



Manual de Instalación y Uso de la librería

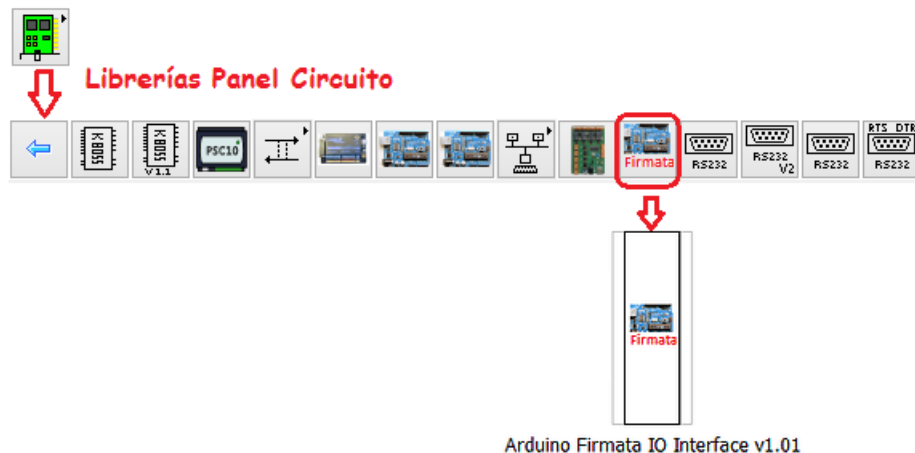
Firmata para **MyOpenLab**

Version 1.0

Prof. José Manuel Ruiz Gutiérrez

MyOpenLab + Firmata

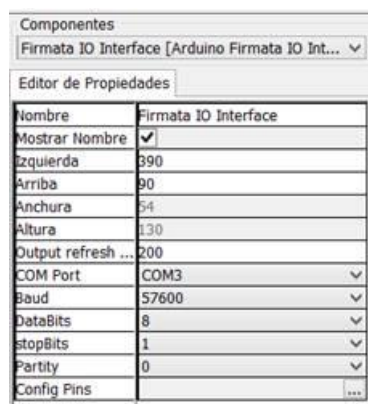
Es posible comunicar MyOpenLab con Arduino a través de la librería Firmata de gestión de comunicaciones. Esta librería es muy popular y está plenamente probada con excelentes resultados.



La librería se encuentra en el conjunto de librerías de comunicación perteneciente a los elementos del Panel Circuito.

La librería se encuentra en el conjunto de librerías de comunicación perteneciente a los elementos del Panel Circuito.

Dentro de la ventana de parámetros de este bloque de librería podemos modificar básicamente los siguientes:

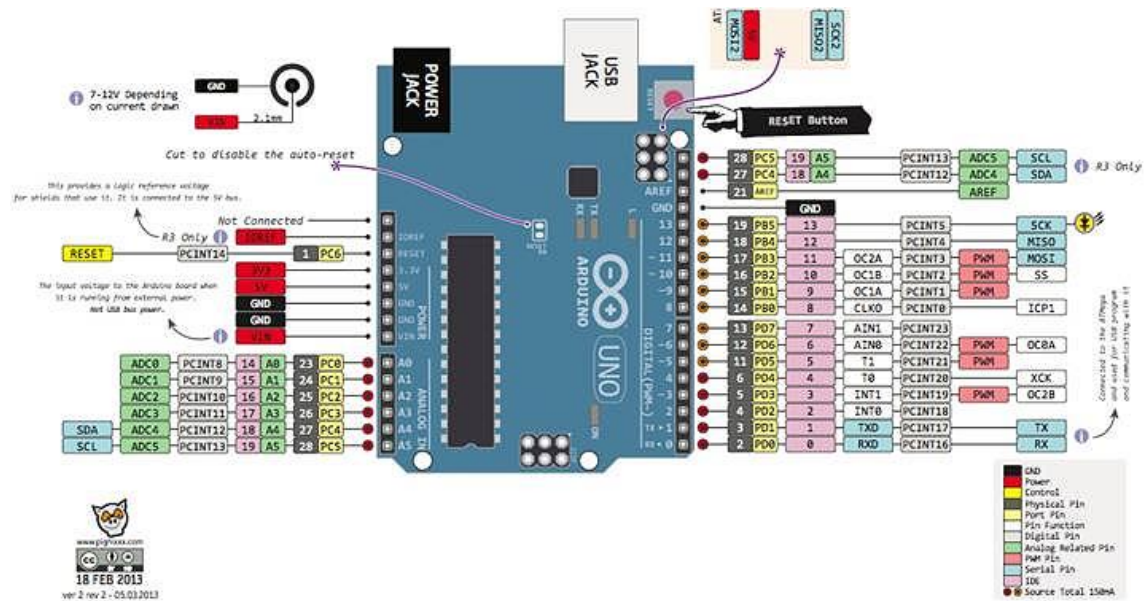


COM Port: Numero de puerto al que conectamos Arduino.

Baud: Velocidad de comunicación del puerto.

Config Pins: Configuración del modo de trabajo de cada uno de los pines de Arduino. Despliega una venta de configuración como la de la figura.

