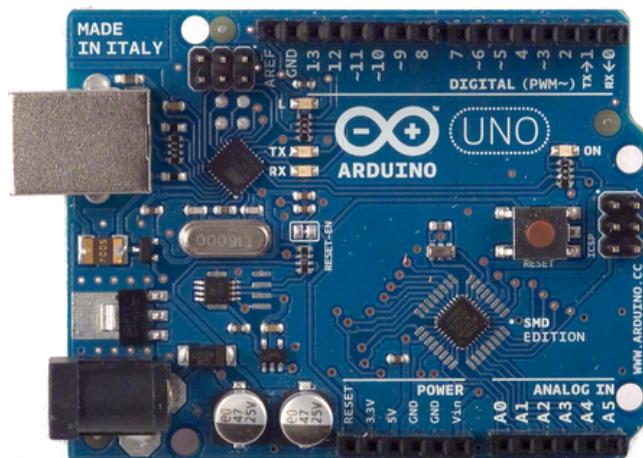


Entornos Gráficos para la Programación

Plataforma Open Hardware Arduino



José Manuel Ruiz Gutiérrez
Noviembre 2011 Ver1.0

Índice

1. Entornos de monitorización esclavos.

- 1.1. [Etoys \(Squeak\)](#)
- 1.2. [S4A \(Scratch\)](#)
- 1.3. [Labview](#)
- 1.4. [Firefly](#)
- 1.5. [MyOpenLab](#)

2. Entornos de programación autónomos

- 2.1. [Minibloq](#)
- 2.2. [Ardublock](#)
- 2.3. [Amici](#)
- 2.4. [ModKit](#)
- 2.5. [VirtualBreadBoard](#)

3. Otros entornos

- 3.1. [Fritzing](#)

4. Apéndice

1.- Entornos de monitorización esclavos.

1.1 ARDUINO + Etoys

[Etoys \(Squeak\)](#)

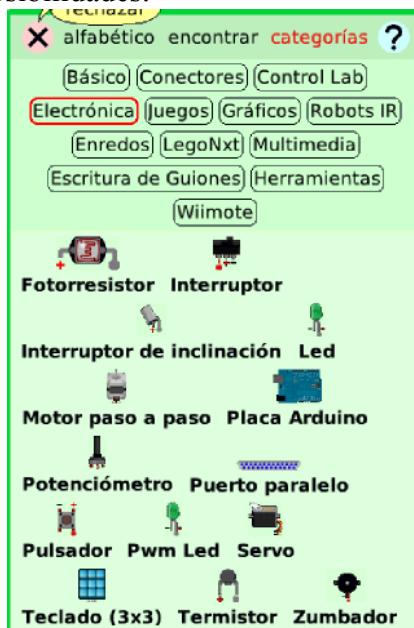
<http://tecnodacta.com.ar/gira/projects/physical-etoys>

Physical Etoys: Control de Arduino desde Etoys (Squeak)

Descripción.

Physical Etoys es una herramienta de programación visual que une el mundo virtual de las computadoras con el mundo físico, con el programa se puede programar fácilmente los objetos del mundo real (por ejemplo, robots) para realizar tareas de interés, o se pueden mover objetos gráficos en la pantalla a través de variables recogidas del mundo físico.

Todo es tan sencillo como coger algunos objetos de la paleta que representan los distintos elementos tanto de control de la tarjeta Arduino como otros auxiliares para dibujar o representar valores en la pantalla y unirlos en un “puzzle” que finalmente permitirá la realización de una aplicación de “visualización y/o control”. Cada uno de los objetos dispone de una serie de parámetros a los que se designara unos valores de acuerdo con nuestras preferencias y los requerimientos de la aplicación. No hay necesidad de tener conocimientos de programación, sólo tiene que utilizar el ratón y explorar el sistema y sus posibilidades.



Physical Etoys es en realidad una "extensión" de Etoys: un entorno de autor para la creación de aplicaciones multimedia y de simulación mediante la representación visual realizado por las mismas personas que crearon Smalltalk. Etoys es un estupendo

software que ayuda a los niños a explorar su propia creatividad de manera divertida y educativa, por lo que hemos querido dar Etoys la posibilidad de interactuar con objetos del mundo real, tales como kits de robótica y joysticks innovadoras. Etoys, está abierto, es gratis, y es muy divertido.

Se trata de una herramienta realizada en el Grupo de Investigación en Robótica Autónoma delo CAETI GIRA que permite el diálogo con la tarjeta Arduino en el modo lectura y escritura de datos. Este proyecto es muy importante dado que incorpora Arduino a una herramienta de muy amplia difusión y gran reputación en el mundo del software académico tal como es Squeak



En la dirección siguiente se puede encontrar el software y los manuales básicos de uso.
<http://tecnodacta.com.ar/gira/projects/physical-etoys/>

El entorno Physical Etoys permite trabajar con distintos modelos de Robots y tarjetas de control:

1. Arduino.
2. Nintendo Wiimote.
3. Puerto paralelo.
4. RoboSapien v2.
5. Roboquad.
6. I-Sobot.
7. Lego Mindstorms Nxt.

Haciendo titilar un led

Empieza seleccionando el tipo y el puerto COM correspondiente a tu placa Arduino y haz clic en "Conectar".

Arduino board's boardType = duemilanove w/ atmega328
Arduino board's portName = com6 Conectar Desconectar

Estado de la conexión: Desconectado

Luego ejecuta el siguiente guión haciendo clic en el ícono correspondiente a un reloj.

↓

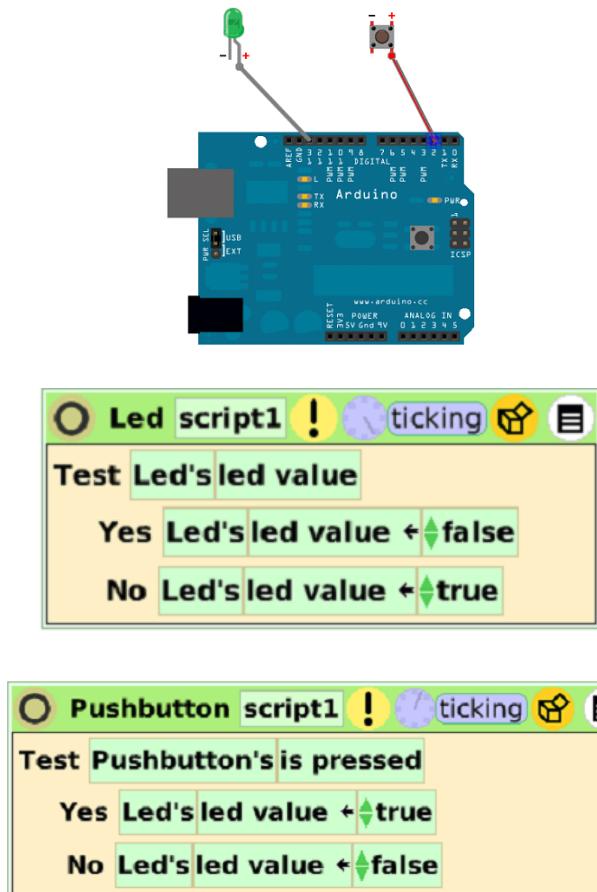
O Led blink ! pausado

Prueba Led's led value
Sí Led's led value ← false
No Led's led value ← verdadero

Arduino board diagram showing a green LED connected to digital pin 13 and ground, and a pushbutton connected to digital pin 2 and ground.

Este es sólo un ejemplo. Puedes hacer muchas otras cosas con Physical Etoys.

Haz clic aquí para aprender más!



1.2 Scratch + Arduino

S4A (Scratch)

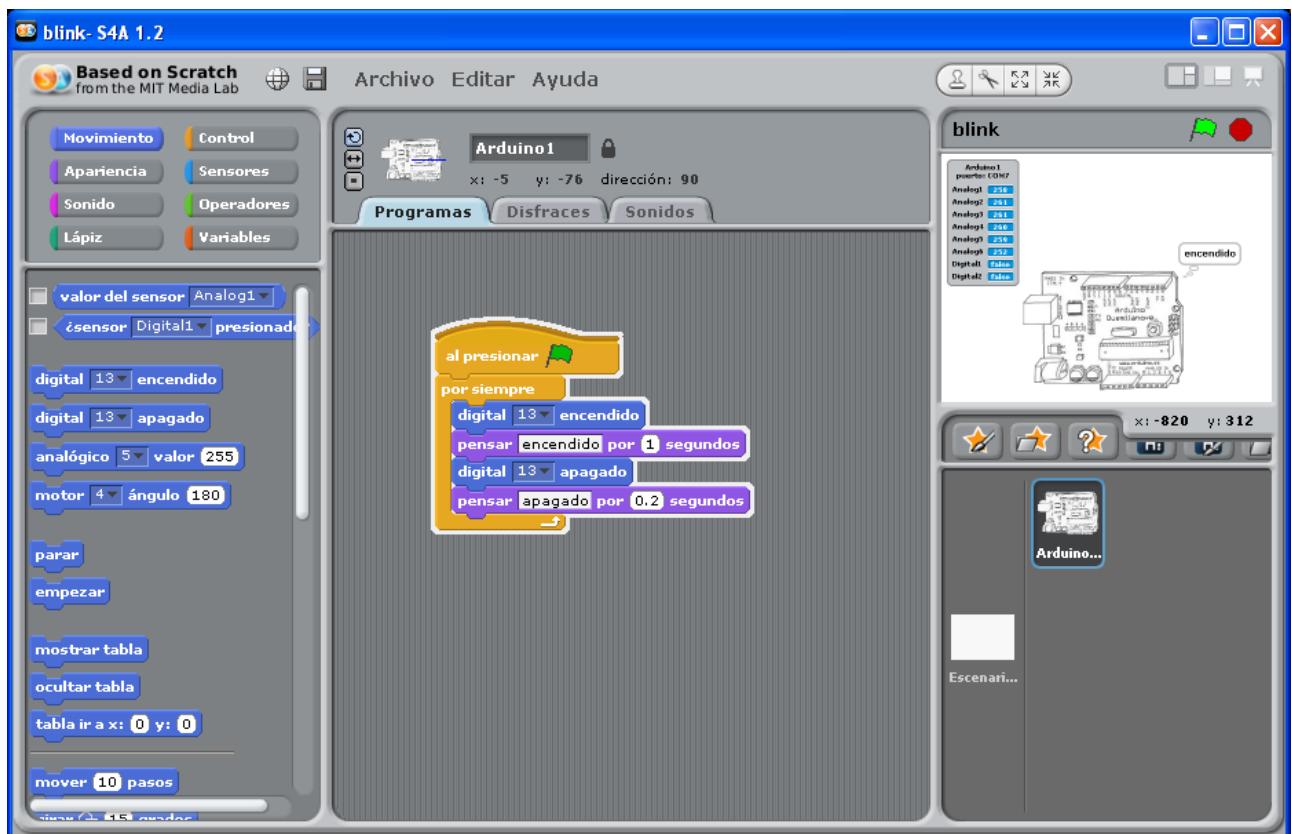
http://seaside.citilab.eu/scratch?_s=hvXNe4iKvwhZ3YaO&_k=6rrYXWDBfmL1NRUy



[Citilab](#)

Introducción

Citilab pone a disposición de la comunidad Arduino una aplicación basada en Scratch para programar de manera gráfica Arduino. La aplicación se llama s4a (ficheros para descargar [S4A](#) y [Firmware](#))



S4A es una modificación de Scratch que proporciona una programación sencilla de la plataforma abierta de hardware [Arduino](#). Incluye nuevos bloques para controlar

sensores y actuadores conectados a Arduino. También hay una tabla que informa del estado de los sensores similar a la PicoBoard.

Ha sido desarrollada para atraer a la gente al mundo de la programación. Su objetivo es también proporcionar una interfaz de nivel alto para programadores de Arduino con funcionalidades como la interacción de varias placas a través de eventos de usuario.

Características

- Los objetos de la librería Arduino ofrecen bloques para funcionalidades básicas del microcontrolador, escrituras y lecturas tanto analógicas como digitales, además de alguna de más nivel. Puedes encontrar bloques para controlar servomotores de rotación continua Parallax.



- Se pueden crear objetos Arduino a través de 3 maneras distintas en el entorno de Scratch. Puedes elegir entre crear una nueva conexión o usar una ya existente. Esta característica permite a los objetos virtuales Arduino funcionar de forma colaborativa usando la misma conexión (el objeto físico). El objeto Arduino encontrará él mismo el puerto USB donde la placa esté conectada.
- S4A interactúa con Arduino enviando el estado de los actuadores y recibiendo el de los sensores cada 75 ms, por lo tanto el ancho de pulso ha de ser mayor que este período. Este intercambio de información se efectúa usando el protocolo de la PicoBoard, el cual ya está implementado en un programa específico (llamado firmware) en la placa. Encontrareis instrucciones de cómo cargarlo a través del [entorno Arduino](#).
- Funciona con las versiones Duemilanove/Diecimila y Uno, quizás también con otras pero no las hemos testeado aún. También se puede controlar una placa de manera inalámbrica si se añade un módulo RF como por ejemplo XBee. Una

característica importante es que se puede crear un proyecto utilizando tantas placas como puertos USB haya disponibles.

- S4A es compatible con Scratch, se puede trabajar con proyectos Scratch y también con la PicoBoard. Sin embargo, no se pueden compartir [proyectos](#) ya que va en contra de los [términos de uso](#) de Scratch.
- Por otra parte, la configuración de entrada/salida aún está siendo desarrollada, así que por ahora los componentes tienen que conectarse de una forma concreta. Dicha configuración ofrece 6 entradas analógicas (pines analógicos), 2 entradas digitales (pines digitales 2 y 3), 3 salidas analógicas (pines digitales 5, 6 y 9), 3 salidas digitales (pines 10, 11 y 13) y 4 salidas especiales para conectar servomotores de rotación continua Parallax (pines digitales 4, 7, 8 y 12).

Creadores: S4A ha sido desarrollado por Marina Conde, Víctor Casado, Joan Güell, José García y Jordi Delgado con la ayuda del [Grupo de Programación Smalltalk del CitiLab](#). Para informar de errores o simplemente darnos sugerencias podéis escribir a: scratch@citilab.eu.

Futuras versiones

Estamos desarrollando un applet de java para cargar el firmware de una forma fácil, además de seguir trabajando en la configuración de las entradas y salidas. También queremos ofrecer la posibilidad de enviar el código principal del proyecto a la placa para no depender del entorno Arduino.

Para las descargas visitad la [sección de descargas](#) para obtener la última versión de S4A y del firmware además de ver algunos vídeos de demostración.

Modo de instalación.

1. Para instalar S4A +Arduino se deberán seguir los siguientes pasos.
2. Primero instalar el [software S4A](#) una vez que se haya descargado del su lugar de origen.
3. Se deberá instalar en la tarjeta Arduino el [Firmware](#) correspondiente que facilita la comunicación con S4A. Esto se realiza cargando el fichero firmware en el IDE de Arduino y después descargándolo sobre la tarjeta.
4. Finalmente se ejecuta S4A y se realiza el diseño haciendo uso de las librerías de bloques correspondientes una parte de las cuales se encarga de la lectura y escritura de datos en la tarjeta de acuerdo siempre con la configuración que

establezca el firmware. A continuación se detallan estas configuraciones de E/S que no olvidemos que no se pueden modificar desde S4A.

ENTRADAS/SALIDAS

- *salidas digitales (pines digitales 10,11 y 13)*
- *salidas analógicas (pines digitales 5, 6 y 9)*
- *entradas analógicas (todos los pines analógicos de entrada)*
- *entradas digitales (pines digitales 2 y 3)*
- *servomotores RC (pines digitales 4, 7, 8 y 12)*

1.3 Arduino + Labview

<https://decibel.ni.com/content/groups/labview-interface-for-arduino>

Introducción.

Arduino ya tiene conexión con Labview lo cual le confiere una gran potencialidad dado que estamos hablando de una herramienta de gran capacidad y muy extendida tanto en el mundo académico como en el industrial y de laboratorio.

La conexión que actualmente ofrece Labview es en modo esclavo. Solo es posible visualizar y realizar control de las E/S de arduino den modo esclavo.

Esta forma de trabajo, por tratarse de Labview, es interesante para usarla en el prototipado de aplicaciones de instrumentación en as que la tarjeta Arduino juega el papel de un sencillo y versátil equipo de adquisición de datos a un costo muy bajo.

Instalación de la interfaz de LabVIEW con Arduino

La configuración de la interfaz de LabVIEW para Arduino es un proceso de seis pasos que sólo tendrá que completar una sola vez. Por favor, siga las siguientes instrucciones para comenzar a crear aplicaciones con la interfaz de LabVIEW para Arduino.

(Para una breve descripción de la interfaz de LabVIEW para Arduino ver post Michaels aquí).

1. Instalar LabVIEW

Si usted compró el paquete de LabVIEW y Arduino de Sparkfun.com puede instalar LabVIEW desde el DVD incluido.

Si usted no posee una copia de LabVIEW se puede descargar e instalar la versión de evaluación de 30 días aquí.

2. Instalar los controladores VISA NI

[Windows Download.](#)

[Linux Download.](#)

[Mac Download.](#)

3. Instale JKI VI Package Manager (VIPM) Community Edition (gratuita).

[All Operating Systems..](#)

4. Instalar la interfaz de LabVIEW para Arduino como se describe en [KB 5L38JQYG](#)

5. Conecta tu Arduino a su PC como se describe en [KB 5INA7UYG](#)

6. Carga la interfaz de LabVIEW para Firmware Arduino en su Arduino como se describe en [KB 5LPAQIYG](#)

Ahora está listo para utilizar la interfaz de LabVIEW para Arduino. Haga clic [aquí](#) para ver algunos ejemplos para ayudarle a empezar.

