b> Permite conectar dispositivos USB tales como teclados, joysticks, memorias, cámaras, dispositivos bluetooth, etc., a Arduino. <sup>49</sup>
	<b>Arduino Proto Shield.</b> Placa de prototipos con el formato y conectores de shields de Arduino UNO, pulsador de <i>reset</i> y 14 pines impresos para montaje de componentes SMD. <sup>50</sup>

Tabla 5.1 Shields Arduino oficiales

Además de las *shields* oficiales, hay disponibles cientos de ellas desde otros suministradores, y además de para las funciones indicadas, también hay para control de matrices de LED, visualizadores LCD, visualizadores LCD gráficos, pantallas TFT y táctiles, módulos GPS, control de relés, módulos RFID, visualizadores de 7 segmentos, acelerómetro, Radio FM, MIDI, adquisición y almacenamiento de datos, comunicación 4-20 mA, RTC, CAN-Bus, MP3 etc, otras funciones similares y diversos modelos de ellas. Jonathan Oxeer en su sitio web <http://shieldlist.org/> tiene la recopilación más

<sup>45</sup> <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/zigbee-mesh-module/xbee-zb-module>

<sup>46</sup> <http://arduino.cc/en/Main/ArduinoWirelessShield>

<sup>47</sup> <http://arduino.cc/en/Main/ArduinoWiFiShield>

<sup>48</sup> <http://arduino.cc/en/Main/ArduinoMotorShieldR3>

<sup>49</sup> <http://arduino.cc/en/Main/ArduinoUSBHostShield>

<sup>50</sup> <http://arduino.cc/en/Main/ArduinoProtoShield>

extensa de *shields* para Arduino, ordenadas por fabricantes o comercializadores y que también se puede buscar por función introduciendo los términos correspondientes en el campo de búsqueda, u obtener un listado completo si se busca sin introducir datos en campo de búsqueda.

Cuando se utilizan *shields*, una información importante es conocer que pines utiliza cada una de ellas y cuales quedan libres y pueden ser utilizados para otra o para otras funciones que requiera el usuario. Esta información está disponible en el sitio web de Arduino <sup>51</sup> para las *shields* oficiales y para otras de distintas fuentes que han proporcionado esta información.

Quien lo desee puede diseñar su propia *shield* y en el sitio web de Arduino encontrará información de utilidad para ello, así como conectar hardware a la placa Arduino. <sup>52, 53</sup>.

## 6. Otros entornos de programación de Arduino

Además del IDE de Arduino, hay multitud de aplicaciones software que permiten programar o utilizar la plataforma Arduino, muchas de ellas son gráficas y resultan muy adecuadas para iniciarse en la programación o en la robótica desde edades tempranas.

Algunos de estas aplicaciones son propiamente entornos de programación, con las que se genera código para cargar a la plataforma Arduino y que funcione de forma autónoma. Hay otro tipo de aplicaciones que utilizan la plataforma Arduino como un dispositivo de entrada y salida a la que simplemente envían o desde la que leen datos, lo que permite realizar funciones de control o de monitorización.

Seguidamente se describen algunas de estas aplicaciones, las tres aplicaciones que siguen son entornos de programación mientras que los 3 restantes son aplicaciones que pueden interactuar con la plataforma. En [3] se puede encontrar una información más completa sobre alguno de estos entornos y en otras monografías específicas del mismo autor.

### 6.1 Ardublock.

Ardublock, se distribuye como un applet de java que se añade a las herramientas del IDE de Arduino. La programación se realiza encajando piezas que son las instrucciones del lenguaje de Arduino y se configuran los parámetros correspondientes, generándose

---

<sup>51</sup> <http://playground.arduino.cc/Main/ShieldPinUsage>

<sup>52</sup> <http://playground.arduino.cc/Main/InterfacingWithHardware>

<sup>53</sup> <http://playground.arduino.cc/Main/ArduinoCoreHardware#Shields>

el código del programa. Se puede descargar y obtener más información en su sitio web <http://blog.ardublock.com/>



Fig. 6.1.1 Ventana de Ardublock

## 6.2 Minibloq

Minibloq, es una herramienta que nos permite programar varios tipos de plataformas Arduino. Mediante sencillos iconos, que contienen bloques de código, que se van encajando unos con otros y configurando sus parámetros, se puede crear programas complejos. La aplicación muestra también el código en lenguaje Arduino. La aplicación genera, compila y envía el código a la plataforma Arduino, es por tanto un entorno de programación.