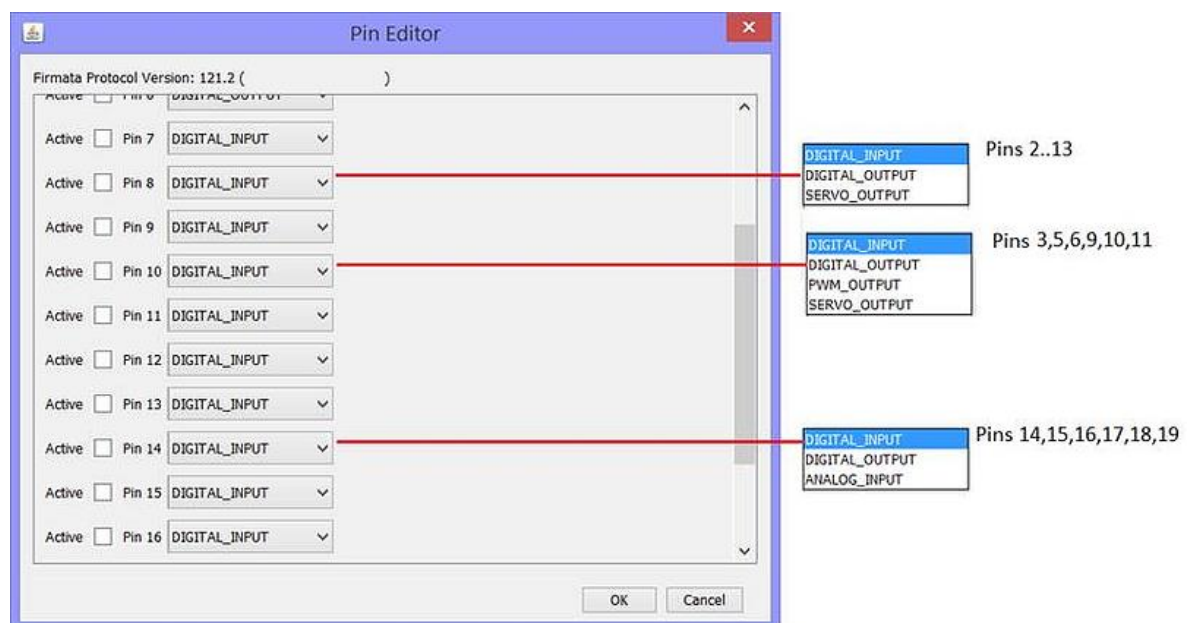
La numeración de los pines se realiza con valores que van desde el 0 al 19. En la imagen siguiente podemos ver a qué tipo de pin corresponde cada número.



Las posibilidades de configuración de cada pin se muestran al hacer clic sobre el número de Pin.

- **Entrada Digital**
- **Salida Digital**
- **Salida PWM**
- **Salida Servo**
- **Entrada Analógica**
- **Salida Analógica**

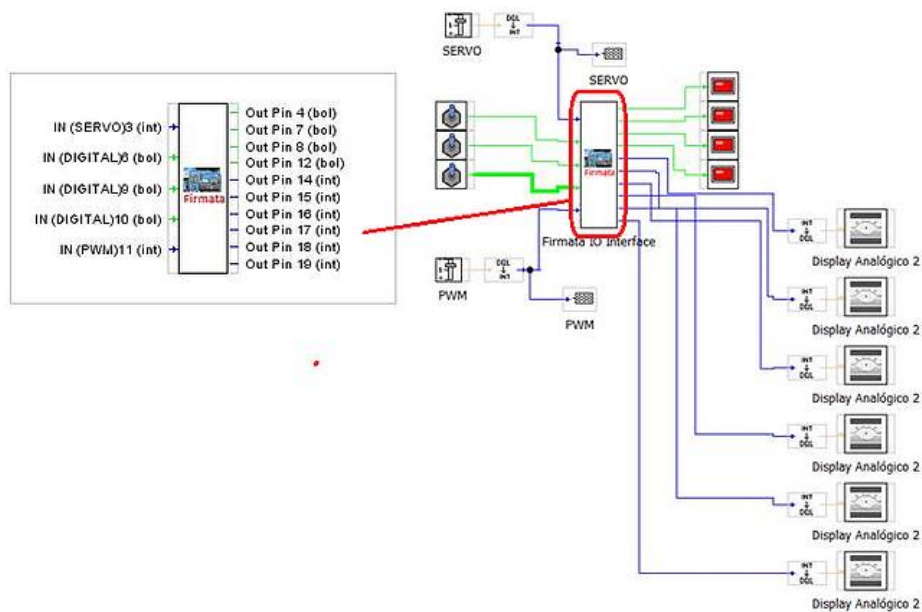
En todo caso se respeta la configuración propia de Arduino.



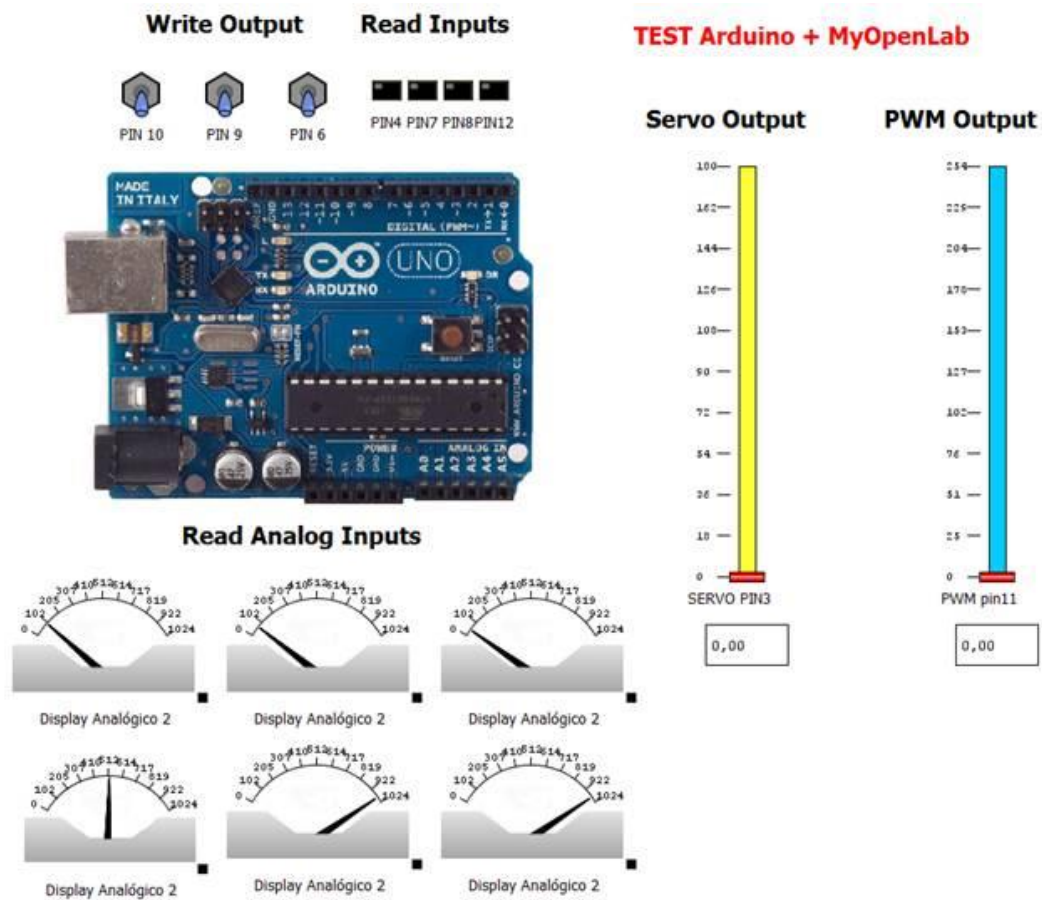
Para la designación de una salida PWM podemos colocar como parámetros los valores de duración de los pulsos Min Pulse y Max Pulse