Ejemplo: Leer una entrada Analógica

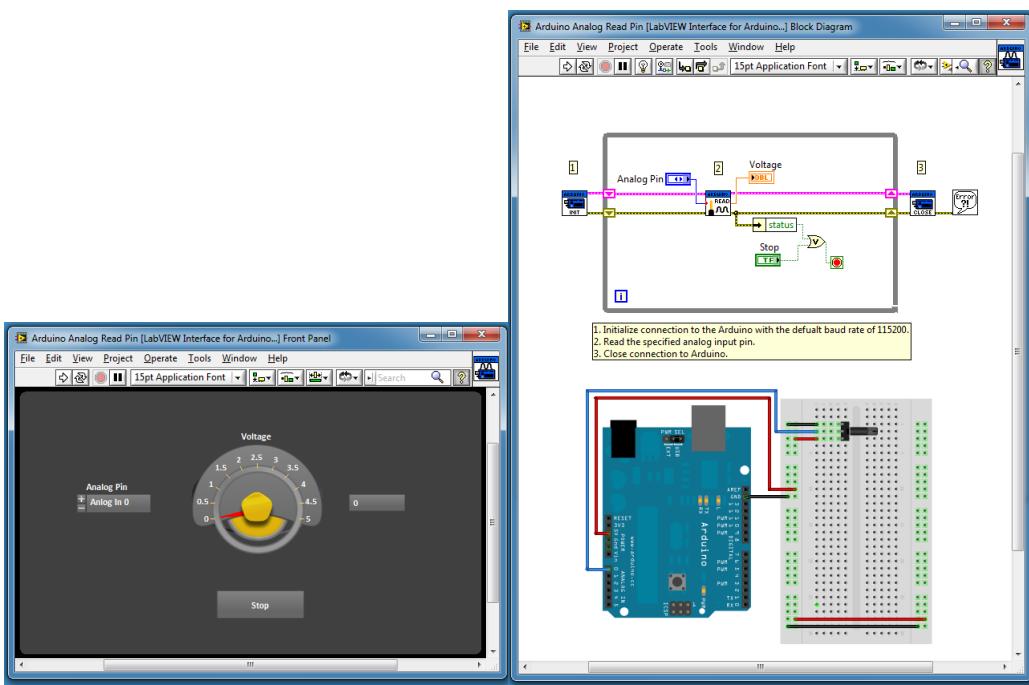
Descripción:

Este ejemplo muestra cómo leer una entrada analógica en el Arduino usando la interfaz de LabVIEW para Arduino.

Tutorial:

1. Asegúrese de que tiene instalada la interfaz de LabVIEW para Arduino.
2. Abra la lectura analógica Pin Ejemplo VI (búsqueda de "Arduino" en el buscador de ejemplo una vez que la herramienta está instalada).
3. En el panel frontal, seleccione el pin de Arduino analógico que desea leer.
4. Construir el esquema del circuito se muestra en el diagrama de bloques.
5. Conecta tu Arduino a su PC con un cable USB. Opcionalmente se puede especificar el puerto COM asociado con la placa Arduino a la terminal de recursos VISA de la Init.vi para disminuir el tiempo necesario para LabVIEW para conectarse a la placa Arduino.
6. Haga clic en Ejecutar.
7. Gire el potenciómetro y observe el medidor y pantalla digital en la actualización del panel frontal.

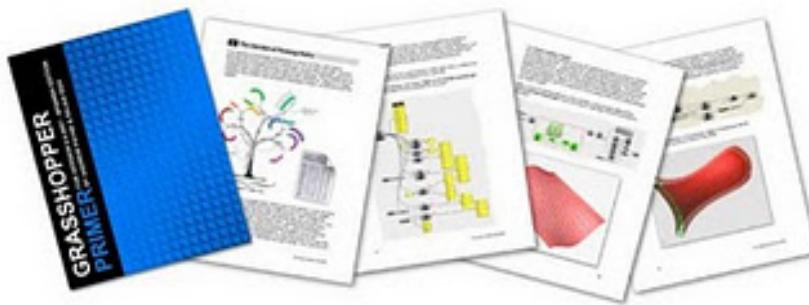
Entornos Gráficos de Programación para la Plataforma Arduino.
José Manuel Ruiz Gutiérrez Ver1.0



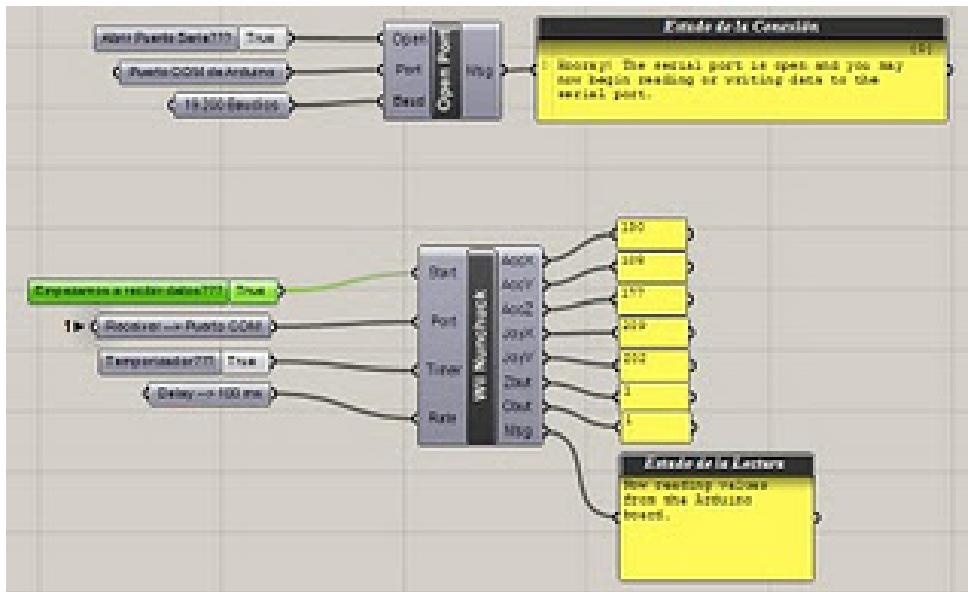
1.4 Conexión de Arduino a Rhino

Introducción

Ya es posible conectar Arduino al poderoso entorno gráfico Rhinoceros a través del conocido plugin **Grasshopper** que es un entorno gráfico muy versátil y fácil de utilizar que se utiliza para la programación de eventos y gobierno de imágenes de Rincho.



La herramienta en cuestión se trata de una librería llamada Firefly que incorpora un conjunto de bloques de función que se conectan con Arduino.

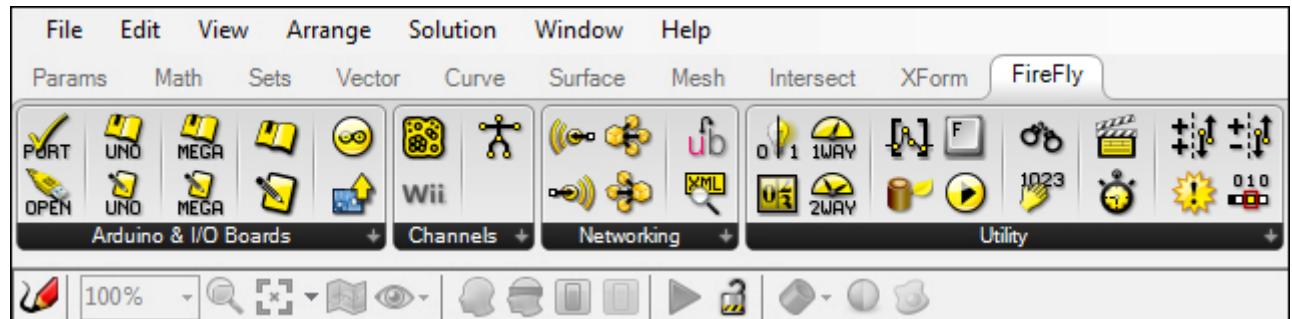


Tutorial de conexión con Grasshopper + Arduino Sketechs del tutorial
Firefly for Grasshopper / Arduino from Jason K Johnson on Vimeo.

Rhino+Grasshoper+Firefly+Arduino

Grasshoper es un pugin de Rhino que añada a esta herramienta la posibilidad de realizar “scripts” de forma grafica para ejecutarlos en Rhino. Una de las librerías de Grasshoper se llama Firefly y esta pensada para interactuar con Arduino en el gobierno de las E/S tanto analógicas como digitales d ela plataforma Arduino.

A continuación se muestra una imagen de la plantilla de componentes disponibles en la actualidad en la barra de herramientas de Firefly.



Descripción básica de los componentes de Firefly

Componentes relacionados con Arduino



Los puertos disponibles

Comprueba los puertos de comunicación disponibles en el momento actual. Si usted tiene una o más placas Arduino conectado a su ordenador, se devolverá el entero asociado a cada puerto. Se puede verificar el número de COM haciendo clic en Herramientas> puerto serie en el interior del IDE Arduino (placa debe estar conectado a la computadora).



Abrir / cerrar el puerto

Abrir o cerrar la conexión del puerto serie. Se utiliza en conjunto con los bloques de lectura y escritura para establecer la conexión entre el Grasshopper y Arduino.



Uno Leer

Este componente va a leer los valores de todos los pines correspondientes digitales y analógicos en las tarjetas Arduino del tipo el Uno Arduino, Duemilanove, Diecimilla, o Lillypad. Nota: Este componente está destinado a ser usado en unión con el FireFly Sketch Uno Firmata.



Uno Escribir

Este componente escribe los valores en todos los pines correspondientes a un Arduino Uno Arduino, Duemilanove, Diecimilla, o la junta Lillypad. Nota: Este componente está destinado a ser usado en unión con el FireFly Sketch Uno Firmata.



Mega Leer

Este componente va a leer los valores de todos los pines correspondientes digitales y analógicos en las tarjetas Arduino Mega. Nota: Este componente está destinado a ser usado en unión con el FireFly Sketch Uno Firmata.



Escribir Mega

Este componente escribe los valores en todos los pines correspondientes a un Arduino Mega. Nota: Este componente está destinado a ser usado en unión con el FireFly Sketch Uno Firmata.



Lea serie (Genérico)

Devuelve cualquier valor de cadena que viene a través del puerto serie. Este componente no tiene ningún tipo de Arduino específicos asociados con él, lo que significa que usted puede crear su propio código para enviar un valor de cadena de la Arduino con el comando Serial.println.



Escribir serie (Genérico)

Escribir cualquier valor de cadena al puerto serie. Usted tendrá que proporcionar también un código de Arduino para procesar el valor de cadena que viene a través del puerto serie (normalmente manejados por el dibujo Firmata).



Arduino Code Generator

Este componente intentará convertir cualquier definición de Grasshopper en el código de Arduino compatible (C++) sobre la marcha. Basta con conectar la cadena de salida desde cualquiera de los componentes Arduino Uno / Mega a la

entrada del generador de código y ver el código de construcción al crear su definición de Grasshopper. Funciona mediante la detección de 'upstream' cualquiera de los componentes de la componente Arduino Uno / Mega. Cuando se detecta un componente, comprueba si tiene o no tiene una función de ese componente. Si lo hace, se añade la función de la parte superior del código, y llama a la función adecuada dentro de la configuración de main () o loop () funciones. Si la biblioteca de funciones no contiene una definición de un determinado componente de Grasshopper, que proporcionará una advertencia. Este código puede ser al mismo tiempo guardado como un archivo. PDE que se abre en el IDE Arduino.



Subir a la I / O

Este componente se intenta cargar cualquier croquis de Arduino a su E / S de la Junta. Si el boceto Arduino ruta del archivo es válido (puede ser desde el generador de código), puede hacer clic en el ícono de la flecha para subir el dibujo a la placa. Un montón de cosas que sucede detrás de las escenas, pero se debe crear un Makefile dinámico y convertir el archivo. PDE en un archivo. Cpp (C++) y luego en el archivo. Hex (código legible por máquina para el microcontrolador). NOTA: WinAVR es necesario instalar para que la carga de E / S componente Junta para que funcione correctamente. WinAVR (pronunciado "siempre") es un conjunto gratuito de ejecutables, herramientas de código abierto de desarrollo de software que incluye avr-gcc (compilador), avrdude (programador), entre otras cosas. Puede descargar la última versión de WinAVR en <http://sourceforge.net/projects/winavr/files/WinAVR/>

Canales



reacTIVision Listener

Listener Reactivision señala a la posición y rotación de cada marcador Reactivision fiduciales. Este componente está destinado a ser usado en conjunto con el software Reactivision, disponible de forma gratuita desde <http://www.reactivision.com/downloads>. Marcadores fiduciales conjunto disponible de <http://reactivision.sourceforge.net/archivos #>. Nota: Usted debe tener el software Reactivision y funcionamiento de la cámara activada para utilizar este componente.



Wii Nunchuck Leer

Este componente le permite leer todos los valores de los sensores (acelerómetro de 3 ejes, joystick, botones C y Z) de la Wii Nunchuck. Nota: Este componente está destinado a ser usado en conjunto con el Wii Nunchuck Sketch Arduino.



Kinect Skeleton Rastreador

Este componente permite realizar un seguimiento de hasta dos personas (esqueletos) con el sensor de Xbox Kinect. Nota: Este componente sólo se ejecutará si tienes Windows 7, Rhino 5.0 beta versión de 32 bits. Este componente está siendo lanzado como descarga por separado, por ahora, y hay instrucciones especiales para trabajar con este componente incluye en esta descarga.

La creación de redes



OSC Listener

Abierto de control de sonido (OSC) son esencialmente los mensajes con un formato especial Protocolos de datagramas de usuario (UDP) transmisiones. El componente de escucha OSC obras de apertura de un puerto UDP y la escucha de los mensajes OSC en la red. Se recibe un mensaje que crea una estructura de árbol de datos que contiene el nombre de la OSC y el valor (puede ser flotantes, enteros, cadenas, etc.) Si un nuevo mensaje con un nombre diferente, una nueva rama se crea para ese mensaje. Sin embargo, si se recibe un mensaje y una rama ya contiene ese nombre, entonces el valor será actualizado. Use el elemento o elemento de árbol para recuperar el último valor de cada mensaje en el árbol de datos. Un restablecimiento de selección puede ser utilizado para borrar los valores árbol de datos.



OSC del remitente

Como se mencionó anteriormente, los mensajes OSC tienen un formato especial mensajes que puedan difundirse a través de una red. Con el fin de enviar un mensaje de OSC, es necesario conocer la dirección IP del dispositivo que está tratando de enviar el mensaje. También debe especificar un único puerto UDP. Una vez que haya especificado la dirección IP y el puerto, se puede conectar cualquier tipo de datos (deslizadores, interruptores, paneles, etc) y automáticamente el formato de los datos en un mensaje de OSC y de difusión que a la dirección especificada.



UDP Listener

User Datagram Protocol (UDP) es un modelo de transporte para enviar / recibir mensajes a través de una red. El componente de escucha UDP simplemente abre un puerto UDP y la escucha de los mensajes entrantes. Una dirección IP no es necesario para escuchar los mensajes (sólo para el envío de mensajes). El valor de retorno es una cadena.



UDP del remitente

El componente del remitente UDP envía ningún mensaje de cadena a una dirección IP especificada en la red a través de un puerto determinado. Esto puede ser útil en el envío de mensajes a otros programas que se ejecutan en otro equipo (como el procesamiento) u otros dispositivos móviles.



Pachube Leer

Leer un archivo de cualquier elemento Pachube en línea. Requerirá de una URL en Internet y su Pachube clave de la API. Visita www.pachube.com para más información.



XML Buscar

Este componente le permite buscar a través de un archivo XML (ya sea desde la web o desde un directorio local) para una etiqueta de elemento específico, que devuelve el valor de esa etiqueta en una lista.

Componentes de servicios públicos



Y Flip Flop

Y el flip-flop se diferencia de la NI flip-flop en el sentido de que la puerta de salida está determinado por su estado de salida actual, así los estados de sus dos entradas. Así, en un flip-flop y, la salida no va a cambiar si ambos S y R son falsas, pero cambiará a su complemento, si ambas entradas se establecen en true.



Explosión

El componente de explosión registra las condiciones de borde para una entrada booleana. El T-salida se registrará un valor verdadero momentánea cuando la entrada cambia a un verdadero estado. El F-salida se registrará un valor verdadero momentánea cuando la entrada cambia de nuevo a un estado falso. Este componente es el equivalente a un componente de Bang en otros programas como Pd, Max / MSP, y VVVV. Nota: Este componente se requiere el uso de un componente temporizador (Parámetros / especiales / Timer).



Blink binario

Oscila 0 y 1 está basado en un modelo de entrada de números enteros. Se puede

utilizar para crear intrincadas (alto / bajo) parpadeando patrones que se enviará a los componentes Escribir serie de LED de control, etc



Buffer

Almacenar un conjunto de valores en la memoria intermedia basado en un tamaño de array dado. Por ejemplo, un array del 0 al 10 se almacenarán los últimos 10 valores. Sin embargo, si utiliza un dominio de buffer de 5 a 15 continuación, el búfer se almacenan los valores 10, pero sólo las que se produjeron cinco posiciones antes de que el valor de entrada actual.



Constreñir

Restringe un número a un rango específico numérico (de dominio).