Minibloq es software libre y con las fuentes disponibles para su modificación, que se puede obtener en <http://blog.minibloq.org>.

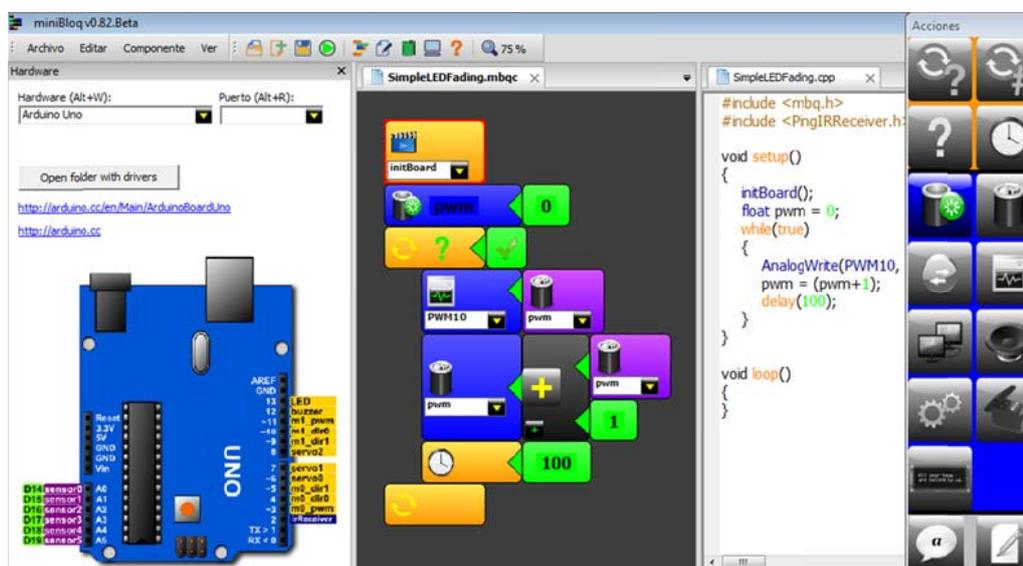


Fig. 6.2.1 Entorno de programación Minibloq

### 6.3 Modkit

Modkit, es un entorno de programación que permite programar varios tipos de plataformas entre ellas Arduino. Se van encajando piezas que son las instrucciones del lenguaje Arduino, que se configuran con los valores correspondientes, generandose el código correspondiente en lenguaje Arduino. Es una aplicación web esto es, se ejecuta desde el navegador.