A continuación se muestra el diagrama de Penal Circuito de una aplicación básica que sirve para realizar el test de comunicaciones de esta librería



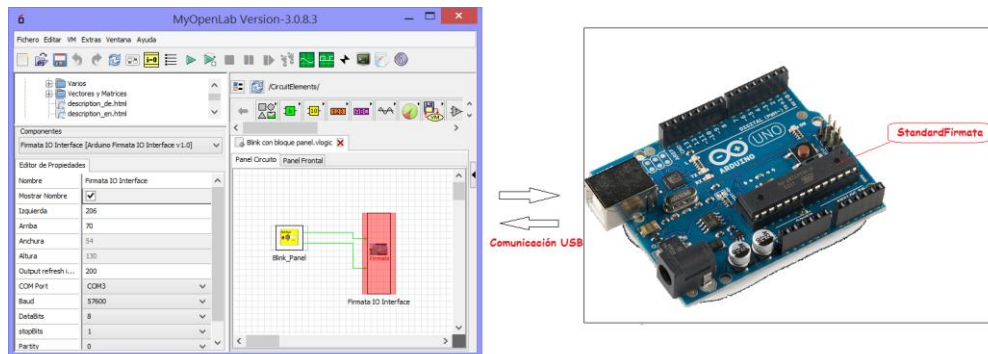
Este sería el diseño del Panel Frontal de la aplicación. Tal como se mostraría una vez que nos pongamos en modo RUN.



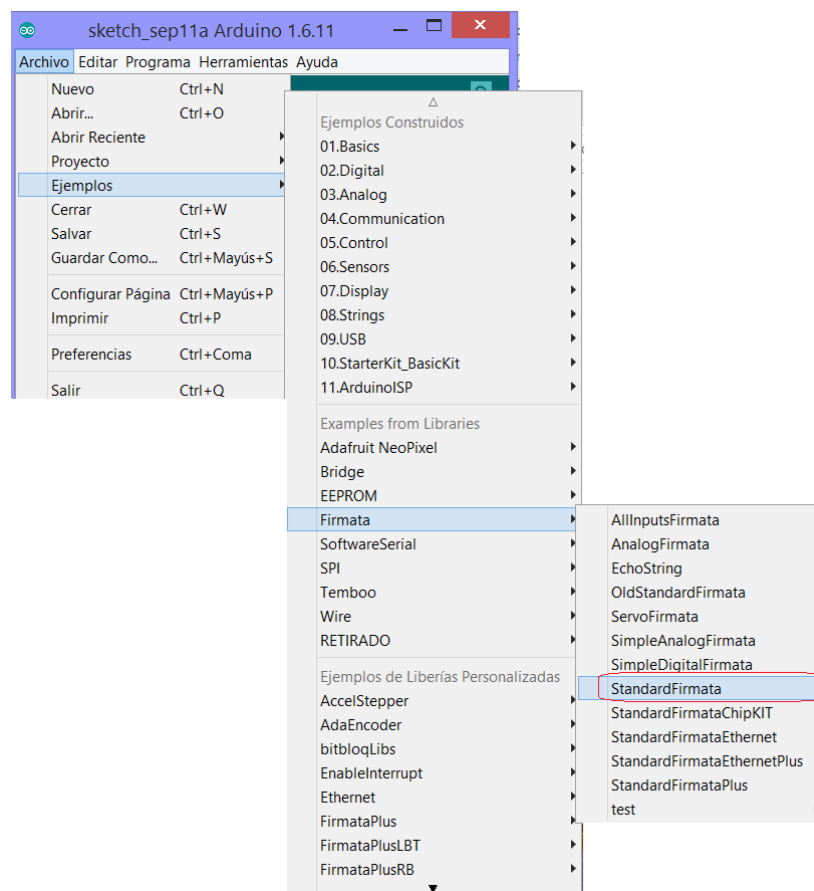
J.M.Ruiz

Antes de nada. Para empezar

Es importante que sepamos que la forma de trabajo de Arduino con MyOpenLab es en modo on-line, es decir siempre tendremos que tener conectada la tarjeta Arduino al puerto USB del PC.

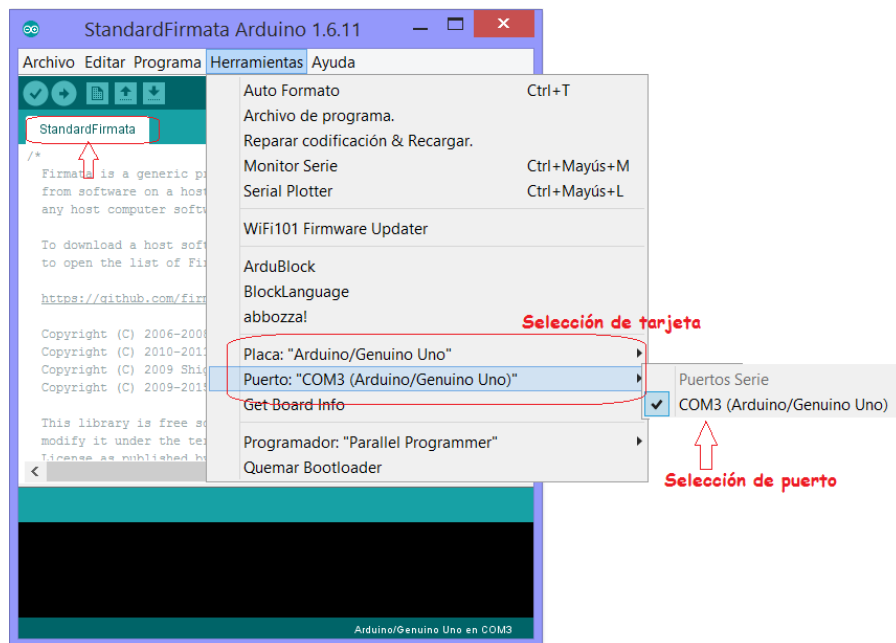


Para que MyOpenLab pueda comunicarse con Arduino debemos tener previamente descargado en la tarjeta Arduino el firmware StandardFirmata que será el encargado de gestionar la comunicación con MyOpenLab. En realidad, lo único que hace MyOpenLab es leer y escribir datos en la tarjeta quedando el control de las E/S al firmware que hemos grabado en ella.