Contador

Cuenta hacia arriba y hacia abajo. Se puede establecer la dirección para contar, el tamaño de paso, y los límites de restricción para el contador. Nota: Este componente se requiere el uso del componente GH_Timer (Params / Special / Temporizador).



Los datos de registro

El componente de registro de datos permite almacenar un conjunto de valores en una lista. Si la longitud de la lista se ajusta a "0", entonces el registro se almacenan los valores indefinidamente (o hasta que se detenga la grabación o restablecer el registro). De lo contrario, la longitud de la lista se determinará cómo muchos valores para almacenar en el registro. Si la entrada Activación se establece en "1" True, los valores comenzarán a re-grabación de los valores de registro anterior después de haber llegado al final de la longitud de la lista.



Fader One Way

Desaparecer de un valor a otro basado en un solo intervalo de tiempo (ms).



Fader de dos vías

Desaparecer entre un valor mínimo y máximo basado en la fade in y fade out intervalo (ms).



Velocidad de cuadros

Devuelve el tiempo en milisegundos ya que los datos han sido actualizado, así como un marco estimado por intervalos de un segundo.



Es la clave prensada

Este componente detecta si una tecla en el teclado ha sido presionado. Sólo tiene que especificar un valor de cadena que le gustaría poner a prueba, y conecta un componente Timer para este componente. La salida devolverá True si la tecla fue pulsada justo. Este componente reconoce todas las entradas de alfa numérico, teclas de función, y teclas especiales (como arriba, abajo, izquierda, derecha, control, desplazamiento, espacio, etc)



NOR Flip Flop

La NI flip-flop tiene dos entradas, es decir, una entrada de set (S) y una entrada de reset (R). Una simple representación de un flip-flop es un par de acoplamiento cruzado NOR, es decir, la salida de una compuerta está ligado a una de las dos entradas de la puerta del otro y viceversa. Por lo tanto, el establecimiento de la puerta de salida en falso requiere $R = S = \text{verdadero}$ y falso, si bien establece la puerta de salida a la verdad requiere $S = R = \text{verdadero}$ y falso.



Reproducción

El componente de reproducción recuperar valores de un archivo de texto (tipos aceptables de archivo:... Txt, csv, y dat) y los devuelve individualmente a una velocidad determinada (s). Puede introducir una tasa de fotogramas múltiples para mantener al mismo tiempo un seguimiento de varias salidas a diferentes intervalos.



Liso

Suave (o promedio) un valor de entrada sobre la base de un nivel de muestreo (número de valores).



Detección de estado

El componente de detección de Estado se utiliza cuando se quiere hacer algún tipo de acción basado en las veces que se oprime un botón. Se comprueba el número de veces que el valor de la corriente ha pasado de (o viceversa) alto a bajo y si el modulo de que el valor del contador es igual a cero. Si es así, se despliega la salida de detección de estado. Por lo tanto, si el valor del módulo es de 2, entonces cada vez que otros que se pulsa el botón,

el valor del estado cambiará. O si el valor del módulo es de 4, y luego cada cuarto de empujar el botón de disparo el valor del estado de cambio.



Cronómetro

El componente intenta ser la réplica de la funcionalidad de un verdadero cronómetro. Cuando se establece la entrada de inicio toggle en true, el temporizador comenzará a contar, y cuando lo vuelva a falso, entonces se devolverá el tiempo en milisegundos para ese intervalo. Puede repetir esta pauta, agregando más tiempo cada vez que el cronómetro está en marcha y se detuvo. Usted puede configurar el intervalo de cronómetro por alternar la entrada de reset.

Primeros pasos para comenzar a trabajar con Firefly

Requisitos previos

Con el fin de comenzar a trabajar con Firefly, usted querrá asegurarse de tener instalado el siguiente software en su ordenador:

1. Descargar [Firefly](#)
2. [Rhinoceros 4.0](#) (o [5.0 beta WIP](#))
3. [Service Release 9](#) (de Rhino)
4. [Grasshopper](#) (versión 0.8.0050 o superior)
5. [Arduino IDE](#) (Integrated Development Environment)

Paso 1

(Instalar la barra de herramientas de Firefly - sólo se hará una vez)

Lo primero que deberá hacer es copiar y pegar los archivos Firefly.gha y Firefly_X.gha (que se encuentra dentro del archivo zip arriba) y el archivo libTUIO.dll en el propio Directorio de Grasshopper.

- Haga clic en Archivo> Carpetas especiales> Carpeta de componentes.
- Eliminar las versiones anteriores de Firefly (ya sea mayor. Gha archivos o la libTUIO.dll) si ya se han instalado en su ordenador.
- A continuación, abra la carpeta de instalación Firefly (en el archivo zip. Que acaba de descargar) y copiar los tres archivos (**Firefly.gha**, **Firefly_X.gha**, y **libTUIO.dll**) en la carpeta de componentes que se abrió desde dentro de Rhino.

Nota: Los tres de estos archivos tienen que ser copiar / pegar en el directorio de Firefly para funcionar correctamente. Este proceso sólo se tiene que hacer una vez! Asegúrese de eliminar cualquier archivos antes de Firefly de instalación.

A continuación se lanzara la aplicación Rhino y el la herramienta Grasshopper, escribiendo "Grasshopper" en la línea de comandos. Esto abrirá el editor de Grasshopper.

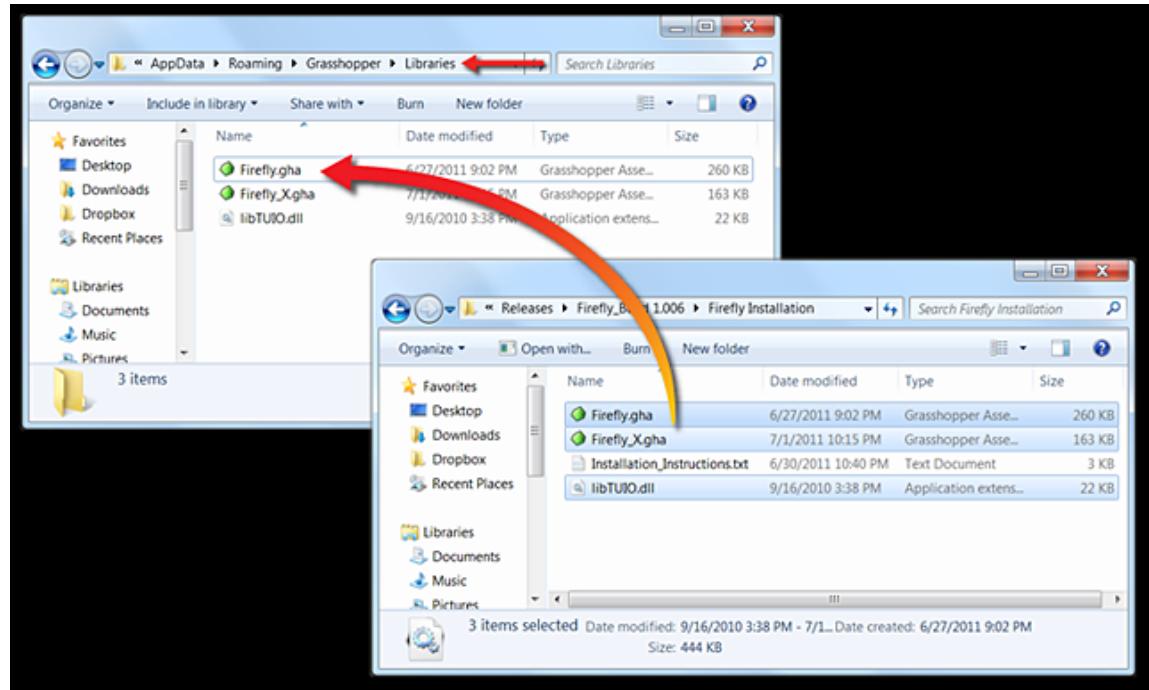
Si ha instalado una versión anterior de Firefly y que habían colocado los archivos de creación en el directorio C:\Archivos de programa (x86)\Rhinoceros 4.0\Plug-ins\Grasshopper directorio \ Componentes de ... ¡por favor, elimínelos!

Ahora, Lanzar Rhino y Grasshopper

Esta vez, al iniciar Grasshopper, debería ver una nueva pestaña de Firefly en la parte superior de la pantalla junto al resto de librerías de Grasshopper.
Editor. ¡Felicitaciones! Usted ha instalado la barra de herramientas Firefly.

Nota: Si ubiese algún problema para ejecutara Firefly y se ha cargado correctamente en Rhino beta 5.0, realice los siguientes pasos:

- Abra la carpeta Componentes Especiales (en la que acaba de copiar los 3 archivos) y haga clic en cada archivo y seleccionar "Propiedades". En la ficha General, marque la casilla para "desbloquear" el archivo. Haga esto para cada uno de los tres archivos.
- Ahora, Reiniciar Rhino y Grasshopper



Paso 2

(Cargar el Firmware de Comunicación con Firefly - sólo se necesita una vez)

Vamos a suponer que está instalado el IDE de Arduino (el software necesario para cargar programas a la placa Arduino) y se han descargado los controladores adecuados.

Con el fin de comenzar a trabajar con Firefly, lo primero que tendrá que programar el Arduino para crear la conexión de entrada y salida con Grasshopper. Afortunadamente, ya hemos realizado el trabajo más complicado, por lo que no tiene que preocuparse de nada. En primer lugar, copiar / pegar la carpeta llamada "Firefly_Uno_Firmata" (que contiene un fichero en formato pde para Arduino llamado *Firefly_Uno_Firmata.pde*) en la carpeta de Sketchbook Arduino que normalmente se encuentra en uno de dos lugares:

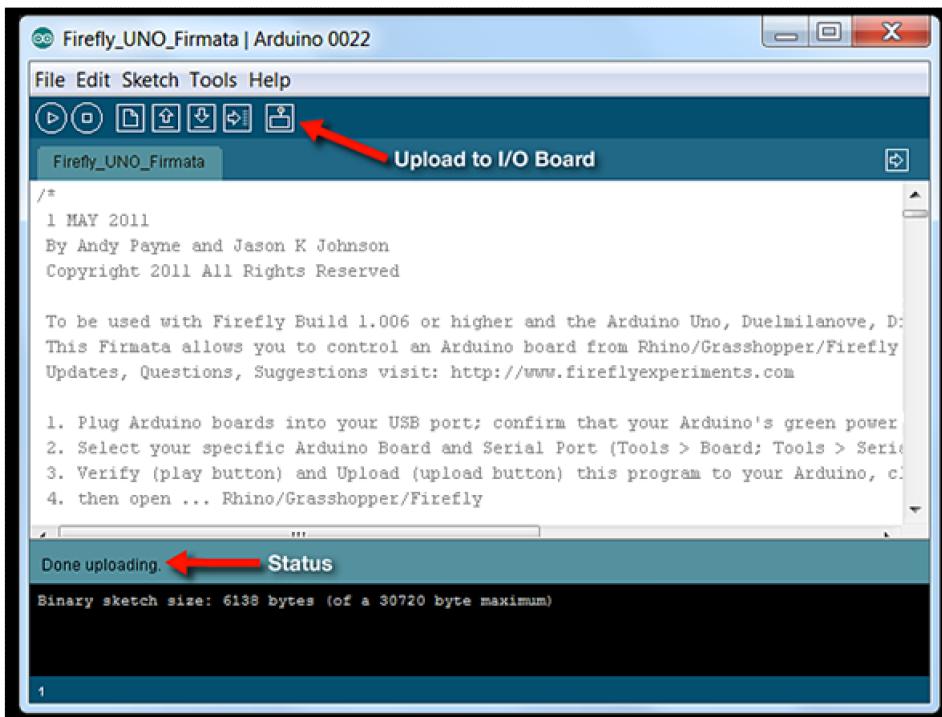
- Mis documentos \ Arduino
- Nombre de usuario \ Documents \ Arduino

Ahora, ejecutar Arduino 0022 aplicación y abrir el ficheru Firefly Firmata: File> Sketchbook> "Firefly_Uno_Firmata.pde". Haga clic en Tools> Board y asegúrese de que tiene asignada la tarjeta Arduino adecuada. Ahora, haga clic en Tools> Serial port y asegúrese de que está conectada la tarjeta en el COM adecuado # (estos se asignan automáticamente al conectar la tarjeta al ordenador).

Nota: Es conveniente tomar nota del número del puerto serie, le vendrá bien. Esta puerto será el que luego utilizaremos cuando controlaremos Arduino desde Grasshopper.

Lo único que queda es cargar el sketch a la tarjeta. Seleccione la opción "Upload" botón (en la parte superior). Esperar unos segundos – se verán parpadear los leds RX y TX de la tarjeta. Si la carga es correcta, el mensaje "Done uploading." Aparecerá en la barra de estado. Se deberá mantener el USB conectado al Arduino - así se podrá comunicar Firefly con los sensores, motores y LED conectado a la placa.

¡Enhorabuena! La tarjeta Arduino está listo para "dialogar" con Grasshoper.



Paso 3

(Instalar WinAVR - Necesario para subir sketches desde Grasshopper)

WinAVR ahora es necesario instalarlo para poder utilizar la nueva carga de E / S componente desde Firefly. WinAVR es una suite de herramientas de código abierto de desarrollo de software que incluye avr-gcc (compilador) y avrdude (programador), entre otras cosas. Para instalar la última versión de WinAVR:

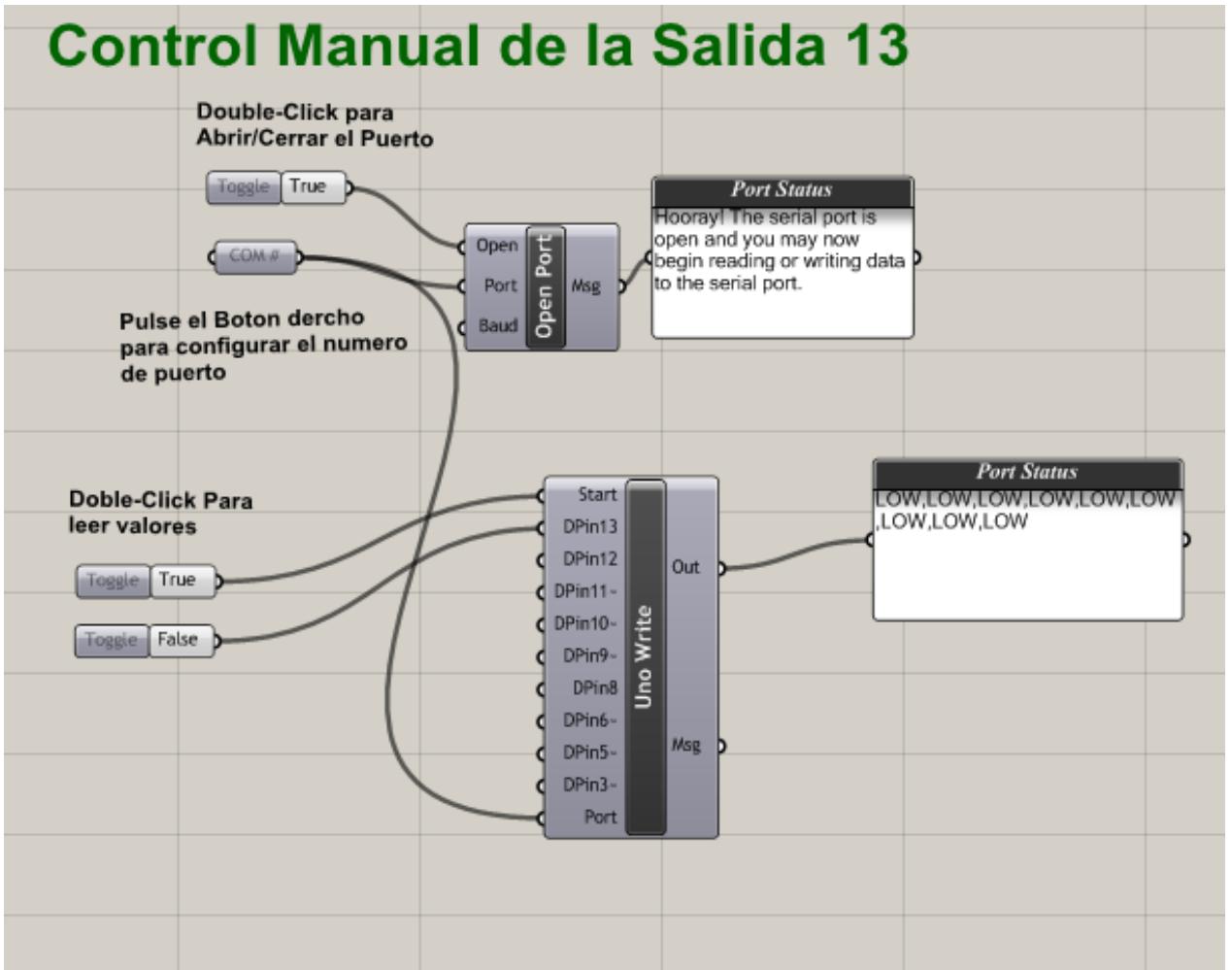
- Ir a: <http://sourceforge.net/projects/winavr/files/WinAVR/>
- Haga clic en la carpeta llamada "20100110" y descargue el archivo **WinAVR-20100110-install.exe**
- Ejecute el archivo ejecutable y siga las instrucciones.