El sitio web oficial del proyecto es <http://www.modkit.com/>

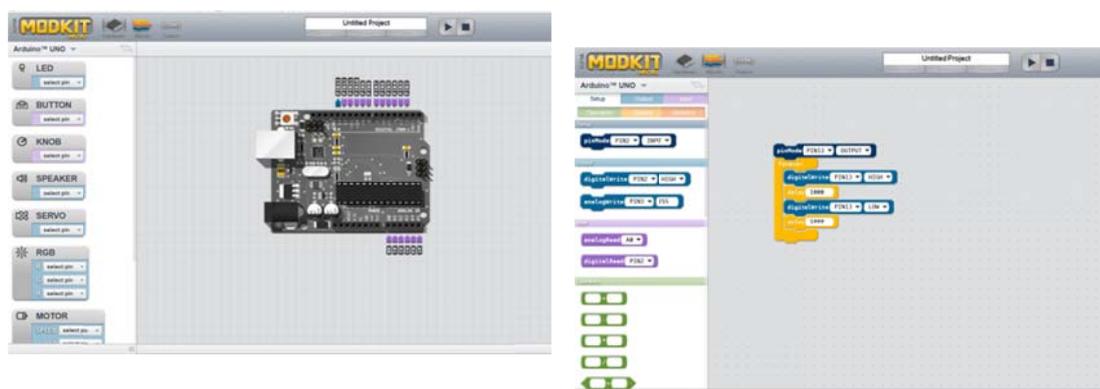


Fig. 6.3.1 (a) y (b) Entorno de programación MODKIT

### 6.4 Scratch

Scratch, es un software educativo de iniciación a la programación desarrollado por el *Lifelong Kindergarten Group* en el *Laboratorio de Medios del MIT*,<sup>54</sup> está orientado para niños en edad escolar, tiene una interfaz sencilla e intuitiva, donde el código se divide en bloques que se van encajando unos con otros para formar el programa. Permite crear historias interactivas, animaciones, juegos y música. El programa es gratuito y su código fuente está disponible para la comunidad propiciando la creación de programas, para varias plataformas, entre ellas Arduino con S4A.

Scratch S4A para Arduino es una modificación de Scratch, que permite manejar la plataforma Arduino de una forma sencilla, de este modo es posible interactuar también con el entorno. Scratch S4A ha sido desarrollado en el Citilab,<sup>55</sup> y desde su sitio web<sup>56</sup> se puede descargar y obtener información sobre este proyecto.

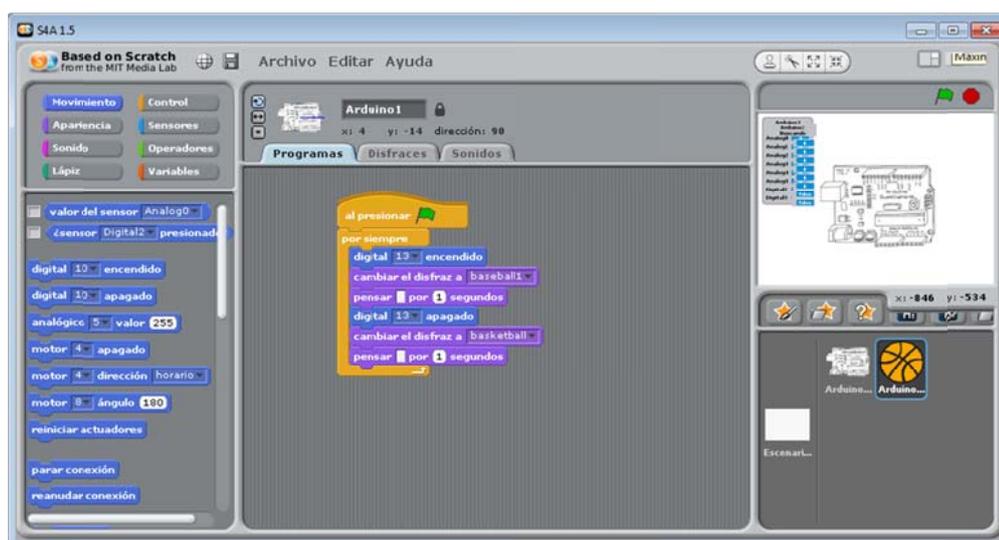


Fig. 6.4.1 Entorno de la aplicación Scratch S4A

### 6.5 Physical Etoys

Physical Etoys, es una herramienta para programación gráfica con fines didácticos que permite programar distintos dispositivos entre los que se encuentra Arduino. Está basado en Etoys,<sup>57</sup> lenguaje de programación gráfico orientado a la educación y la

<sup>54</sup> <http://scratch.mit.edu/>

<sup>55</sup> <http://citilab.eu>

<sup>56</sup> <http://s4a.cat>

<sup>57</sup> [http://en.wikipedia.org/wiki/Etoys\\_%28programming\\_language%29](http://en.wikipedia.org/wiki/Etoys_%28programming_language%29)

enseñanza de los niños. Physical Etoys es una herramienta de programación visual que une el mundo virtual de los ordenadores con el mundo físico, permite interactuar con la tarjeta Arduino, leyendo y escribiendo datos. Es software abierto y se puede obtener en su sitio web <sup>58</sup>.