¿Cómo grabar el firmware StandardFirmata?

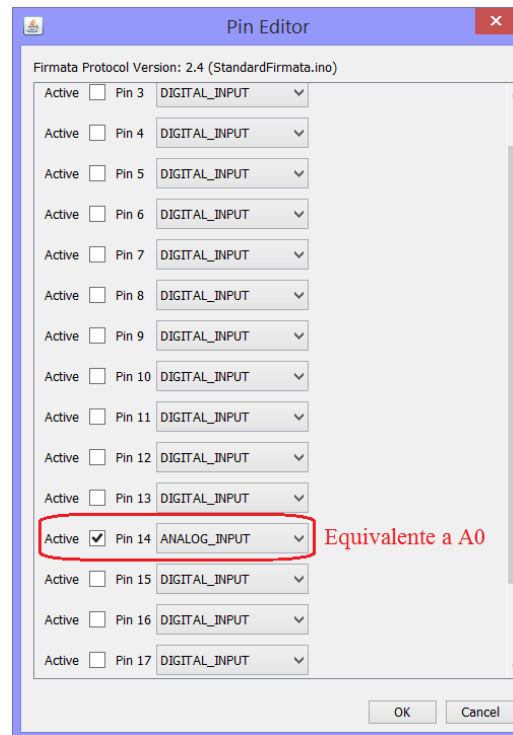
- Conectamos la tarjeta Arduino al PC
- Abrimos el software IDE Aduino.
- Buscamos en el menú la opción ejemplos y seleccionamos el grupo de ejemplos "Firmata" y dentro de él veremos un fichero llamado "StadardFirmata". Lo abrimos
- Seleccionamos la tarjeta que tenemos y el puerto de comunicación, despues descargamos el fichero StandardFirmata sobre la tarjeta.
- Una vez realizada la descarga podremos cerrar el IDE Arduino y abrir MyOpenLab. Hay que mantener conectada la tarjeta Arduino al abrir MyOpenLab de lo contrario se ponemos la librería Firmata de Myopenlab en el Panel Circuito no reconocerá el puerto de conexión.



Entrada Analógica

Con este ejemplo vamos a realizar la lectura de un canal analógico de Arduino A0 (PIN14) y vamos a representar su valor con la ayuda de un objeto "trazador grafico" en tiempo real.

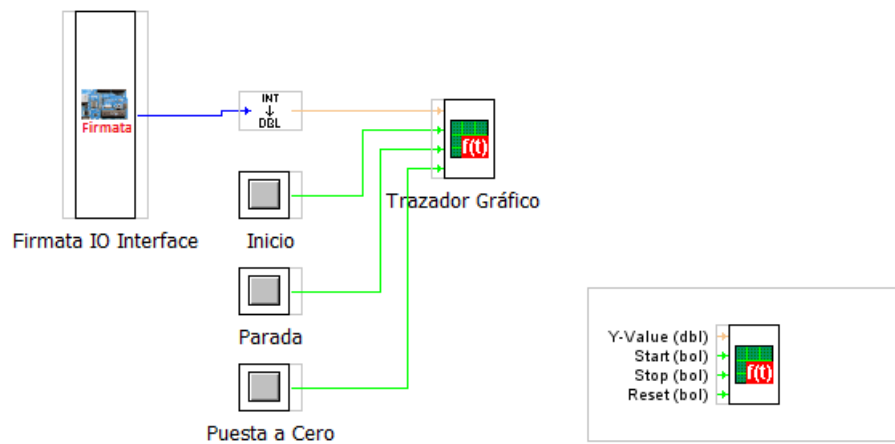
Configuración de la conexión:



En la figura vemos el esquema montado en el Panel Circuito en el que se distingue el bloque de comunicación con Arduino "Firmata IO Interface" del que sale la señal del canal A0 que es de tipo *Integer*, por lo que debemos pasarla a tipo *Double* (bloque *INT-DBL*-