Paso 4

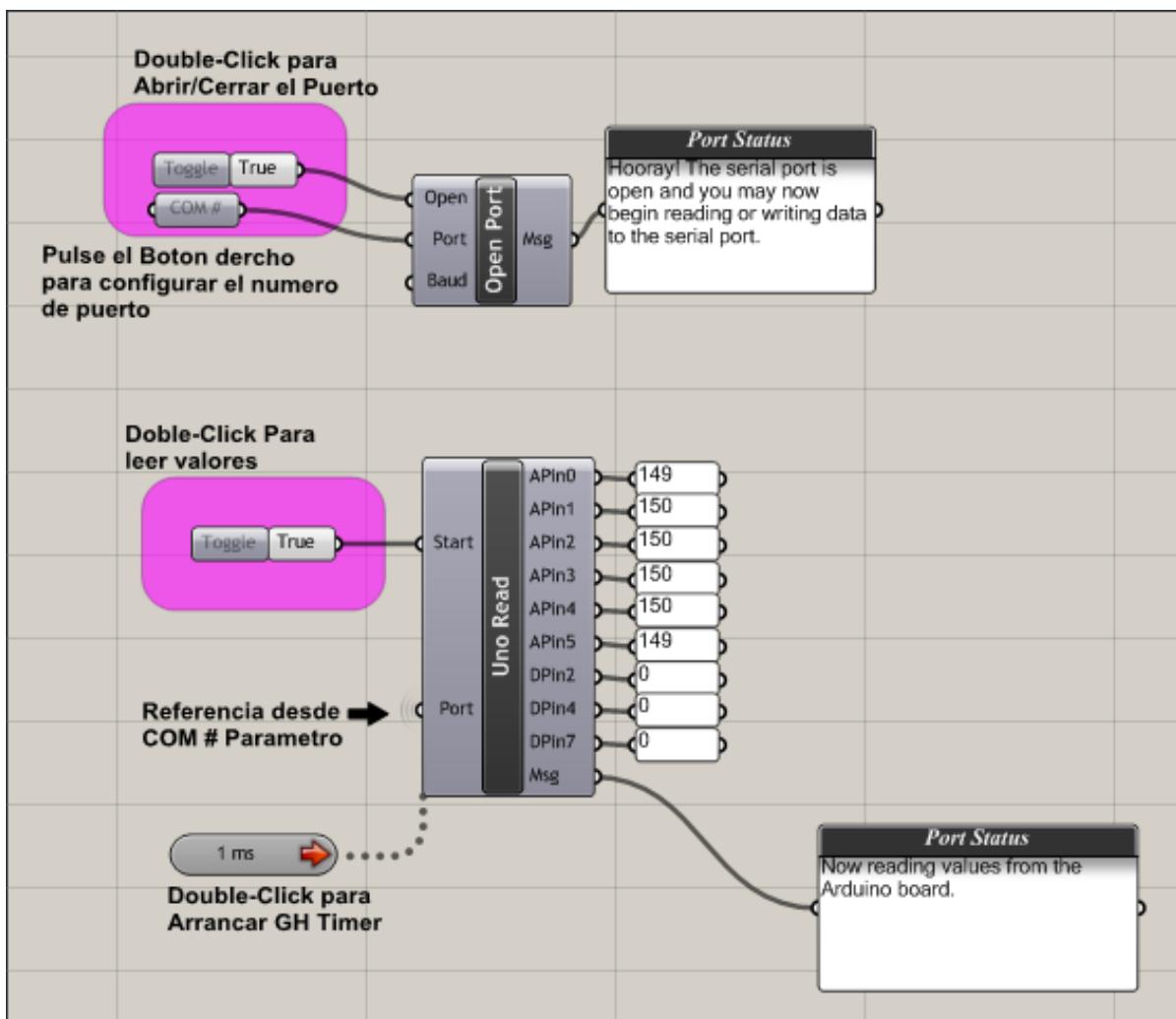
(Trabajar con Firefly de manera ordinaria - cada vez que sea necesario)

Siempre que quiera trabajar con Arduino y Rhino deberá ejecutar Firefly. Lanzado Rhino escribir la palabra "Grasshopper" en la línea de comandos para iniciar el Editor de Grasshopper. Abrir uno de los archivos de ejemplo en la carpeta "Ejemplos de Grasshopper" (dentro del archivo zip) y comenzar a explorar la conexión entre los mundos físico y virtual!

Varios ejemplos



Ejemplo en el que se gobierna la salida 13 de Arduino



Ejemplo en donde se leen los datos de la tarjeta Arduino

1.5 MyOpenLab

Una herramienta para la modelización y simulación orientada a la educación.

Descripción.

MyOpenlab es un entorno orientado a la simulación y modelado de sistemas físicos, electrónicos y de control con un amplio campo de aplicaciones.

La aplicación esta desarrollada en el lenguaje JAVA y por ello resulta portable a distintas plataformas. En el campo del modelado y simulación es muy interesante contar con una herramienta flexible que a partir de una amplia biblioteca de bloques funcionales permita realizar modelos a base de conectar bloques funcionales.

MyOpenLab es capaz de conectarse al mundo físico mediante una interface de amplia difusión en el mercado K8055 de Valleman y también a la tarjeta Arduino

La presentación de los resultados y/o el control de las simulaciones se hace mediante un potente conjunto de bloques de función de visualización y/o interacción capaz de manejar todo tipo de datos (analógicos, digitales, matrices, vectores, imágenes, sonidos, etc.)

Mediante MyOpenLab es posible diseñar instrumentos virtuales (VI) a través de los cuales se puede realizar una aproximación a los sistemas de medida y control de una manera mas realista.

La realización de una simulación se hace mediante dos pantallas o áreas de trabajo: Panel Circuito y Panel Visualización. En el primero se diseña el algoritmo de simulación mediante "bloques" o "elementos de función" y el segundo se muestran los datos o se generan los estímulos cuando se esta en el modo de 2simulación"

El programa puede funcionar en plataformas Linux y sus requerimientos de sistema son muy poco restrictivos, lo cual lo hace ideal para usar en casi cualquier equipo. bastará que se instale el runtime de JAVA JRE o JDK.

Esta herramienta está recomendada para estudiantes de prácticamente todos los niveles: ESO, Bachillerato, Formación Profesional y Primeros Cursos de las Carreras Técnicas Universitarias.

Para conseguir el programa os podéis dirigir a : <http://es.myopenlab.de>

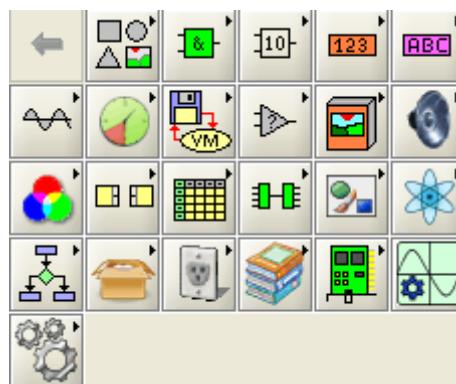
[**Guía de usuario Versión 3.010**](#)
[**Objetos CANVAS.pdf**](#)
[**Tratamiento de Datos.pdf**](#)

[Diagramas de Flujo V2.4.9.9.pdf](#)
[Simulación de Modelos Matemáticos y Temporales.pdf](#)
[Manejo de Datos en Matrices y Tablas.pdf](#)
[Robotica MyOpenLab.pdf](#)

CARACTERÍSTICAS

- Facilidad de uso
- Amplia biblioteca de funciones tanto para manejo de señales analógicas como digitales.
- Posee una potente biblioteca de objetos gráficos tipo "canvas" mediante la que se puede dotar de movimiento cualquier objeto o imagen asociándola a variables de los modelos a simular.
- Tratamiento de los tipos de datos y operaciones con estos.
- Realización de las aplicaciones mediante el uso de bloques de función con la posibilidad de encapsularlos en "macros".
- Facilidad para crear pantallas de visualización que recojan el estado de las variables y eventos de las simulaciones.
- Posibilidad de ampliación de su librería de componentes, editándolos en código JAVA
- Posibilidad de creación de "submodelos de panel" y "submodelos de circuito" encapsulados.
- Algunas librerías que incorpora MyOpenlab:

Librería de elementos de Visualización y Control (Panel Frontal)



*Elementos de Decoración
Elementos de visualización numérica
Elementos de activación digital
Elementos de Entrada y salida de cadenas de caracteres*

*Elementos de entrada y salida tipo vectores y matrices de datos
Elementos de visualización grafica en ejes coordenados I
Elementos de visualización grafica en ejes coordenados II*

*Librería de Extras
Elementos de Automatización*

*Elementos de librería de usuario
Robot 2D*

Librerías de Elementos Funcionales (Panel Lógico)



Elementos de Decoración

Operadores Digitales

Operadores Numéricos

Tratamiento de Caracteres

Elementos Analógicos

Utilidades

Ficheros de Entrada/Salida

Comparators

Tratamiento de Imágenes

Tratamiento de Sonidos

Color

Pines de E/S

Vectores y matrices

Agrupación de Elementos

Objetos Gráficos "canvas"

Librería de Física

Librería de Diagramas de Flujo

Librería de Extras

*Librería de Conexiones entre
aplicaciones*

Librería definida por el Usuario

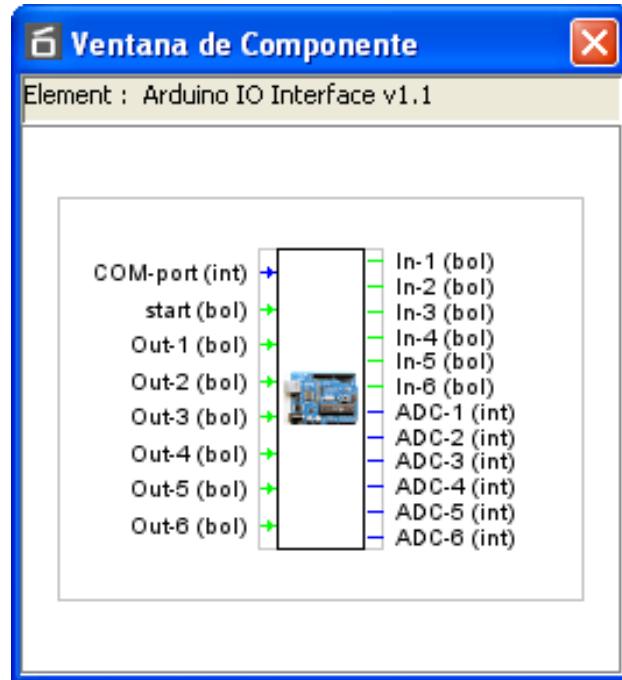
Automation+ibrería de

Automatización Interfaces

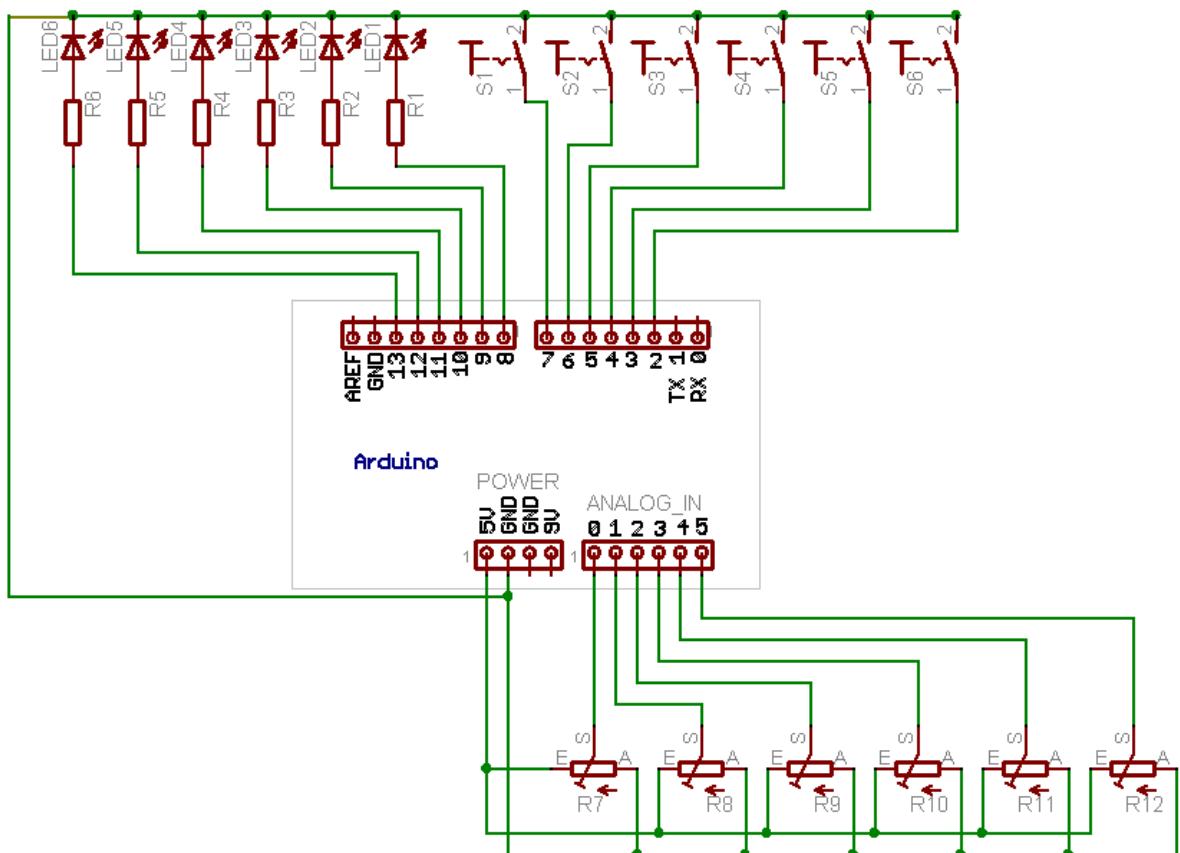
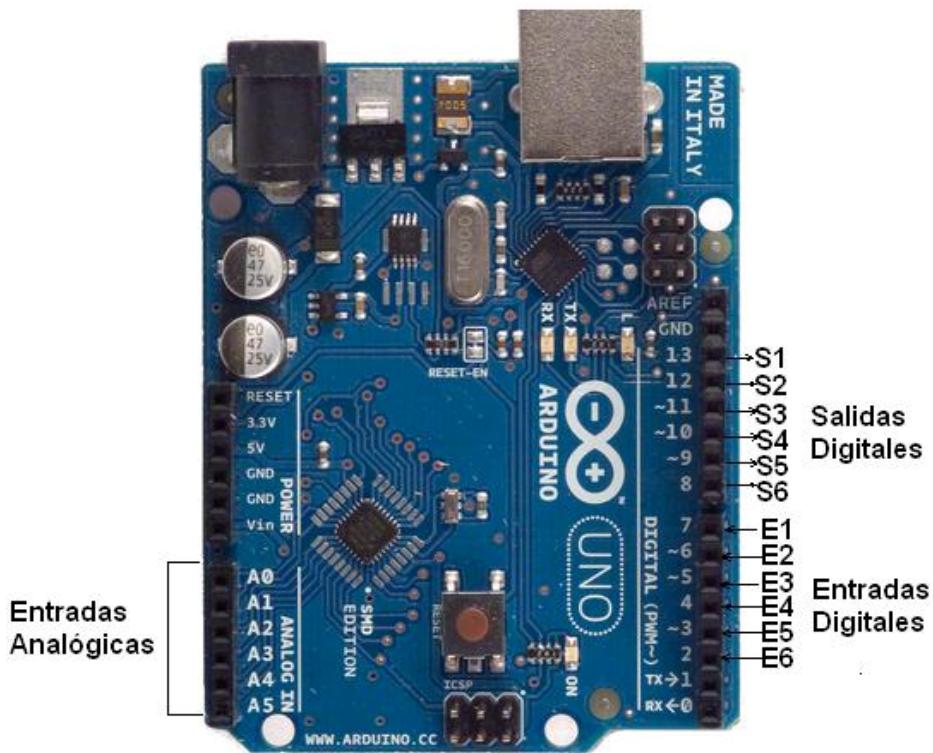
Forma de conexionado con Arduino.

Para conseguir la comunicación entre Arduino y Myopenlab se seguirán los siguientes pasos:

1. Cargar el Firmware en la tarjeta Arduino con la ayuda del IDE Arduino. El fichero se encuentra en la carpeta D:\distribution3032\MyOpenLab_and_Arduino V\arduino y el fichero se llama: arduino.pde
2. Realizar el diseño dentro del entorno de Myopenlab utilizando la librería “Arduino IO Interface 1.1” que es la que se encargara de la comunicación con la tarjeta Arduino.
3. Consignar los elementos de entrada y salida de acuerdo con la configuración que el firmware coloca en la tarjeta y que se muestra en la figura.
4. No olvidar que hay que dejar previsto un elemento de entrada de valor numerico para consignar el numero del puerto por el que se realizara la comunicación y una entrada digital (tipo pulsador) para que se active el diálogo entre Arduino y Myopenlab
5. En el panel de visualización se podrán colocar elementos graficos para mostrar los valores de entrada y salida que le darán a la aplicación un aspecto visual muy interesante.
6. Finalmente una vez realozado el diseño se activara el modo Run de Myopenlab y veremos la aplicación funcionar.