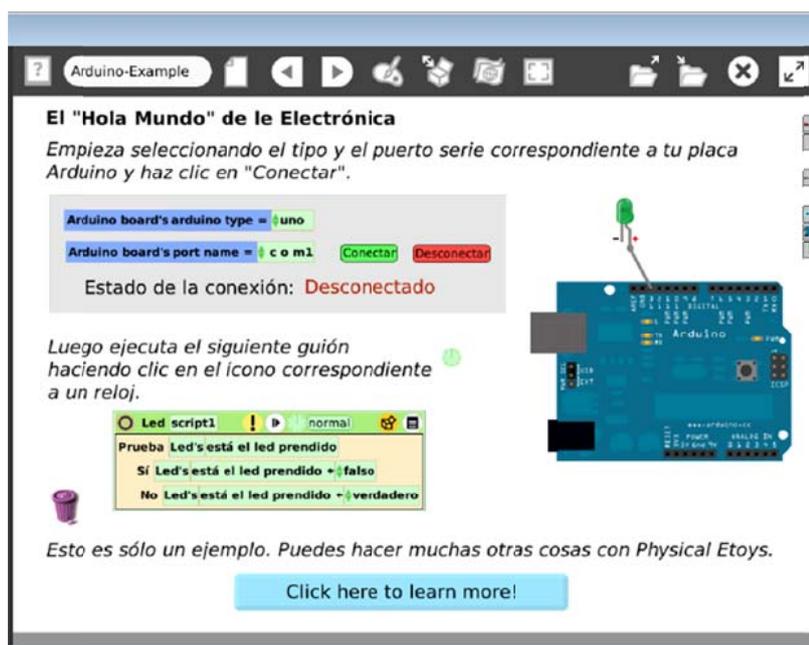


Fig. 6.5.1 Entorno de la aplicación Physical Etoys

## 6.6 Ardulab

**ArduLab** es un entorno de trabajo que permite interactuar con la plataforma Arduino, y crear un laboratorio virtual en el que el usuario puede experimentar con sensores y actuadores conectados a la plataforma, sin necesidad de programación y realizar actividades orientados al aprendizaje de conceptos sencillos relacionados con la electrónica y la robótica.

ArduLab es un desarrollo del Grupo Avanzado del Aula de Robótica Complubot <sup>59</sup> se distribuye bajo licencia *freeware*, <sup>60</sup> y se puede conseguir desde su portal web. <sup>61</sup>

<sup>58</sup> <http://tecnodacta.com.ar>

<sup>59</sup> [Aula de Robótica Complubot](http://Aula.de.Robótica.Complubot)

<sup>60</sup> [licencia freeware](http://licencia.freeware)

<sup>61</sup> [http://complubot.educa.madrid.org/proyectos/arduino/ardulab/ardulab\\_index.php](http://complubot.educa.madrid.org/proyectos/arduino/ardulab/ardulab_index.php)

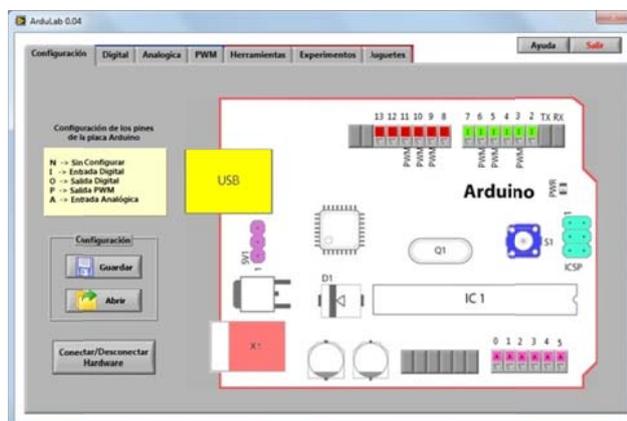


Fig. 6.6.1 Entorno de la aplicación ArduLab

Una relación muy completa de entornos de programación y de aplicaciones software con las que Arduino puede comunicarse y transferir datos entre ellos están disponibles en el sitio web de Arduino <sup>62</sup>, <sup>63</sup>.