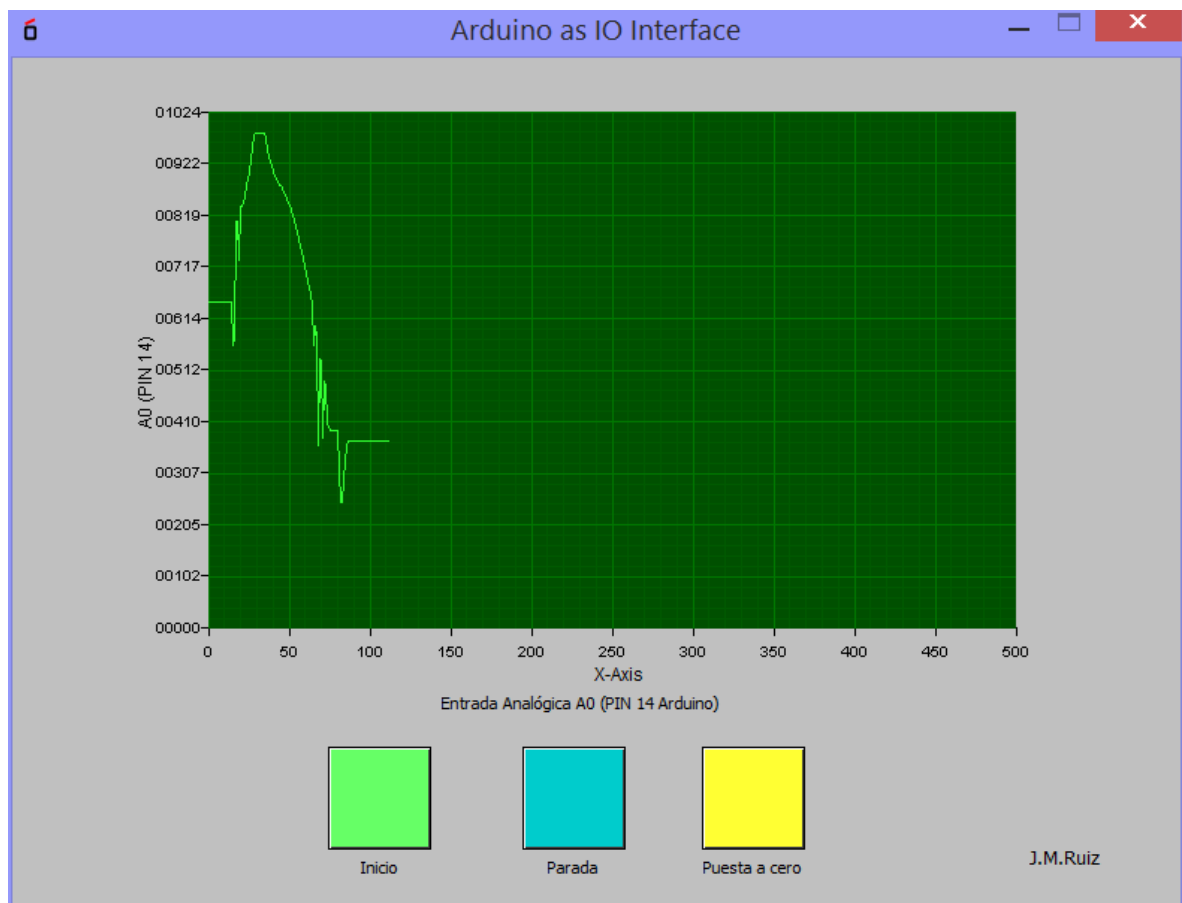
>*DBL*" para llevarla al trazador.





El trazador tiene tres entradas de control de tipo boolean: Start, Stop y Reset.

El "Panel Frontal" del montaje quedaria como se muestra en la figura.




Es importante que configuremos los parametros del trazador ara hacer mas cómoda la visualización.

Componentes

Entrada Analógica A0 (PIN 14 Arduino) [Time Gr... ▼]

Editor de Propiedades

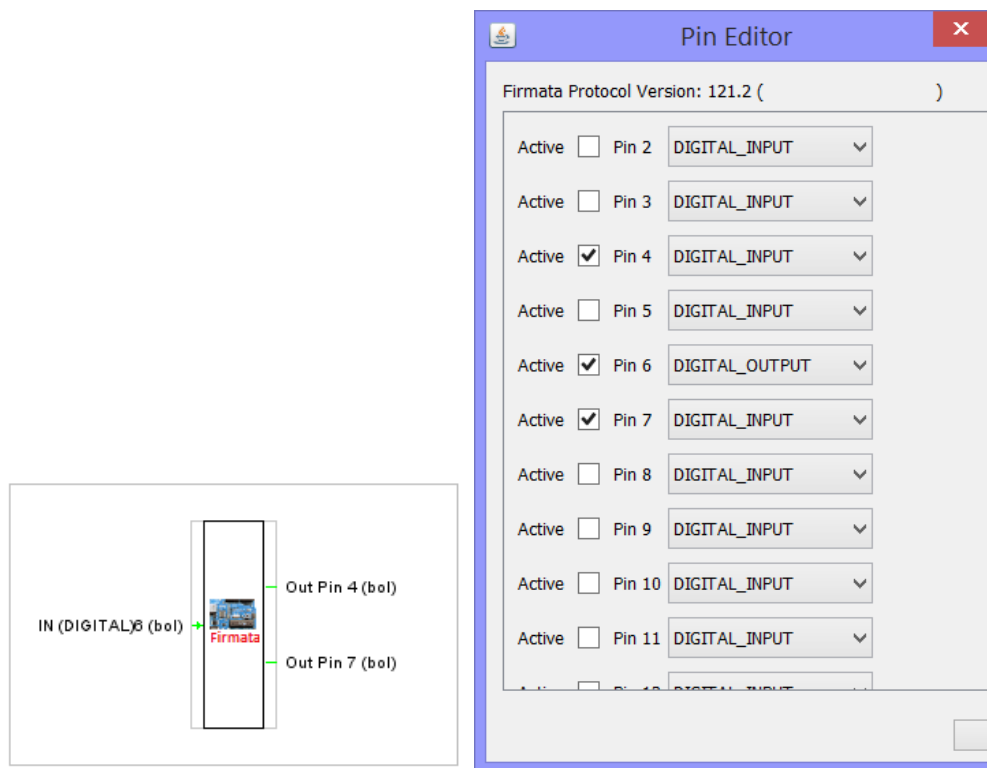
Nombre	Entrada Analógica A0 (PIN 14 Arduino)
Mostrar Nombre	<input checked="" type="checkbox"/>
Izquierda	80
Arriba	90
Visible	<input checked="" type="checkbox"/>
Anchura	570
Altura	300
Background Tr...	<input type="checkbox"/>
Grid Backgroun...	...
Grid Line Color	...
Grid Subline C...	...
Null Line Visible	<input checked="" type="checkbox"/>
Null Line Color	...
x/y Axis Font	Arial
x/y Axis Font C...	...
x/y Axis Visible	<input checked="" type="checkbox"/>
X-Axis Min	0.0
X-Axis Max	500.0
X-Axis Format	#0
X-Axis Text	X-Axis
X-Axis Text Font	Arial
X-Axis Text Color	...
Y-Axis Min	0.0
Y-Axis Max	1024.0
Y-Axis Autozoom	<input type="checkbox"/>
Y-Axis Format	#00000
Y-Axis Text	A0 (PIN 14)
Y-Axis Text Font	Dialog
Y-Axis Text Color	...
Auto Show Co...	<input type="checkbox"/>

Para proceder a la ejecución pulsamos el botón "Arrancar VM"  del entorno después de pulsar el botón "Inicio" veremos cómo se muestra la gráfica de la señal procedente del canal A0 (PIN 14 de Arduino)

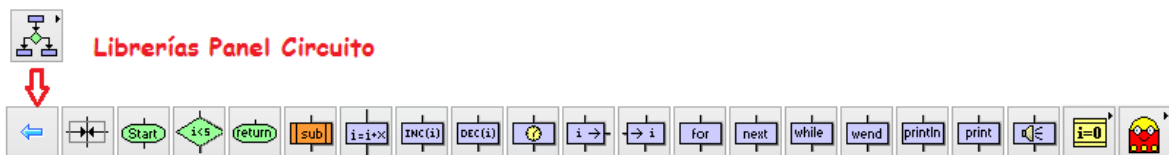
Función AND

En el siguiente montaje vamos a ver cómo podemos implementar una función lógica desde Arduino haciendo uso de la herramienta "FlowChart". La entradas A y B de la función se tomarán de los pines A=PIN4 y B=PIN 7 de la tarjeta Arduino y la salida será S= PIN6.



