



# Manual de Instalação e Utilização da Biblioteca

## Firmata para MyOpenLab

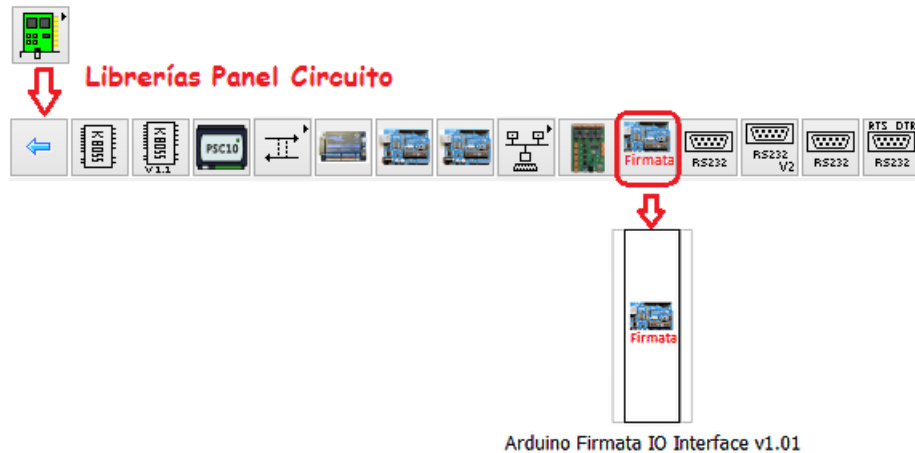
### Versão 1.0

Prof. José Manuel Ruiz Gutiérrez

Traduzido para Português por:  
Paulo Galvão  
V 0.1

# MyOpenLab + Firmata

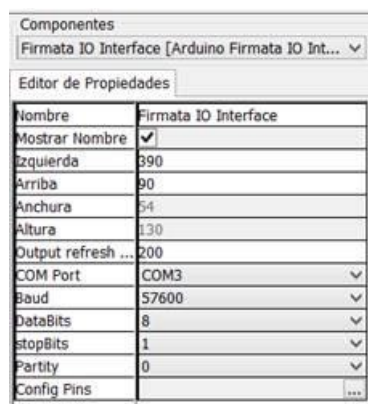
É possível comunicar MyOpenLab com Arduino através da biblioteca Firmata de gestão de comunicações. Esta biblioteca é muito popular e está plenamente testada e com excelentes resultados.



A biblioteca encontra-se no conjunto de bibliotecas de comunicação pertencente aos elementos do Painel Circuito.

A biblioteca encontra-se no conjunto de bibliotecas de comunicação pertencente aos elementos do Painel Circuito.

Dentro da janela de parâmetros deste bloco de biblioteca podemos modificar basicamente os seguintes:

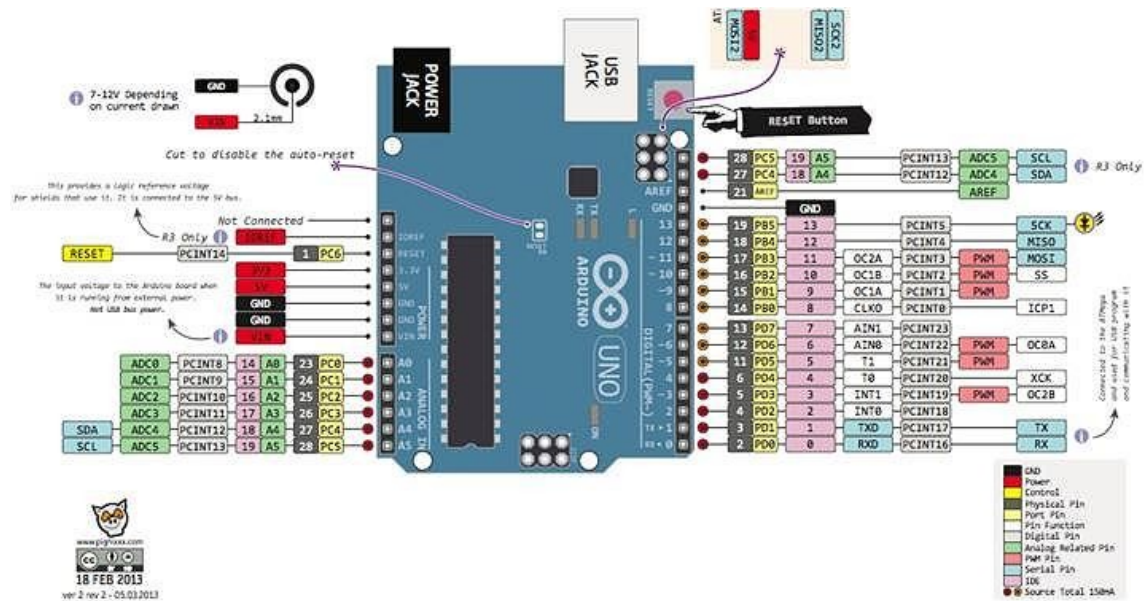


**COM Port:** Número da porta a que ligamos o Arduino.

**Baud:** Velocidade de comunicação da porta.

**Config Pins:** Configuração do modo de trabalho de cada um dos pinos de Arduino. Aparece uma janela de configuração como a da figura.