## 7. Clones y compatibles Arduino

Arduino es una plataforma de hardware abierto (OHS) de la que todos los esquemas son públicos y pueden utilizarse tal cual o modificarse, por ello es posible montar una placa Arduino o modificar el diseño añadiéndole o limitando algunas prestaciones y mantener la compatibilidad con Arduino, de modo que se puedan utilizar el entorno de desarrollo de Arduino y el firmware ya que también son abiertos OSS. Esto junto con la numerosa comunidad de usuarios de la plataforma, hace que se hayan desarrollado multitud de plataformas compatibles con Arduino, bien casi clónicas, o añadiéndole alguna funcionalidad para un fin específico, según las necesidades de cada cual. A menudo a estas plataformas, sus creadores les suelen dar un nombre que suele incluir el sufijo -duino, de modo que se respeta el nombre oficial de Arduino pero se pueden relacionar con este, a modo de ejemplo, algunas muy conocidas son: Freeduino, Seeduino, Boarduino, Sanguino, Roboduino.

Se pueden encontrar comercializadas plataformas similares a Arduino que además incorporan conectividad bluetooth <sup>64</sup>, o Ethernet <sup>65</sup>, o ZigBee <sup>66</sup> etc., u orientadas al

<sup>62</sup> <http://playground.arduino.cc/Main/DevelopmentTools>

<sup>63</sup> <http://playground.arduino.cc/Main/InterfacingWithSoftware>

<sup>64</sup> [http://www.dfrobot.com/index.php?route=product/product&product\\_id=1044](http://www.dfrobot.com/index.php?route=product/product&product_id=1044)

manejo de LEDS <sup>67</sup>, matrices de LED, visualizadores LCD o de 7 segmentos, etc., manejo de motores, o para robots, que incluyen además sensores y actuadores, <sup>68</sup>

Muchos de los diseños también persiguen obtener un sistema más económico, reduciendo en lo posible los costes de fabricación <sup>69</sup>, <sup>70</sup> o haciendo diseños más simples <sup>71</sup>, o que facilitan la autoconstrucción, utilizando por ejemplo diseños del circuito impreso de simple cara, <sup>72</sup>, <sup>73</sup>, <sup>74</sup>, o simplemente con un tamaño más pequeño <sup>75</sup>.

En el sitio web de Arduino, se orienta como hacer un desarrollo propio de Arduino, o si se prefiere de como colaborar en la mejora y expansión del proyecto Arduino, <sup>76</sup>, así mismo se da información sobre multitud de sistemas parecidos a Arduino <sup>77</sup> y el sitio web [www.freeduino.org](http://www.freeduino.org), aunque no está muy actualizada recoge gran cantidad de diseños similares a Arduino, al igual que en [http://en.wikipedia.org/wiki/List\\_of\\_Arduino\\_boards\\_and\\_compatible\\_systems](http://en.wikipedia.org/wiki/List_of_Arduino_boards_and_compatible_systems)

## 8. Arduino en el espacio, ArduSat

Arduino también está en el espacio, ArduSat-1 and ArduSat-X son dos satélites idénticos, que fueron lanzados al espacio, el 4 de agosto de 2013, <sup>78</sup> desde el Centro Espacial de Tanegashima en Japón, bordo de la nave de carga HTV4 Kounotori 4, <sup>79</sup> propulsada por un cohete H-II B de la agencia espacial japonesa Jaxa, en una de las misiones de reabastecimiento de la Estación Espacial Internacional, (ISS). ArduSat <sup>80</sup> es un satélite de código abierto, diseñado bajo el estándar Cubesat, construido y gestionado

---

<sup>65</sup> <http://www.freetronics.com/products/etherten>

<sup>66</sup> <http://www.logos-electro.com/zigduino/>

<sup>67</sup> [http://www.toastedcircuits.com/html/product/Lightduino\\_5.0.html](http://www.toastedcircuits.com/html/product/Lightduino_5.0.html)

<sup>68</sup> <http://www.inexglobal.com/products.php?type=micro&cat=AVR&model=popbotxt>

<sup>69</sup> <http://richduino.com/>

<sup>70</sup> <http://www.pablogindel.com/fotos/?album=1&gallery=15>

<sup>71</sup> <http://playground.arduino.cc/Main/1000PadsLuigino>

<sup>72</sup> <http://unitaperiferica.altervista.org/index.php/elettronica-2/8-easyduino>

<sup>73</sup> <http://m0xpd.blogspot.co.uk/2013/02/what-do-i-know.html>

<sup>74</sup> <http://vonkonow.com/wordpress/2012/10/nanino-the-diy-friendly-arduino/>