Lo primero que haremos será configurar los pines mediante la pantalla de configuración del bloque Firmata IO.

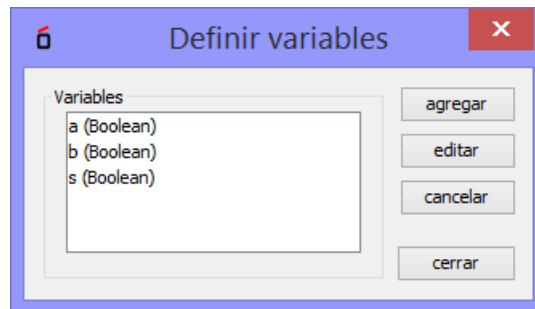




Utilizaremos la librería correspondiente de "Diagrama de flujo" de la que sacamos los bloques para montar la aplicación.

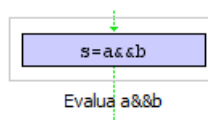



El montaje del diagrama de flujo de la aplicación, como siempre comienza por la colocación de un bloque "Start" activado permanentemente con un bloque de "Constante Binaria" dato digital colocado en valor "1".

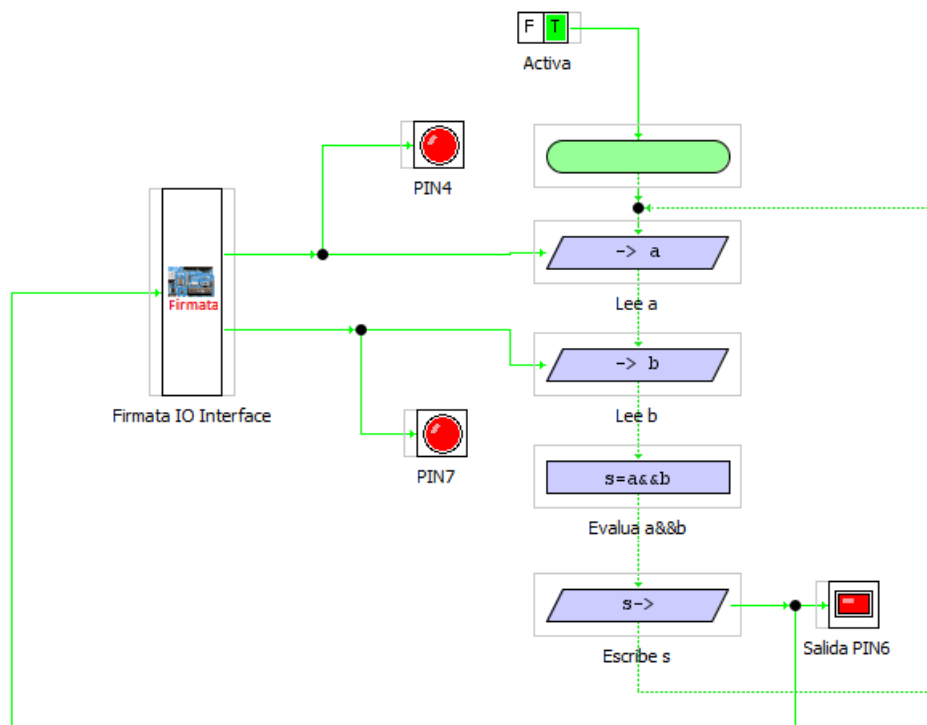
Como vamos a trabajar con la herramienta de librerías **"Diagramas de Flujo"**  debemos definir las variables de nuestro ejemplo y lo haremos en la ventana correspondiente que se activa con el botón **"Definir Variables"** .



Seguidamente se recogerán los valores "a" y "b" de las variables de entrada de PIN 4 y PIN7 con la ayuda de los bloques de tipo "Leer" . La operación AND se realiza con un bloque de tipo "Evaluar Expresión"  en donde escribimos la expresión de la función "s=a&b".



La obtención del resultado y su entrega a un indicador LED se realizará mediante un bloque del tipo "Escribir" .



Procederemos a etiquetar los elementos del "Panel Circuito" y los ubicamos en las posiciones que nos interese.



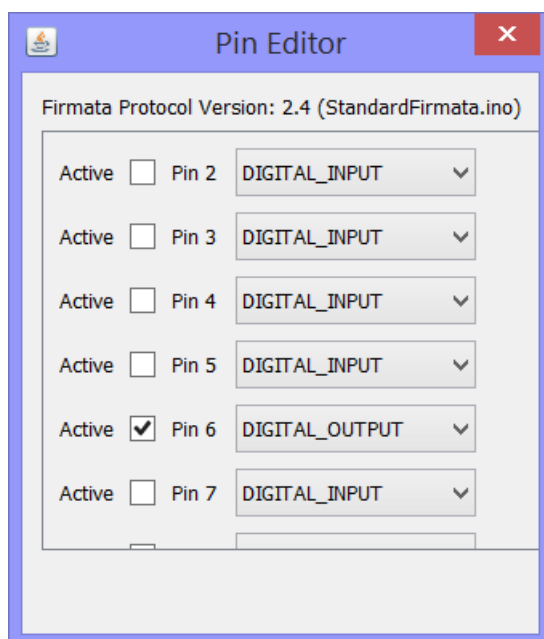
Finalmente iniciaremos la simulación y se probaran las distintas combinaciones de las variables de entrada "Entrada A Pin 4" y "Entrada B Pin 7" observando el estado del LED "Salida PIN 6" que se corresponde con la salida S

Salida Intermitente

La siguiente aplicación es la más sencilla de todas y suele servir para probar que las cosas funcionan. Se trata de encender y apagar la salida digital establecida en el pin PIN 6.