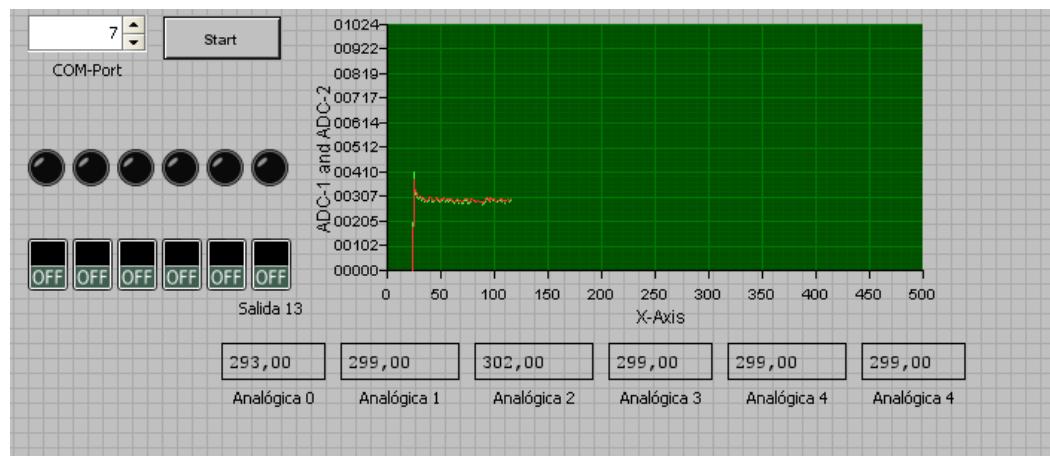
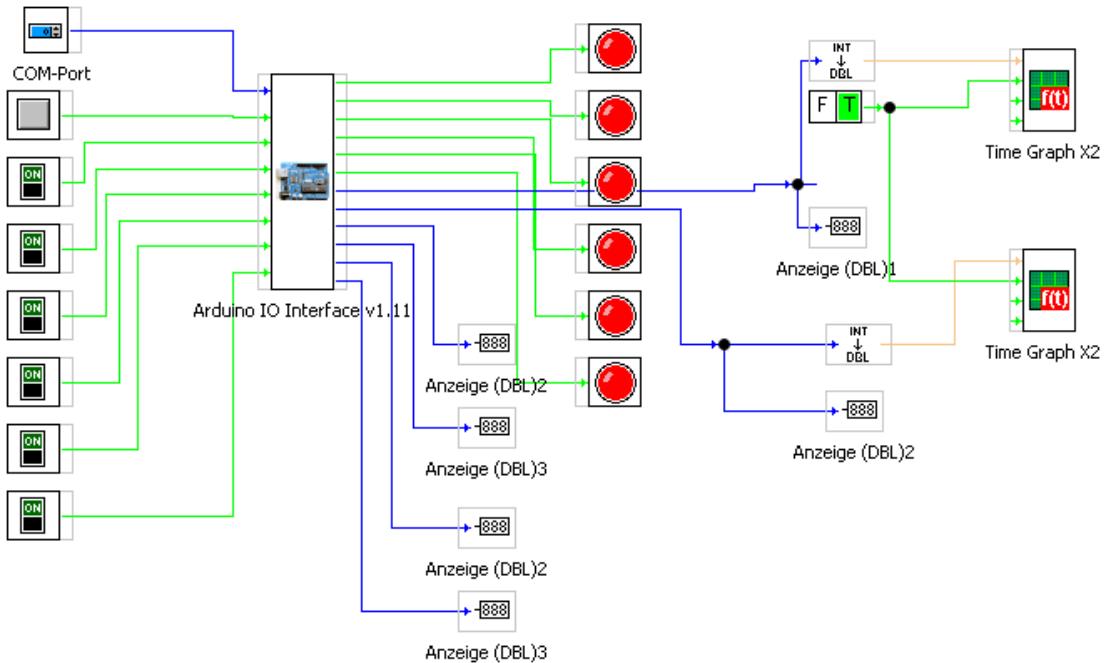


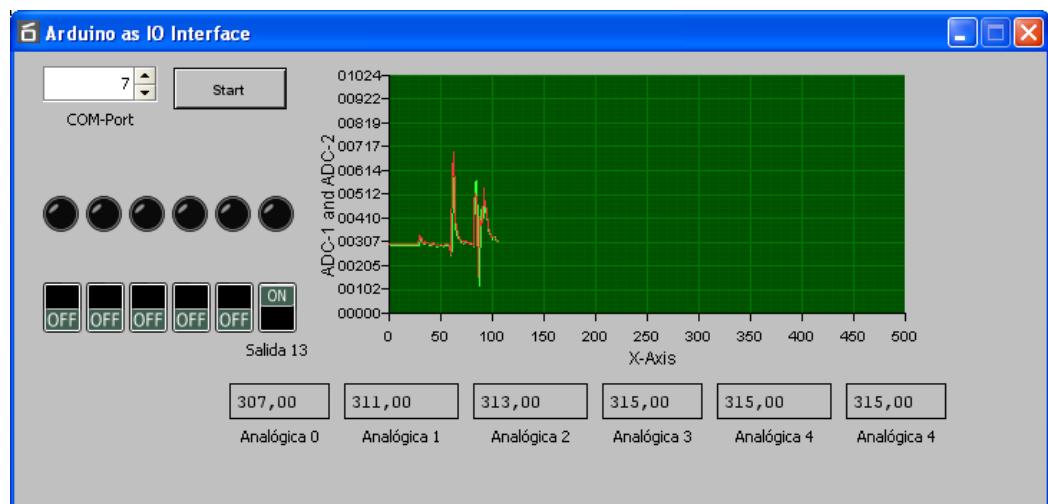
Componente de librería para conexión con Arduino



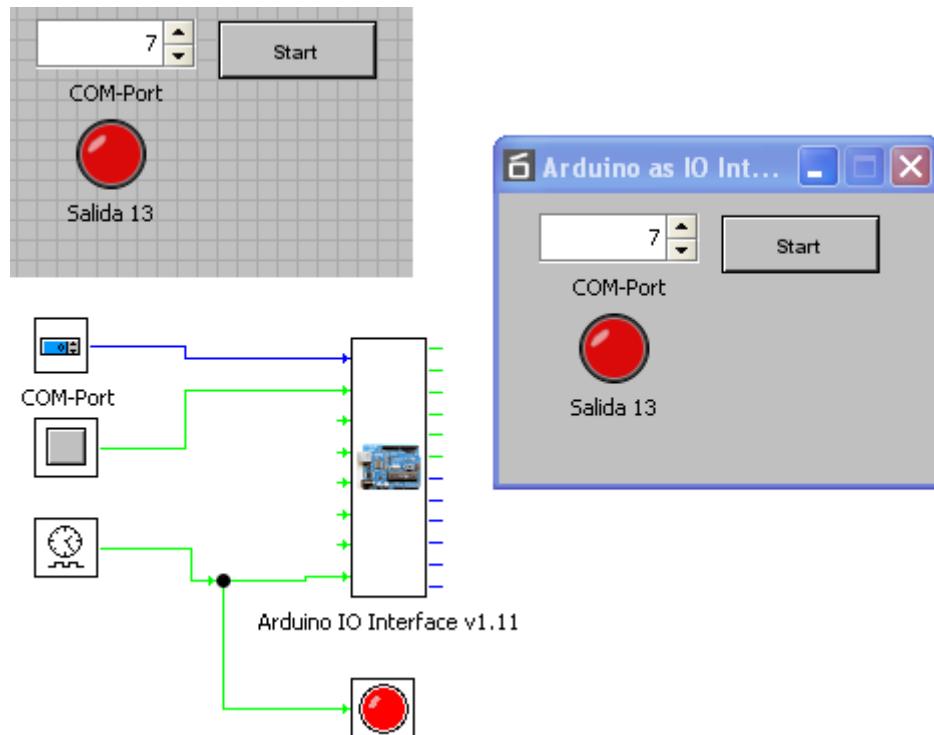
Designación de E/S para la conexión de Arduino con Myopenlab

Ejemplo de conexionado operación manejando todas las E/S de la tarjeta Arduino





Ejemplo de salida intermitente en el PIN 13 de Arduino salida 0



2 Entornos de programación autónomos

2.1. Minibloq

Descripción General

Minibloq es un entorno de programación gráfica para Arduino™, Multiplo, dispositivos físicos informáticos y robots. Una de sus principales objetivos es llevar la computación más físico y plataformas robóticas a la escuela primaria, los niños y principiantes. Si desea ver una breve introducción, se puede echar un vistazo al video utilizado para la campaña de pedal de arranque.

Características

Minibloq está en desarrollo. Estas son las características implementadas en la última versión de trabajo:

Fácil: Sólo unos pocos clics y su primer programa se está ejecutando.

En tiempo real generador de código: Se crea el código, mientras que usted está agregando bloques o modificar los valores de parámetros, que muestra el código en una ventana de sintaxis de colores. De esta manera, *Minibloq facilita la transición a la programación basada en texto*.

Tiempo real, la comprobación de errores.

Drag & drop básico con giro automático.

Interfaz avanzada: zoom, cortar y pegar, ventanas acoplables, y el teclado de navegación son sólo algunas de las características de la interfaz gráfica de usuario de Minibloq. Y hay más ...

Terminal incorporado: Hay una terminal integrado que le permite enviar y recibir datos a la junta a través de puertos serie / USB.

Todo-en-uno-listo-para-uso-solución: Se trata de "baterías incluidas software". El paquete incluye todo lo posible para empezar a trabajar.

Portable: No requiere la instalación (a excepción de los controladores necesarios para tarjetas específicas, como Arduino™). Se puede ejecutar desde un pen drive también. Ah, y se ejecuta por completo fuera de línea, todos ellos en su propio ordenador. Más información: Usted puede tener copias paralelas de Minibloq, incluso con la configuración de diferentes que se ejecutan en el mismo equipo.

Rápido: Es una aplicación nativa, compilado con C ++ (GCC), con wxWidgets. Por esta razón, Minibloq es adecuado para ordenadores de gama baja y netbooks. Y, también incluye los núcleos precompilados, se construye y se descargan los programas muy rápido. Pruébelo!

Modular y ampliable: El usuario puede crear nuevos bloques es propia.

Aquí está un pequeño video que muestra algunas de estas características:

Próximamente características

Libre y con las fuentes: El programa estará disponible de forma gratuita, en su versión completa. Sin cargos para funciones avanzadas o similares. Sólo tienes que descargar y empezar a usarlo. Además, el código fuente completo también estará disponible. La licencia? Algo que se llama RMPL (RobotGroup-Multiplo-pacifista-License). Esta licencia es básicamente una licencia MIT, pero con una restricción que prohíbe los proyectos militares. Más sobre esto más adelante. Por ahora, se puede ver la versión en español de esta licencia aquí

Fácil integración con hardware nuevo: Añadido soporte para nuevas plataformas y placas será sencillo. Esto puede no ser una característica para los principiantes, pero no será tan difícil de todos modos. Compiladores y lenguajes diferentes, incluso se podría añadir.

Internacionalización: La primera versión estará disponible en Inglés y Español, pero el usuario y la comunidad pueden contribuir con nuevas traducciones, ya que esto sólo implica editar un archivo de texto.

Como utilizarlo:

Descargar la herramienta en <http://blog.minibloq.org/>

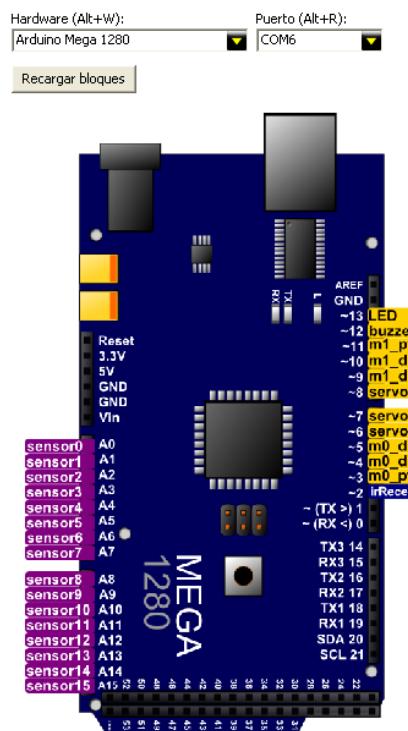
Una vez instalado el software bastará con iniciar el programa y aparecerá el entorno:

Proceso a seguir:

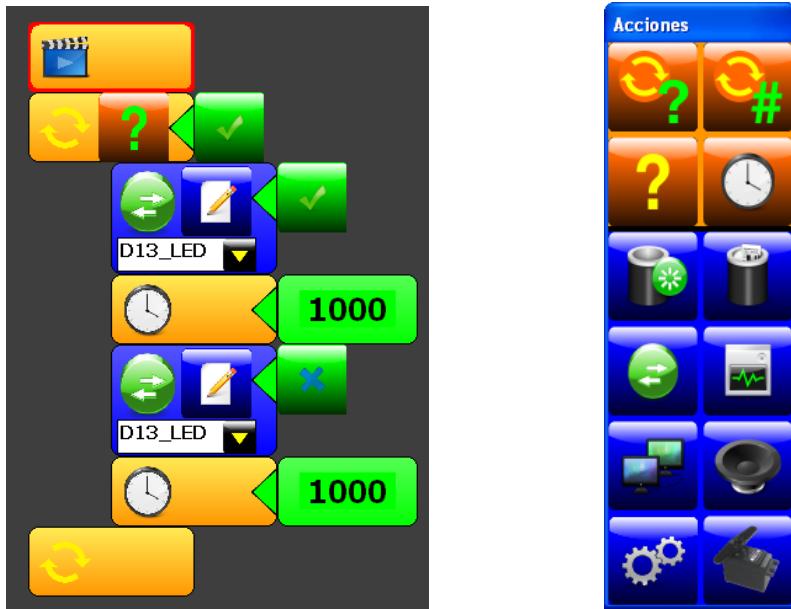
1. Seleccionamos la tarjeta de entre el grupo de tarjetas  con las que se comunica Minibloq.



2. Se conecta la tarjeta al puerto USB y dejando unos segundos para que la detecte el software se selecciona el puerto en el que se ha conectado en el lugar correspondiente del entorno.



3. Ya estamos en disposición de empezar a programar arrastrando bloques y colocándolos en el área correspondiente. Cada bloque deberá programarse con sus parámetros correspondientes.



4. Si lo deseamos podemos abrir la ventana de código y ver cómo se va escribiendo el código a medida que colocamos bloques en nuestra aplicación.

```
#include "stdlib.h"
#include "IRremote.h"
#include "pitches.h"
#include "Minibloq.h"

void setup()
{
    initBoard();
    while(true)
    {
        digitalWrite(D13_LED, true);
        delay(1000);
        digitalWrite(D13_LED, false);
        delay(1000);
    }
}

void loop()
{}
```

5. Una vez escrito el programa se “Ejecuta” enviándose a la tarjeta correspondiente.

Cada componente tiene unos parámetros que son configurables

Por ejemplo en la figura se muestran el menú de configuración de un bloque de temporización



Los bloques disponibles para programar Arduino son:



2.2. Ardublock

([ArduBlock](#))

Descripción general

Esta herramienta está basada en la tecnología de programación mediante bloques funcionales tan extendida en la actualidad. Realmente se distribuye como un applet de java que se añade a las herramientas del IDE Arduino.

Realmente Ardublock es una utilidad grafica cuya misión es generar código compatible con el entorno IDE Arduino. Sus ventajas son:

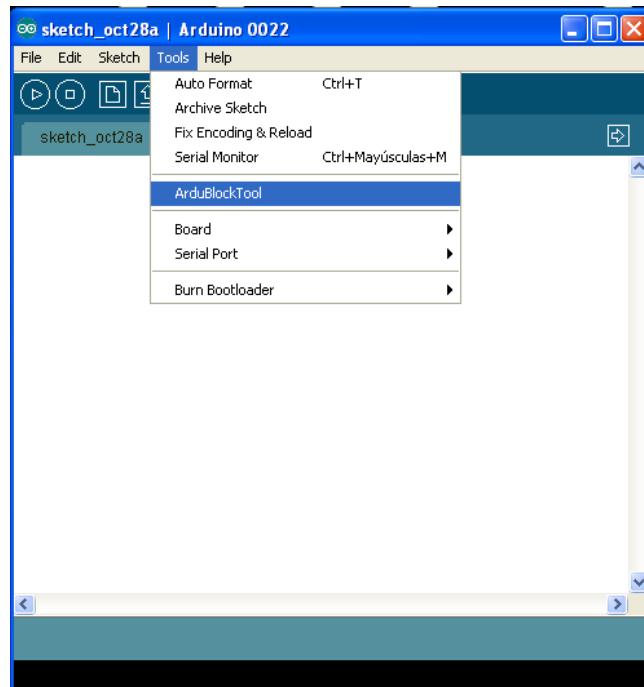
- Es una herramienta gratuita
- Facilita la creación de sketch para Arduino.
- Genera código directamente
- Ofrece una colección de bloques funcionales muy básicos que facilitan la comprensión de la programación.
- Esta muy indicado para aplicarlo en niveles educativos básicos en donde el usuario no necesita tener conocimientos de programación.
- Es una aplicación muy sencilla de instalar
- Es muy sencillo de utilizar

Procedimiento e instalación

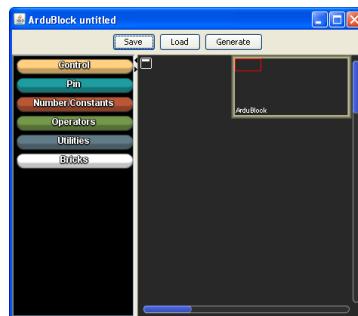
1. Descargar *ardublock-all.jar* [ArduBlock](#)
2. En la carpeta en donde este instalado el IDE Arduino debemos incluir el fichero *ardublock-all.jar* “*....arduino-022/tools/ArduBlockTool/tool/ardublock-all.jar*”

Modo de trabajo

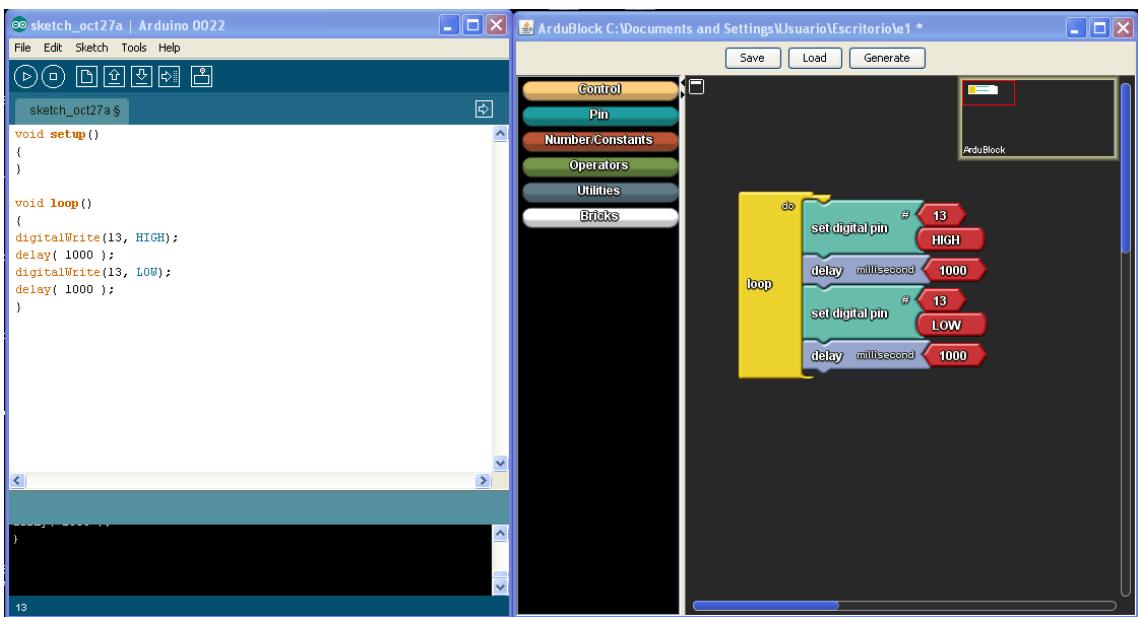
Una vez instalado el fichero ardublock-all.jar en la correspondiente carpeta se arranca el IDE de Arduino y para invocar Ardublock basta que seleccionemos *Tools->Ardublock Tool*



Después de seleccionar la herramienta aparece la pantalla de programación gráfica de la figura.