<sup>75</sup> <http://www.microduino.cc/>

<sup>76</sup> <http://www.arduino.cc/en/Main/Policy>

<sup>77</sup> <http://playground.arduino.cc/Main/SimilarBoards>

<sup>78</sup> <http://blog.arduino.cc/2013/08/12/ardusat-successfully-launched-in-space/>

<sup>79</sup> <http://innovacion.ticbeat.com/sensores-espaoles-de-camino-al-espacio-en-los-satlites-ardusat/>

<sup>80</sup> [https://www.kickstarter.com/projects/575960623/ardusat-your-arduino-experiment-in-space?ref=nav\\_search](https://www.kickstarter.com/projects/575960623/ardusat-your-arduino-experiment-in-space?ref=nav_search)

por la compañía aeroespacial Nanosatisfi<sup>81</sup> mediante un proyecto de *crowdfunding* en *Kickstarter*,<sup>82</sup>.

ArduSat es un cubo de 10 cm de lado, con un peso de 1 Kg, su carga útil la forman más de 25 sensores distintos, que se controlan y gestionan mediante plataformas Arduino. Entre los sensores con los que cuenta ArduSat, se encuentran cámaras, espectrómetros ópticos, contadores Geiger, magnetómetros de 3 ejes, giróscopos digitales de 3 ejes, acelerómetros de 3 ejes, sensores de temperatura de diversos tipos, sensores de luminosidad.

El uso de ArduSat está abierto al público interesado para llevar a cabo investigaciones y experimentos utilizando los diversos sensores con que cuenta. El periodo para la realización de experimentos es de una semana y la comunicación con el satélite se realiza a través de los servidores de Nanosatisfi. La vida prevista para los satélites ArduSat es de 2 años.

## 9. Fritzing

Fritzing es una herramienta muy útil para documentar los diseños con Arduino u otras plataformas. Permite documentar fácilmente el montaje de prototipos sobre placas de inserción, añadir comentarios, etc. A partir del montaje sobre placa de inserción, se genera el esquema eléctrico del montaje y también es posible realizar el diseño del circuito impreso.

Este programa es una iniciativa de código abierto, con licencia GNU GPL, para el código y Creative Commons para los ejemplos que incorpora. La página oficial del proyecto es: <http://fritzing.org/> desde donde se puede descargar.

El entrono tiene un área de trabajo donde se sitúan y se conectan los componentes del circuito, si bien se pueden conectar los componentes directamente, lo habitual es hacerlo sobre una placa de inserción. Las librerías incluyen componentes discretos, integrados, módulos, plataformas de microcontroladores, etc.

---

<sup>81</sup> <http://www.nanosatisfi.com/>

<sup>82</sup> <https://www.kickstarter.com/projects/575960623/ardusat-your-arduino-experiment-in-space/posts>