Para ello se selecciona el bloque de función de la librería correspondiente **Arduino IO Interface v1.11** y se procede a la conexión de un elemento de entrada de valor tipo Integer para la selección del COM de comunicaciones y un botón para iniciar la comunicación con la tarjeta Arduino. Esta operación será común y necesaria en todas las aplicaciones que montemos.

Definiremos también el PIN6 como salida en la pantalla de designacion de E/S.



La señal que hemos de mandar en forma de pulsos se obtiene de un bloque de función llamado **Temporizador** de la librería **Utilidades** del **Panel de Circuito**. A este bloque le damos los valores correspondientes de tiempo encendido (Tiempo activado) y tiempo apagado (Tiempo desactivado).

Colocamos un led para mostrarnos información del estado de la señal en el lado físico de la tarjeta Arduino (Pin Digital 13)

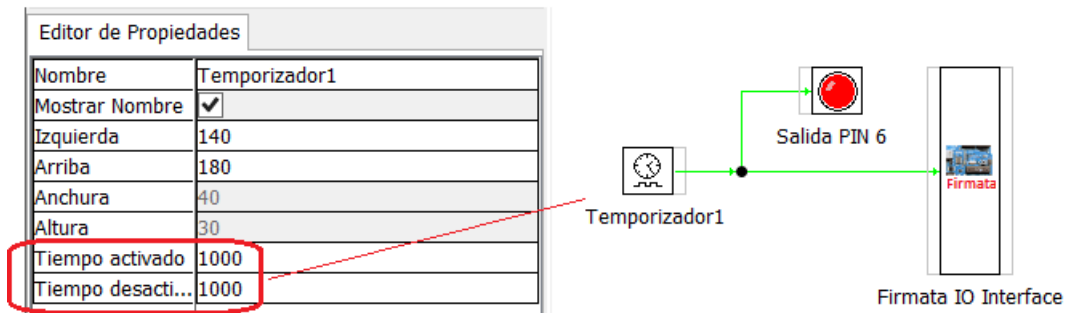
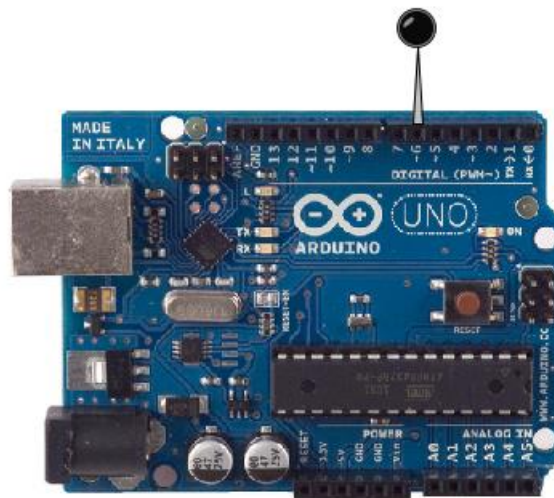


Figura del Panel Circuito

En el Panel Frontal podremos una imagen de Arduino haciendo uso del objeto **Raster Imagen** de la librería **Decoración** perteneciente al Panel Frontal. Pondremos

también un texto mediante el objeto **Label** de la misma librería. El resto de objetos, el boton y el Spinner (INT) aparecen dado que se pusieron en el Panel Circuito.

Blink Salida PIN 6



J.M.Ruiz

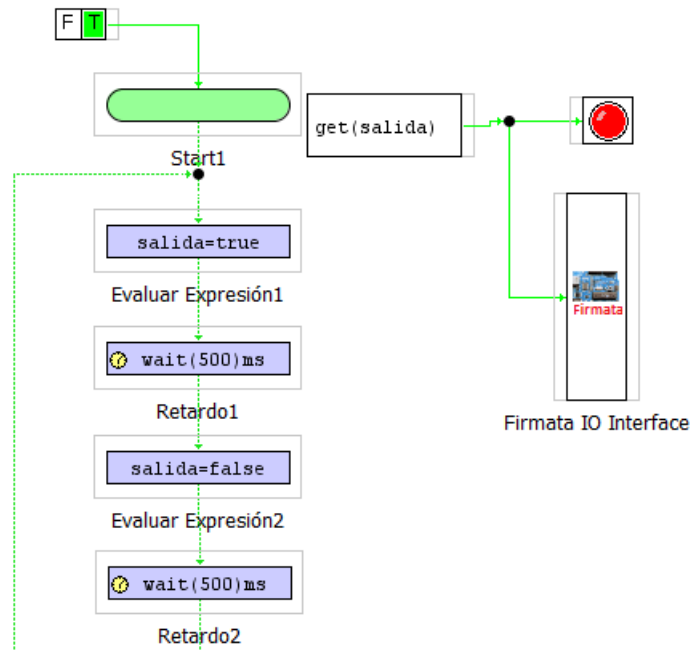
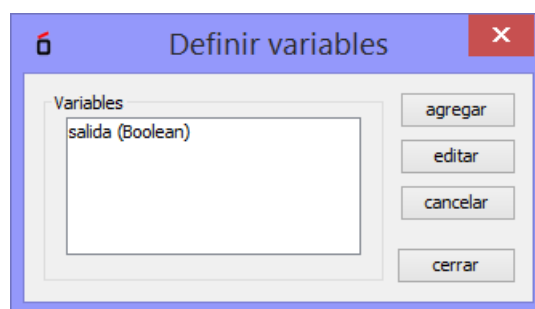
Para iniciar la aplicación ejecutamos mediante el botón Arrancar VM del entorno. Seleccionamos el puerto con el selector y pulsamos el Botón “Inicia comunicación con arduino”

Si todo va bien veremos parpadear la salida 13 de Arduino en la que lógicamente habremos colocado un diodo led de acuerdo a la figura siguiente.

Salida Intermitente (otra opción)

Se trata de realizar una aplicación haciendo uso de la librería de programación de Diagramas de Flujo del Panel de Circuito de MyOpenLab. Se crea una variable que se llama “salida” y se activa y desactiva cada 500 ms de acuerdo con el valor que se establece en el correspondiente bloque Retardo de la librería.