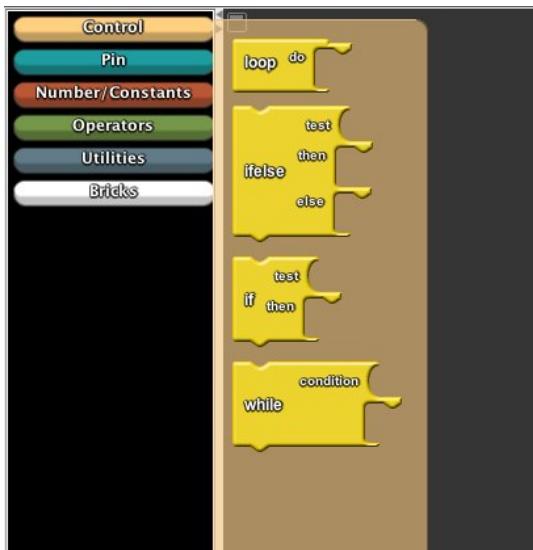


Se realiza la programación gráfica con la ayuda del entorno seleccionando los bloques correspondientes y una vez termina se activa sobre el botón “Generate” y aparece en el IDE de Arduino el código escrito de la aplicación lista para ser transferido a la tarjeta Arduino.



Librerías de Ardublock.

A continuación se muestran las librerías de bloques con las que se cuenta en el entorno.



Control



Pin



Numeros/Constanets

Operadores



Utilidades

Bloques de Hardware

2.3. Amici

(<http://dimeb.informatik.uni-bremen.de/eduwear/about/>)

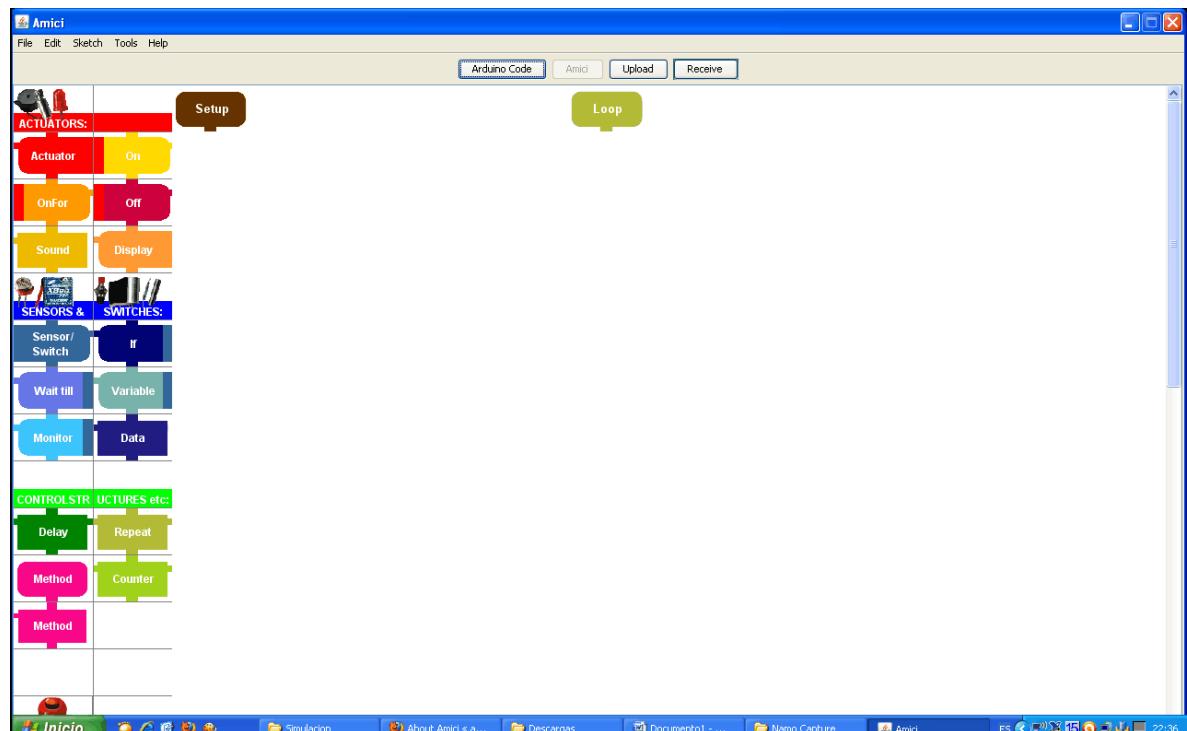
Introducción

Amici es un Lenguaje de Programación Visual para Arduino que permite programar a los novatos la placa Arduino sin tener que dominar el lenguaje textual de programación de Arduino. Es de código abierto (el código fuente bajo petición), publicado bajo la licencia GPL.

Amici ha sido desarrollado por investigadores de [DIMEB](#) (Digitale los medios de comunicación en la Educación) de la Universidad de Bremen, como parte de la EduWear proyecto europeo. El software ha utilizado en más de 25 talleres por niños y jóvenes.

El entorno se suministra unido a una versión completa del IDE de programación de Arduino, lo cual permite sin tener que cargar ningún firmware ni aplicación alguna, realizar el programa en un entorno gráfico y directamente se genera el código a la vez que se abre la aplicación IDE Arduino y desde ella se descarga la aplicación en la tarjeta Arduino.

En la figura vemos un aspecto del entorno Amici.

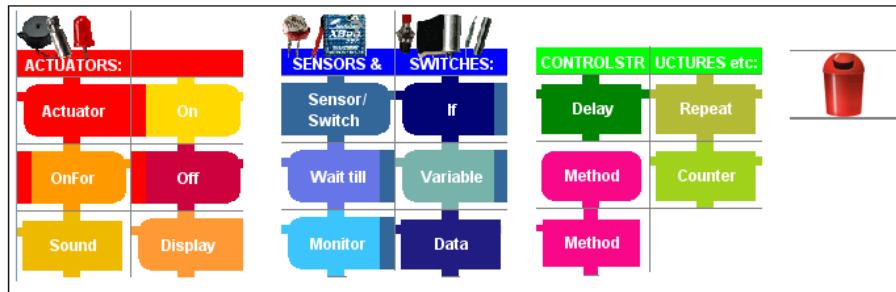


La aplicación permite arrastrar los bloques uno debajo del otro en el área de trabajo y a la vez dando valores a los parámetros de cada uno de ellos.

La forma de trabajar es muy sencilla, una vez realizado el diagrama de bloques se pulsa en botón “Arduino code” y se genera el código compatible con el lenguaje de programación de la tarjeta Arduino y desde esa pantalla , una vez asignado el puerto de conexión de la tarjeta y el tipo de tarjeta se descarga la aplicación.

Lo que no permite la herramienta es realizar la operación inversa, es decir, pasar de código a organigrama. Si pulsamos el botón “Amici” volveremos al entorno grafico Amici y podremos realizar en el las modificaciones oportunas.

Bloques disponibles en Amici:

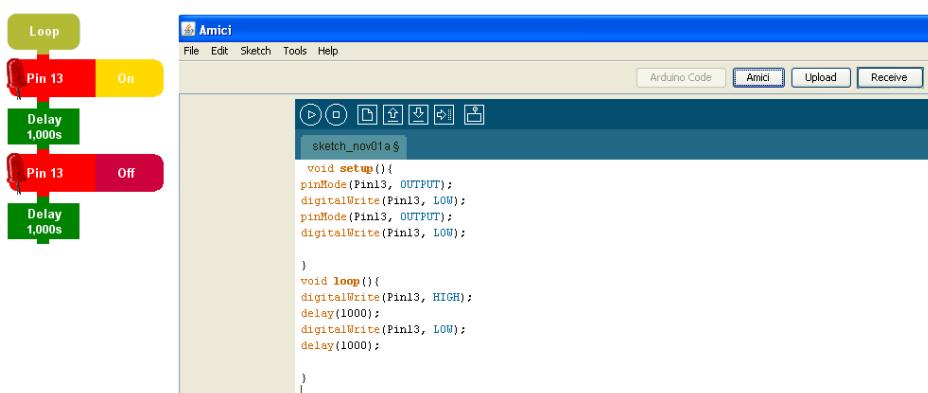


Existen los siguientes tipos de bloques agrupados.

- Actuadores
- Sensores e interruptores
- Estructuras de programación

Ejemplo realizado con el entorno.

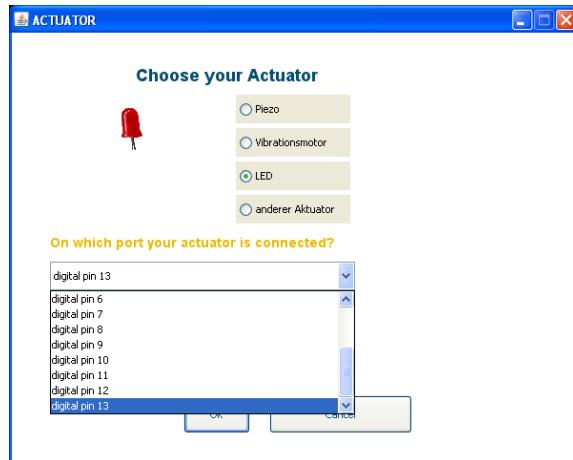
En la figura vemos un sencillo ejemplo realizado en el modo gráfico con el entorno Amici y a su lado el código generado.



Se ha programado la salida 13 que se activará y desactivará cada 1000 ms.

Designación de parámetros en cada bloque.

La mayoría de los bloques al pulsar dos veces sobre ellos despliegan una ventana para asignarles parámetros. En la figura vemos la ventana de parámetros del bloque “Actuador”. Este mecanismo es muy interesante de utilizar.



Opinión de la herramienta.

Este entorno es muy interesante como hemos dicho al principio para la iniciación en la programación, pero no es válido para niveles de aplicación medios y superiores de la plataforma Open Hardware Arduino, dado que se encuentra muy limitado en la potencialidad de sus bloques de función, aunque es importante decir que bien usados estos bloques permiten muchas y variadas formas de programación.

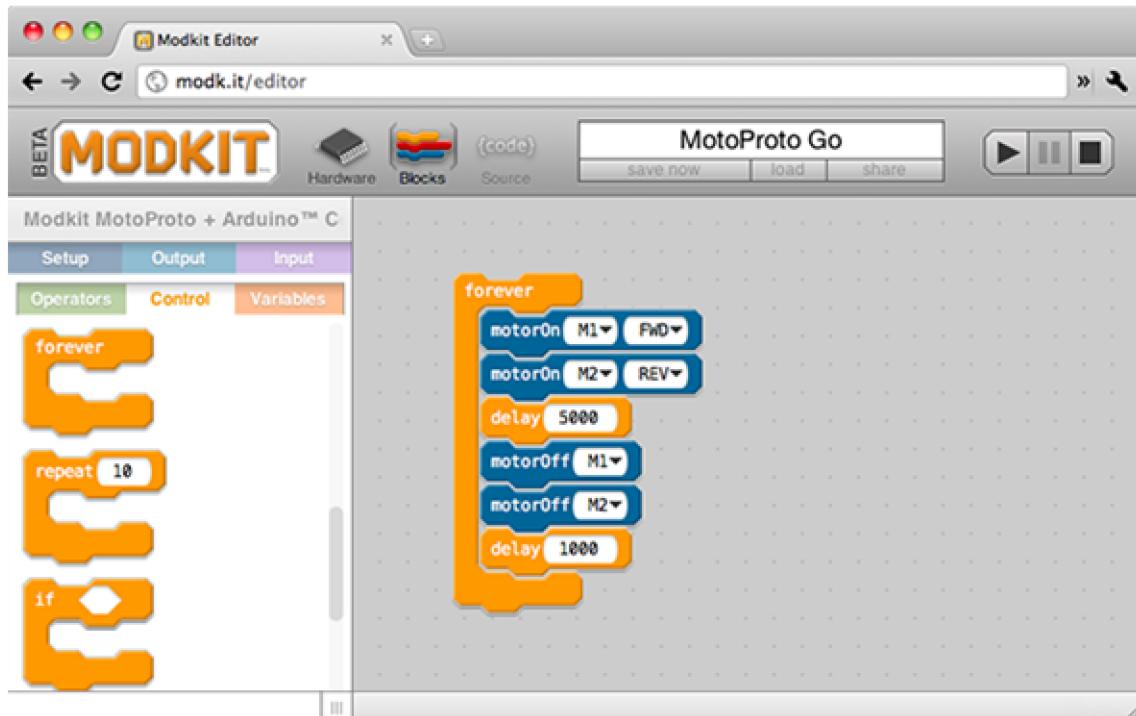
Al estar disponible el código bajo petición la herramienta es susceptible de ser mejorada y ampliada.

2.4.MODKIT

ModKit: Entorno de programación gráfica para arduino!

Introducción

Modkit es una herramienta completa para el diseño de aplicaciones con Arduino y otras tarjetas haciendo uso de un lenguaje de programación gráfica a base de bloques basados en el modelo creado por el MIT en el entorno Squeak.

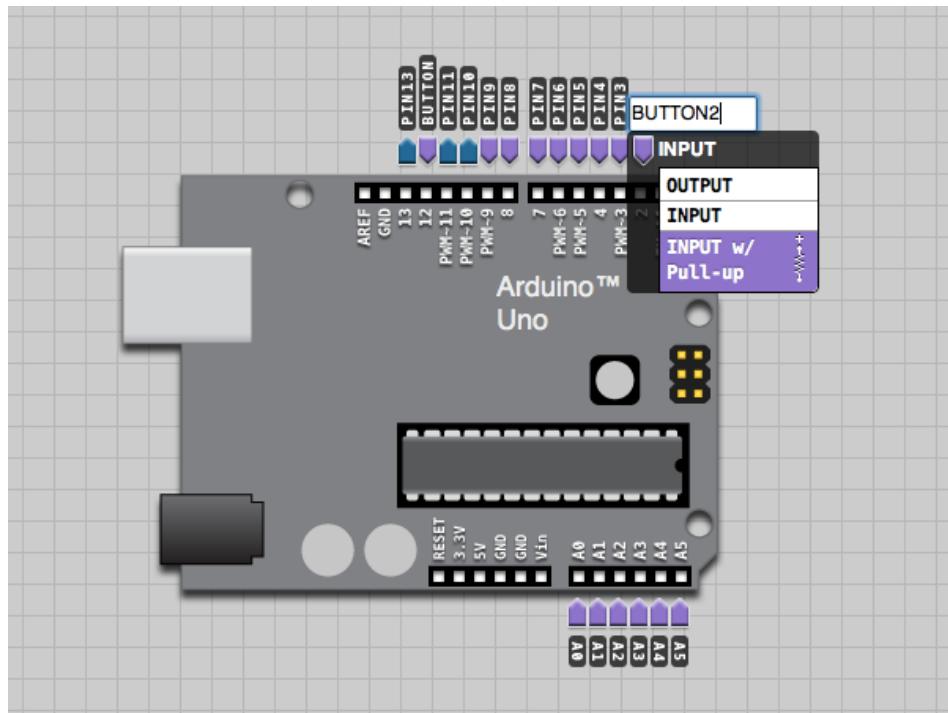


Tiene las siguientes características a resaltar:

1. Drag & Drop
Enlazado de bloques gráficos de código para crear programas, evitando errores de sintaxis.
2. Basada en la Web
Modkit se ejecuta en los navegadores modernos y permite trabajar con proyectos en cualquier lugar.
3. Compatible Arduino
Modkit trabaja con el popular ™ Arduino
4. Usarlo de forma gratuita
Queremos que nuestras herramientas sean accesibles a todo el mundo.
Empezar a trabajar con nuestra versión gratuita hoy.