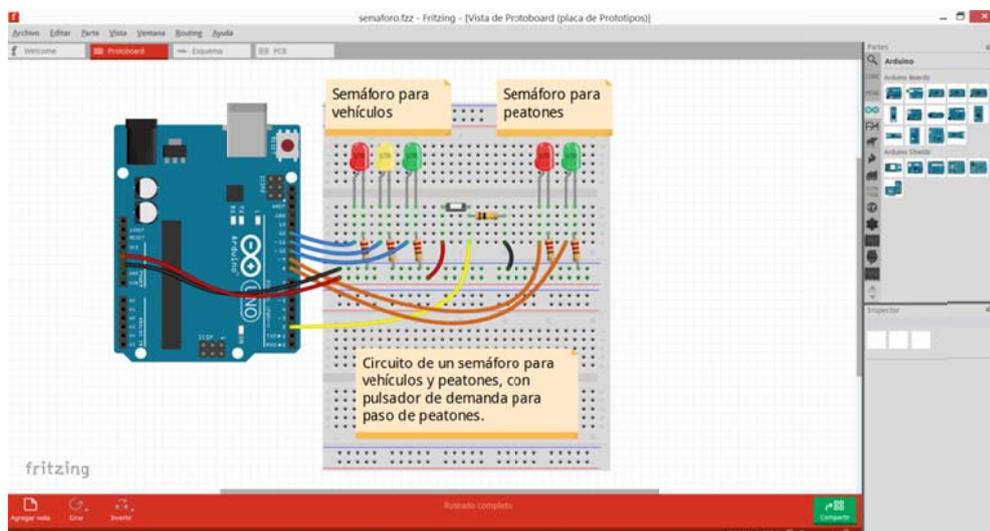


Fig. 9.1 Diseño de semáforo, realizado a partir del ejemplo incluido en Fritzing

Otra posibilidad que brinda la aplicación es la de añadir notas en la pantalla de trabajo, lo que resulta útil para incluir aclaraciones sobre el diseño, el montaje, componentes, etc. o de su funcionamiento, lo que sin duda contribuye a mejorar la documentación de un diseño.

A la vez que se va realizando el montaje en la placa de prototipos, el programa va generando de forma simultánea el esquema eléctrico del mismo.

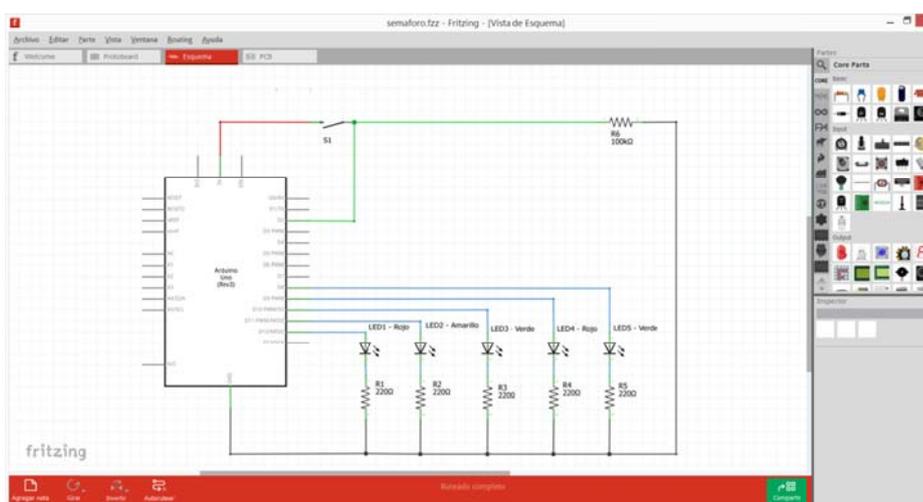


Fig. 9.2 esquema eléctrico del semáforo, realizado a partir del ejemplo incluido en Fritzing.

Así mismo también a la vez que se realiza un montaje, el programa va situando los componentes, esta vez con su encapsulado sobre una placa de circuito impreso, señalando las conexiones eléctricas entre ellos. Una vez distribuidos los componentes sobre la placa se puede indicar al programa que realice el ruteo de las pistas por una o ambas caras. Finalizado este, es posible modificar el diseño propuesto por el programa y desviar pistas o cambiarlas de cara y por último verificar que se cumplen las reglas de diseño.

La aplicación permite elegir el formato de salida del fichero de PCB, según el método de fabricación de PCB que se vaya a utilizar, fotorresist, transferencia de toner, o formato Gerber para producción profesional.