La variable salida me da un bloque Recoge dato se lleva a la salida del PIN13 del bloque de librería de Arduino y a un Led para que sirva de indicar del estado de la señal en la pantalla





Comparador de Entrada analógica con una constante.

En esta aplicación se trata de realizar la comparación de una de las señales de entrada analógica “i” de la tarjeta Arduino (procedente de un sensor) con un valor ajustable a través de un Slider “P” en la pantalla del ordenador. En función del resultado de la comparación se activará la salida PIN 13 de Arduino (0 si $i < p$ y 1 en caso contrario)

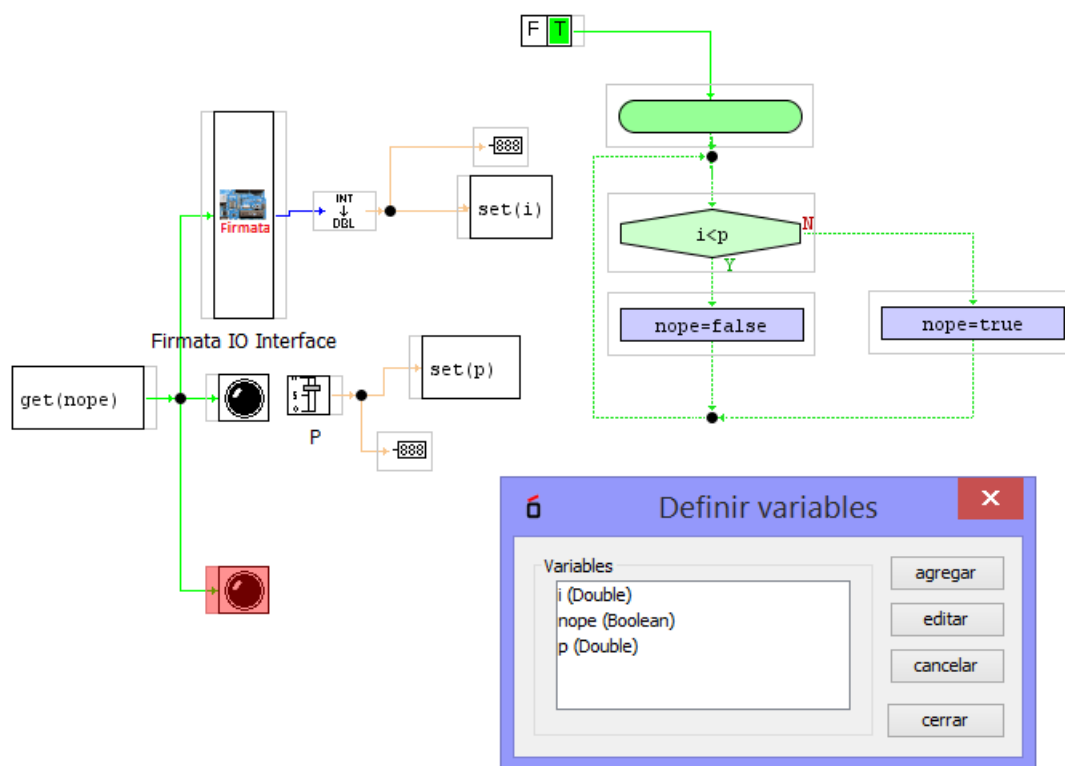
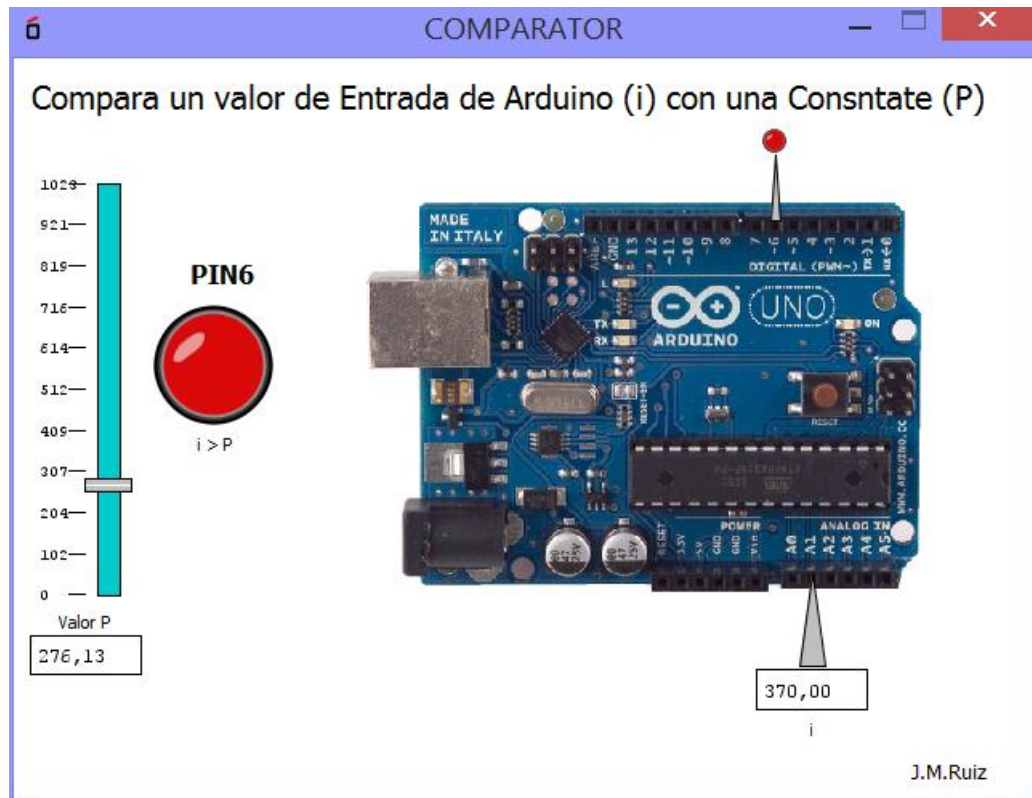


Diagrama funcional de la aplicación contenido en el Panel circuito de MyOpenLab

Las variables que debemos definir en la aplicación son las mostradas en la ventana de definición de variables de MyOpenLab.



En la figura se muestra el Panel Frontal de la aplicación