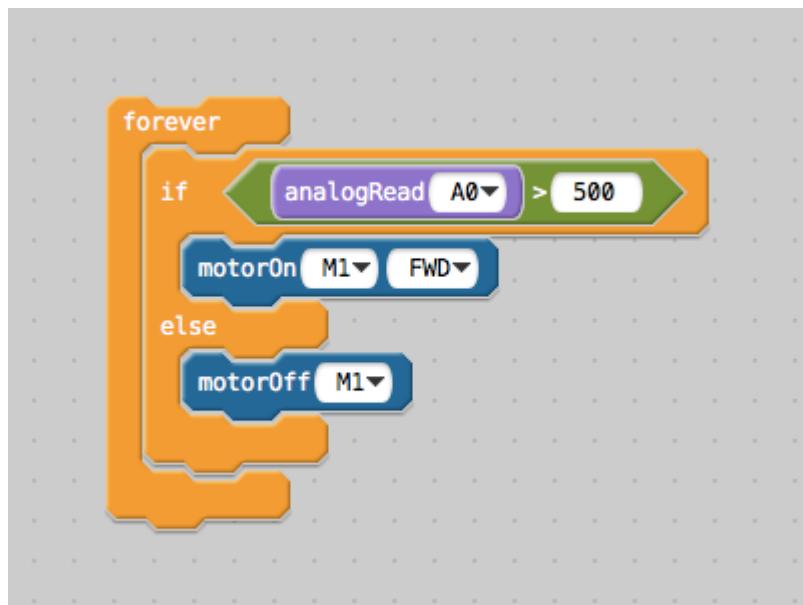
Características de la versión libre

Permite seleccionar el hardware que vamos a utilizar



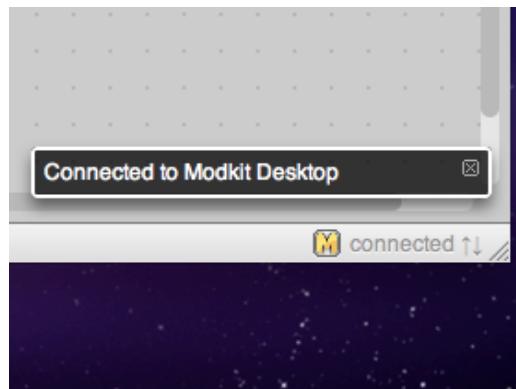
Los proyectos deben empezar configurando el hardware y los programas.

Pantalla de edición del programa con bloques gráficos



Después de configurar el hardware de gráficos, Modkit hace que la programación sea tan simple como arrastrar y soltar los bloques de código gráfico para crear programas interactivos.

Widget de escritorio



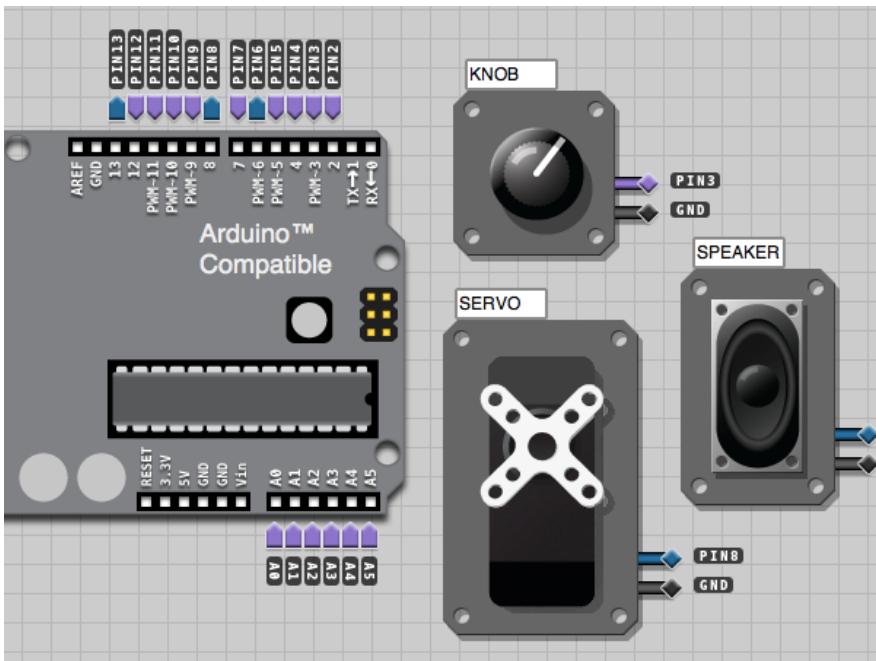
Editor revolucionario Modkit está basada en tecnologías web estándar y se ejecuta en tu navegador. Sólo tienes que descargar e instalar nuestro widget de escritorio libre para programar sus tablas.

Ver código

A screenshot of a code editor interface titled "MotoProto Go". On the left, there is a button labeled "{code}" with a "Source" link below it. The main area contains the following Arduino-like pseudocode:

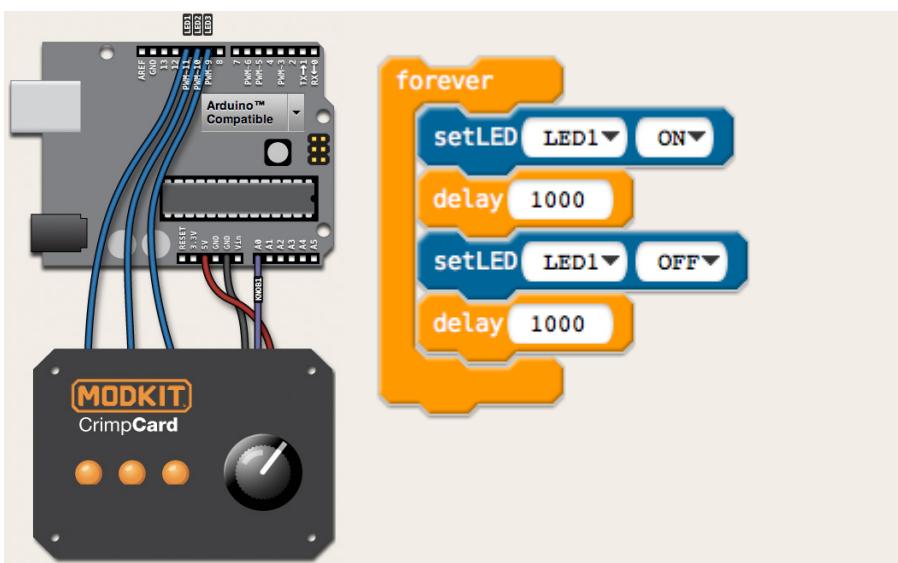
```
forever{
    if((analogRead(A0)) < (900)){
        motorOn(M1,FWD);
    }
    else{
        motorOff(M1);
    }
}
```

Los principiantes pueden comenzar con los bloques y pasar a la programación de texto tradicional. Los programadores experimentados pueden pasar por alto los bloques sin dejar de tomar ventaja de medio ambiente rico en características de Modkit. Obtener acceso temprano a la vista de código como un miembro del Club Alpha.



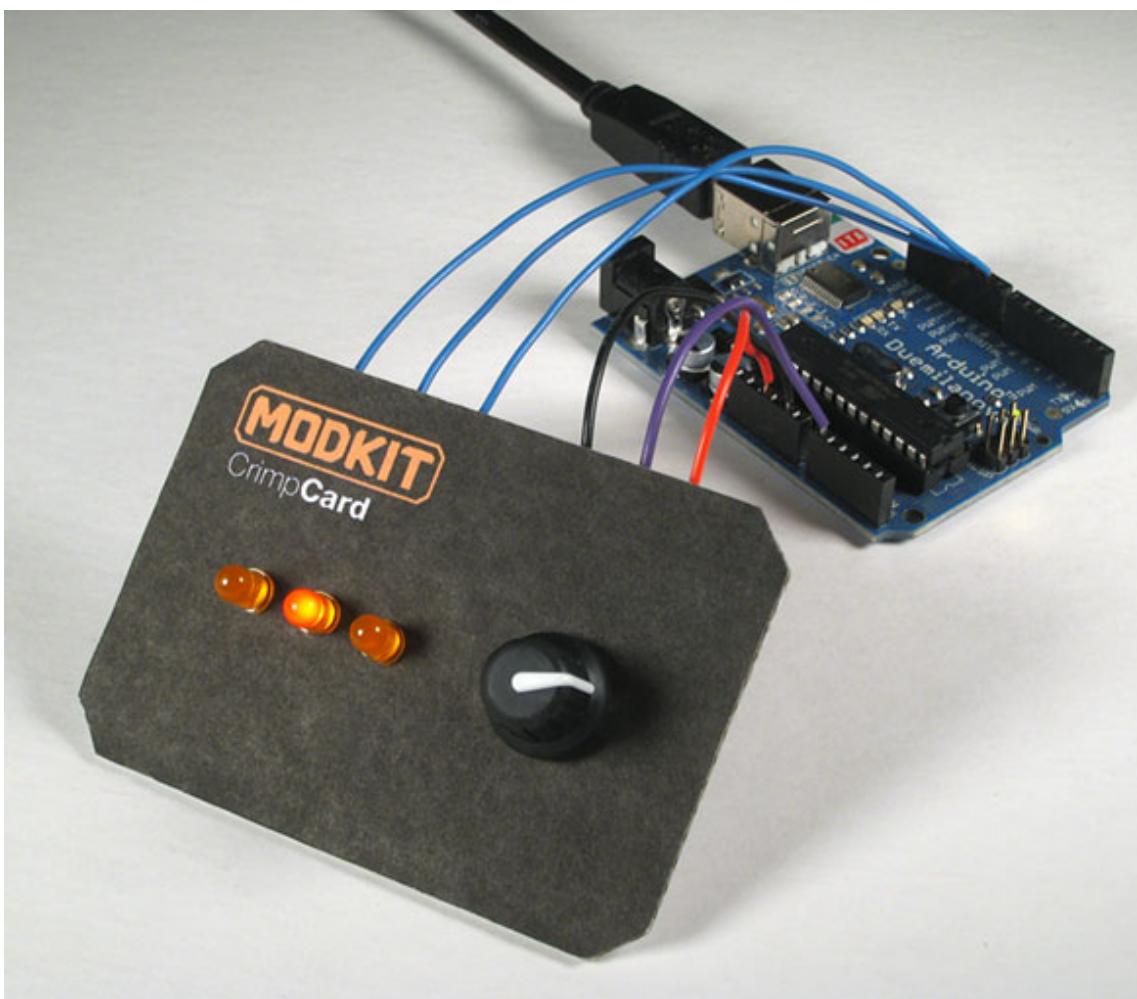
Ir más allá de configuración de pines de arrastrar y soltar los componentes de hardware, lo que le permite usar los bloques de programación adicionales para controlar los componentes.

Una nueva herramienta de programación gráfica de Arduino que está a punto de terminarse y que facilitará mucho la programación de la tarjeta.



Hardware | Blocks | **Source** | Photos & Videos

```
forever{
    setLED(LED1,ON);
    delay(1000);
    setLED(LED1,OFF);
    delay(1000);
}
```



2.5 Virtual BreadBoard

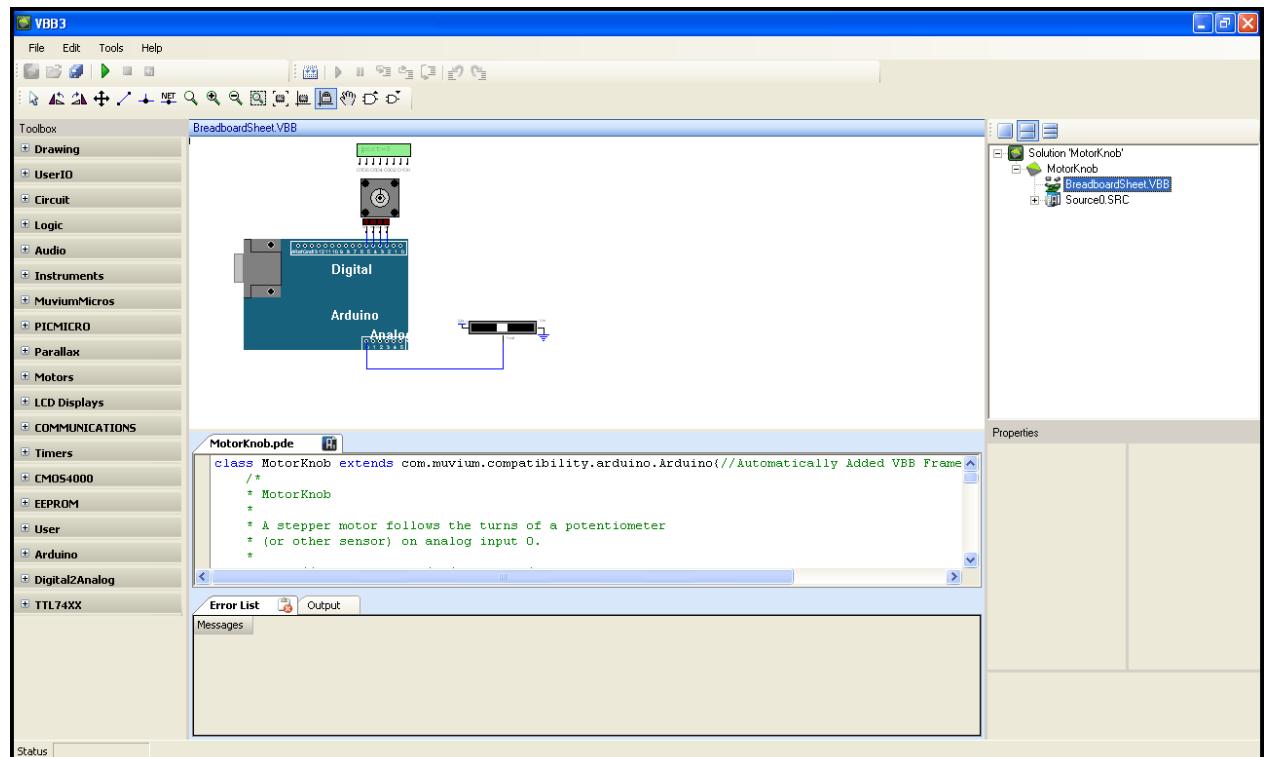


<http://www.virtualbreadboard.com/>

Introducción.

VirtualBreadboard es un entorno de simulación y desarrollo de aplicaciones integradas que utilizan los microcontroladores. Es fácil de usar y puede sustituir a un protoboard para experimentar con nuevos diseños.

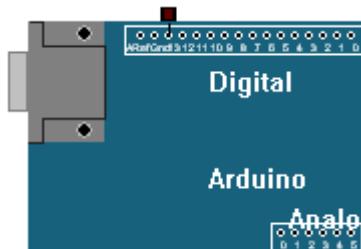
VBB simula muchos de los PIC16 y dispositivos PIC18 y el microcontrolador Arduino. Además de una amplia variedad de componentes de simulación tales como LCD, Servos, la lógica y otros dispositivos IO que se pueden utilizar para modelar y simular circuitos de alto nivel.



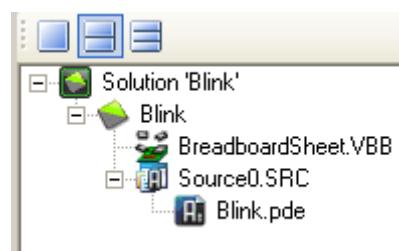
Creación de un proyecto para Arduino

Digamos que tienes una fuente de Arduino archivo de código que desea simular. Hay un montón de ejemplo Arduino integrado en VBB, pero este tutorial te muestra cómo hacerlo desde cero?

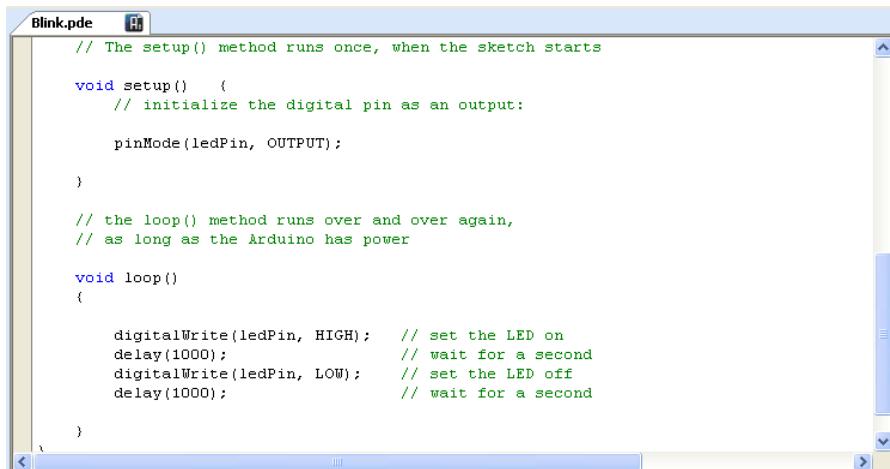
1. Partamos de que tenemos una aplicación de Arduino llamada Blink.pde en el escritorio.
2. Iniciamos VBB y crear un nuevo proyecto.
3. Arrastramos y soltamos una placa Arduino en el área de diseño y luego arrastramos y soltar un LED uniéndolo al pin 13



4. Guardar el proyecto: Téngase en cuenta la importancia para guardar el proyecto antes de agregar proyectos de código. Llamamos al proyecto a grabar MyBlink
5. Copiamos la aplicación Arduino Blink.pde en la carpeta del proyecto MyBlink VBB
6. Añadimos un proyecto de Código Arduino a la carpeta haciendo clic derecho sobre el proyecto y seleccionar de la lista.
7. Agregamos el archivo de código fuente para Arduino Blink.pde al proyecto de código Arduino haciendo clic derecho sobre el proyecto de Código Arduino.