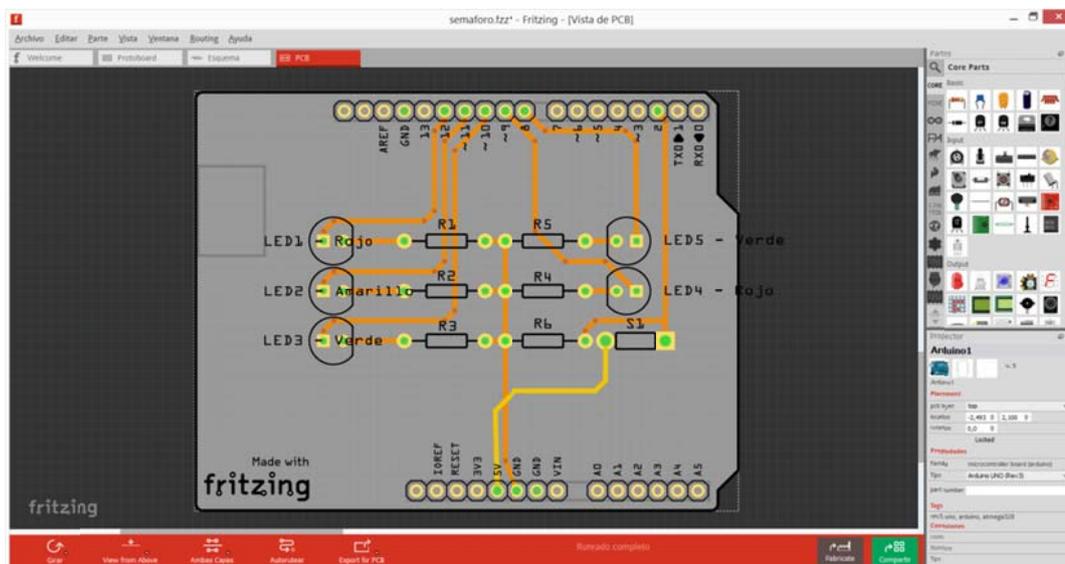


Fig. 9.3 PCB final del semáforo del ejemplo incluido en Fritzing.

El programa también proporciona una ventana de edición para editar programas para Arduino, aunque de momento no permite la compilación y el envío del programa a la plataforma, pero sí su edición.

## 10. Conclusiones

Desde que en 2005 apareciera el primer modelo, Arduino es un referente del hardware libre y aglutina una gran comunidad de usuarios que desde intereses muy diversos comparten sus creaciones. En este periodo la plataforma Arduino ha evolucionado en diferentes modelos, para dar respuesta a las demandas de los usuarios

y que puedan elegir el que mejor se adapte a las necesidades. De forma simultánea a la evolución de la plataforma Arduino y al crecimiento del número de usuarios, también ha ido ampliándose el número de entornos de desarrollo y de aplicaciones, programas y software de escritorio que se pueden comunicar e interactuar con esta plataforma y que expanden y facilitan su utilización.

La gran comunidad de usuarios y la filosofía del hardware libre propician que se puedan comercializar tanto las plataformas Arduino como las *shields* compatibles con un coste muy bajo, prácticamente el coste de producción.

Finalmente señalar que las plataformas de hardware abierto como Arduino resultan de gran utilidad en el mundo educativo en multitud de campos tales como robótica, electrónica, física, programación, creaciones artísticas, etc., y en todos los niveles educativos, desde la educación primaria a la universidad.

## 11. Bibliografía

- [1]- Brian w. Evans. Arduino programming notebook. Edición española Traducción: Jose Manuel Ruiz Gutierrez. Adaptacion: Jose Manuel Escuder Martinez. ver. 1.2 de 18/08/2011. [http://www.ardumania.es/wp-content/uploads/2011/10/Arduino\\_programing\\_notebook\\_ES.pdf](http://www.ardumania.es/wp-content/uploads/2011/10/Arduino_programing_notebook_ES.pdf)
- [2] Enríquez Herrador, Rafael. Guía de Usuario de Arduino I.T.I. Sistemas, Universidad de Córdoba. 13 de noviembre de 2009. [i52enher@uco.es](mailto:i52enher@uco.es). Este trabajo está publicado bajo la licencia: Creative Commons Attribution-Noncommercial-Share Alike 3.0.
- [3] Ruiz Gutierrez, J.M. Herramientas de programación gráfica de Arduino (3-11-2011). Programacion gráfica de Arduino.pdf. <http://josemanuelruizgutierrez.blogspot.com.es/>

### Otros recursos consultados y no referenciados

Portal oficial del proyecto Arduino <http://arduino.cc/> consultado septiembre de 2014.

Entrevista a Massimo Banzi <http://readwrite.com/2014/05/12/arduino-massimo-banzi-diy-electronics-hardware-hacking-builders> consultado septiembre de 2014.

Arduino The Documentary <http://vimeo.com/18390711> consultado septiembre de 2014.

Portal kickstarter: <https://www.kickstarter.com/projects/575960623/ardusat-your-arduino-experiment-in-space/posts> consultado septiembre de 2014.



Esta obra está licenciada bajo la Licencia Creative Commons Atribución-CompartirIgual 4.0 Internacional. Para ver una copia de esta licencia, visita <http://creativecommons.org/licenses/by-sa/4.0/>.