8. Puede seleccionar Blink.pde de la lista porque está en la carpeta del proyecto ya.
9. NOTA: Para ver el proyecto de código que necesita para arrastrar y soltar en un panel de vista.



The screenshot shows the Arduino IDE interface with the file 'Blink.pde' open. The code is as follows:

```
// The setup() method runs once, when the sketch starts
void setup() {
    // initialize the digital pin as an output:
    pinMode(ledPin, OUTPUT);
}

// the loop() method runs over and over again,
// as long as the Arduino has power
void loop()
{
    digitalWrite(ledPin, HIGH);      // set the LED on
    delay(1000);                  // wait for a second
    digitalWrite(ledPin, LOW);      // set the LED off
    delay(1000);                  // wait for a second
}
```

10. Ahora necesitará vincular el proyecto de código fuente para el procesador Arduino para que haga clic en la placa Arduino y luego seleccione la aplicación de la propiedad el nombre de la fuente del proyecto.
11. Ya está listo para ejecutar la simulación para haga clic en Ejecutar y verá el parpadeo del LED
12. Usted puede in-situ editar el código de arduino pero hay que guardarla antes de compilar. Si usted comete errores de sintaxis que se vea en la lista de los errores.

Sistemas operativos compatibles

VBB3 es una aplicación Windows desarrollada con Visual Studio 2005 y debería funcionar bien en una PC moderna corriendo Windows XP y Vista o Windows 7

Linux no es compatible en este momento sin embargo, el objetivo a largo plazo se va a ejecutar en la plataforma VBB Mono.Net que es soportado por Linux. Sin embargo Mono todavía no es compatible con DirectX y otras bibliotecas por lo que sigue siendo un trabajo considerable que queda antes de portar Linux es compatible.

Instrucciones de instalación

VirtualBreadboard es una caja de arena ejecutable autocontenido Xenocode que no requiere instalación. Usted puede descargar a su lugar preferido y haga doble clic para lanzar.

Dependencias.

Las siguientes dependencias deben instalarse por separado.

- .Net 2.0 Redistributable. Disponible de Microsoft [aquí](#).
- J # 2.0 Redistributable. Disponible de Microsoft [aquí](#).



3. Otros entornos

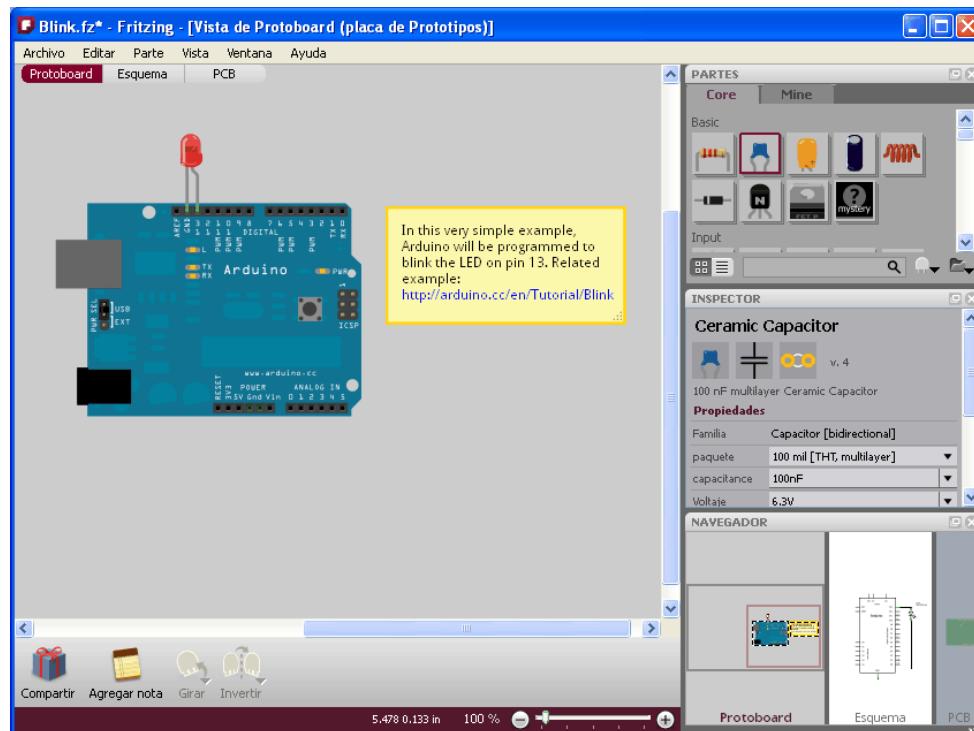
3.1.Fritzing.

<http://fritzing.org/>

Descripción

Fritzing es una iniciativa de código abierto para apoyar a los diseñadores, artistas, investigadores y aficionados para facilitar la elaboración de los prototipos físicos al producto real. La herramienta permite documentar las aplicaciones desde el prototipo hasta el diseño de la tarjeta PCB que se podría fabricar. La herramienta Fritzin está colocada en un sitio web que facilita a los usuarios compartir diseños de acuerdo a la filosofía del “Open hardware” y poder discutir e intercambiar las experiencias así como poder reducir los costes de fabricación.

Fritzing es esencialmente un software de automatización de diseño electrónico con un nivel de exigencia en el manejo muy bajo, lo que permite cubrir necesidades y expectativas de los diseñadores y artistas. Se utiliza la idea de “protoboard” (placa de montajes sin soldaduras) que es ideal en el laboratorio de prototipado.. El software luego se encarga de sacar el alambrado (esquema eléctrico) y después el esquema de pistas de la tarjeta de circuito impreso a partir de ahí es posible crear diseños de PCB para convertirlo en un producto susceptible de ser fabricado.



Forma de uso de Fritzing

Las herramientas de creación de prototipos físicos han recorrido un largo camino y ya permiten que los ingenieros puedan convertir rápidamente sus ideas en prototipos funcionales interactivas.

Se trata de una herramienta para diseñar a nivel de protoboard o circuito impreso una aplicación con Arduino. Tiene un gran interés didáctico porque permite la identificación de los componentes electrónicos que forman parte del prototipo, su

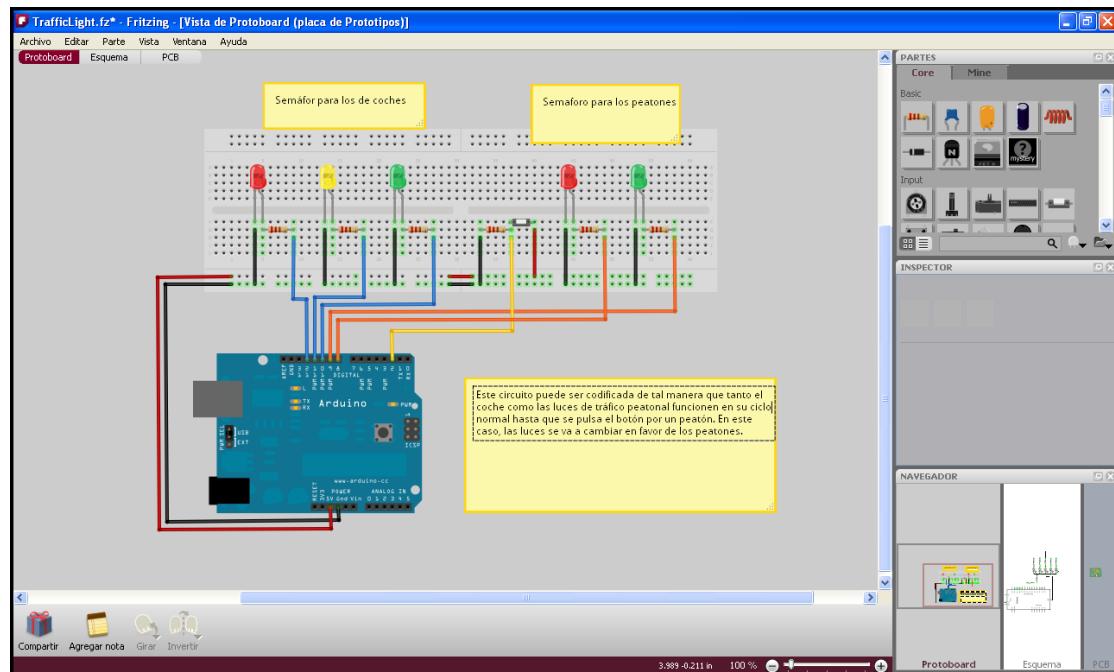
Distribución y su conexionado. En la página existen números proyectos ya realizados. La herramienta se puede descargar aquí: <http://fritzing.org/download/>

Elaboración de un diseño.

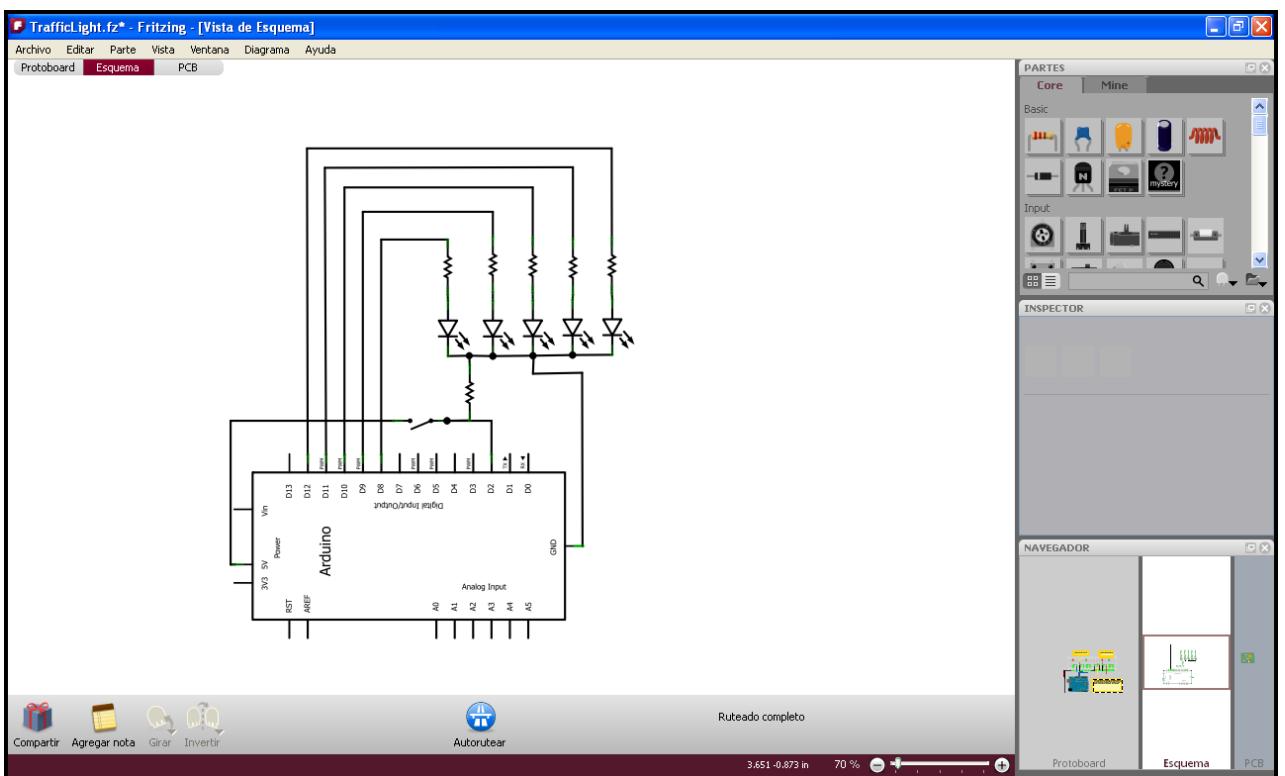
El entorno posee una librería de componentes electrónicos que incluye naturalmente la tarjeta Arduino. La idea es ir colocando cada elemento unido mediante un sistema de cableado convencional los componentes tal y como se supone que se deberían conectar en la realidad haciendo uso de una tarjeta protoboard.

Las fases en el desarrollo de un proyecto son tres:

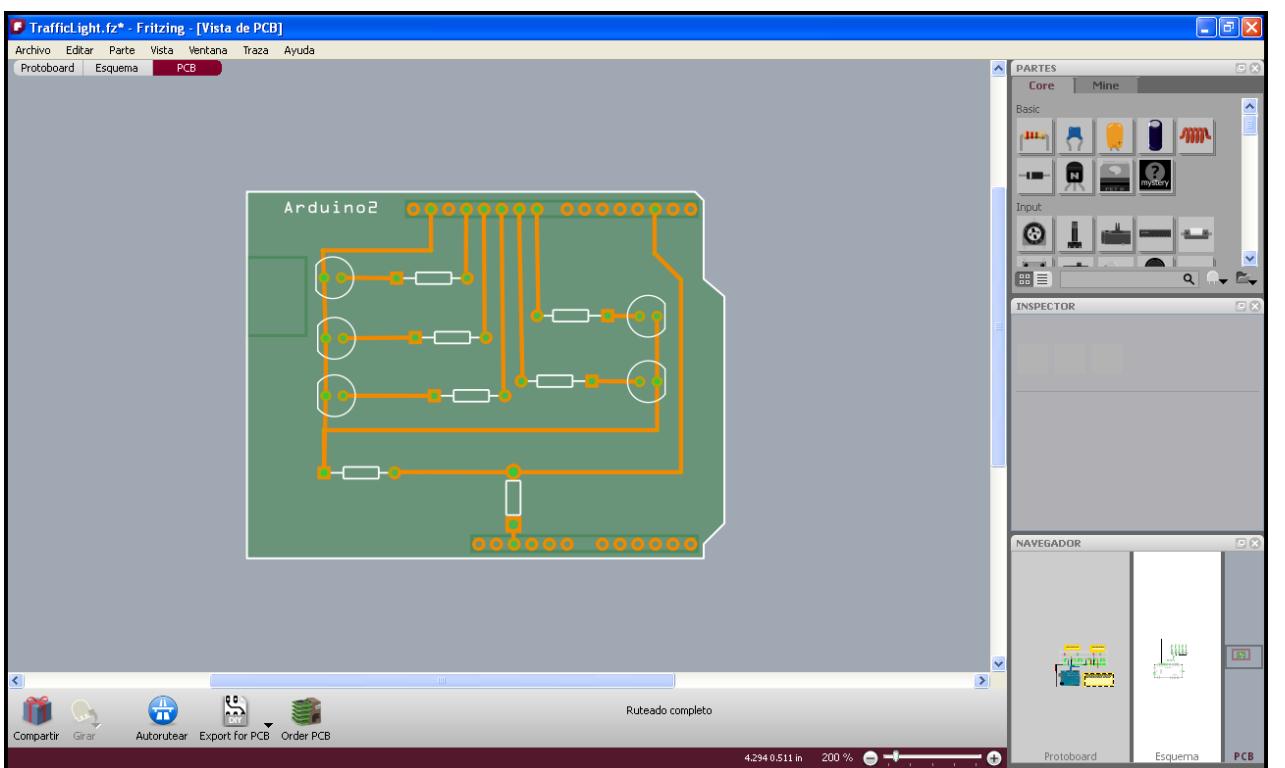
1. Elaboración del circuito con la ayuda de la Protoboard
2. Generación automática y edición, si procede, del esquema eléctrico
3. Generación y edición , si procede, de la PCB



Elaboración del prototipo haciendo uso de la Protoboard y todos los componentes de la librería.



Elaboración del esquema eléctrico del diseño.



Elaboración de la PCB final.

Opinión sobre la herramienta

La utilización de Fritzing en el “laboratorio de prototipado” es realmente muy válida. Podemos enumerar las ventajas:

- Facilidad de utilización de la herramienta
- Compendio lógico de la secuencia de trabajo en el diseño de prototipos.
- Posibilidades de exportación del diseño en formatos compatibles con las máquinas de elaboración de tarjetas de circuito impreso.
- Posibilidad de añadir nuevos elementos en la librería de componentes

4. Apéndice.

Donde encontrar las herramientas comentadas en este artículo:

Etoys (Squeak)

<http://tecnodacta.com.ar/gira/projects/physical-etoys/>

S4A (Scratch)

http://seaside.citilab.eu/scratch?_s=hvXNe4iKvwhZ3YaO&_k=6rrYXWDBfmL1NRUy

Labview

<https://decibel.ni.com/content/groups/labview-interface-for-arduino>

Firefly

<http://www.fireflyexperiments.com/>

MyOpenLab

<http://es.myopenlab.de/>

Minibloq

<http://blog.minibloq.org/>

Ardublock

<http://ardublock.com/>

Amici

<http://dimeb.informatik.uni-bremen.de/eduwear/about/>

ModKit

<http://www.modk.it/>

VirtualBreadBoard

<http://www.virtualbreadboard.net/>

Fritzing

<http://fritzing.org/>

5 de Noviembre de 2011 Versión de Documento: V1.0

José Manuel Ruiz Gutiérrez j.m.r.gutierrez@gmail.com

Blog de referencia: <http://josemanuelruizgutierrez.blogspot.com/>