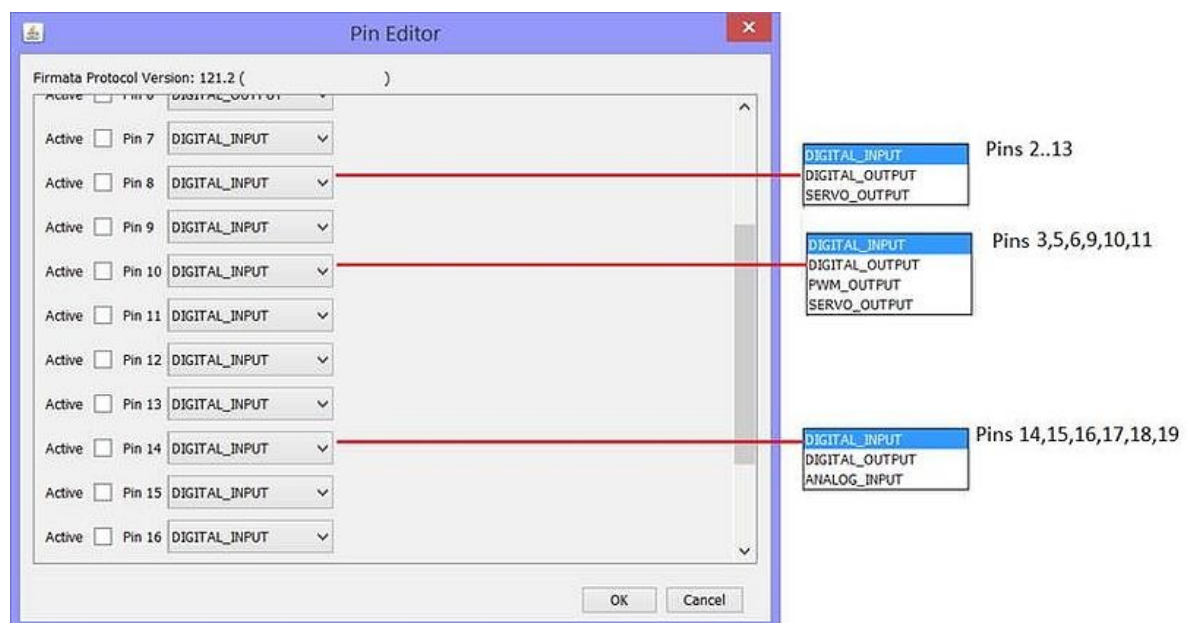
A numeração dos pinos realiza-se com valores que vão desde o 0 a 19. Na imagem seguinte podemos ver a que tipo de pino corresponde cada número.



As possibilidades de configuração de cada pino mostram-se ao se fazer clic sobre o número do Pino.

- **Entrada Digital**
- **Saída Digital**
- **Saída PWM**
- **Saída Servo**
- **Entrada Analógica**
- **Saída Analógica**

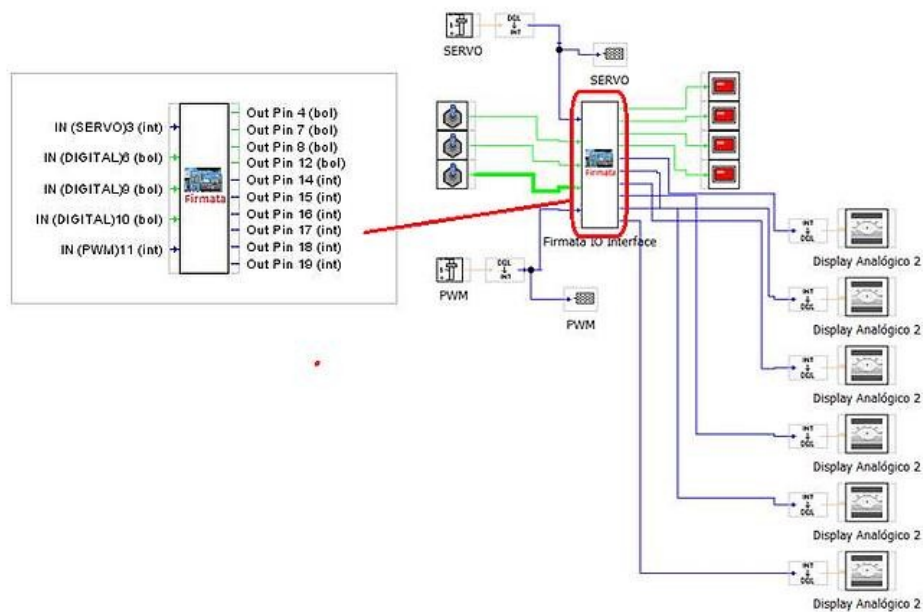
Em todos casos respeita-se a configuração própria do Arduino.



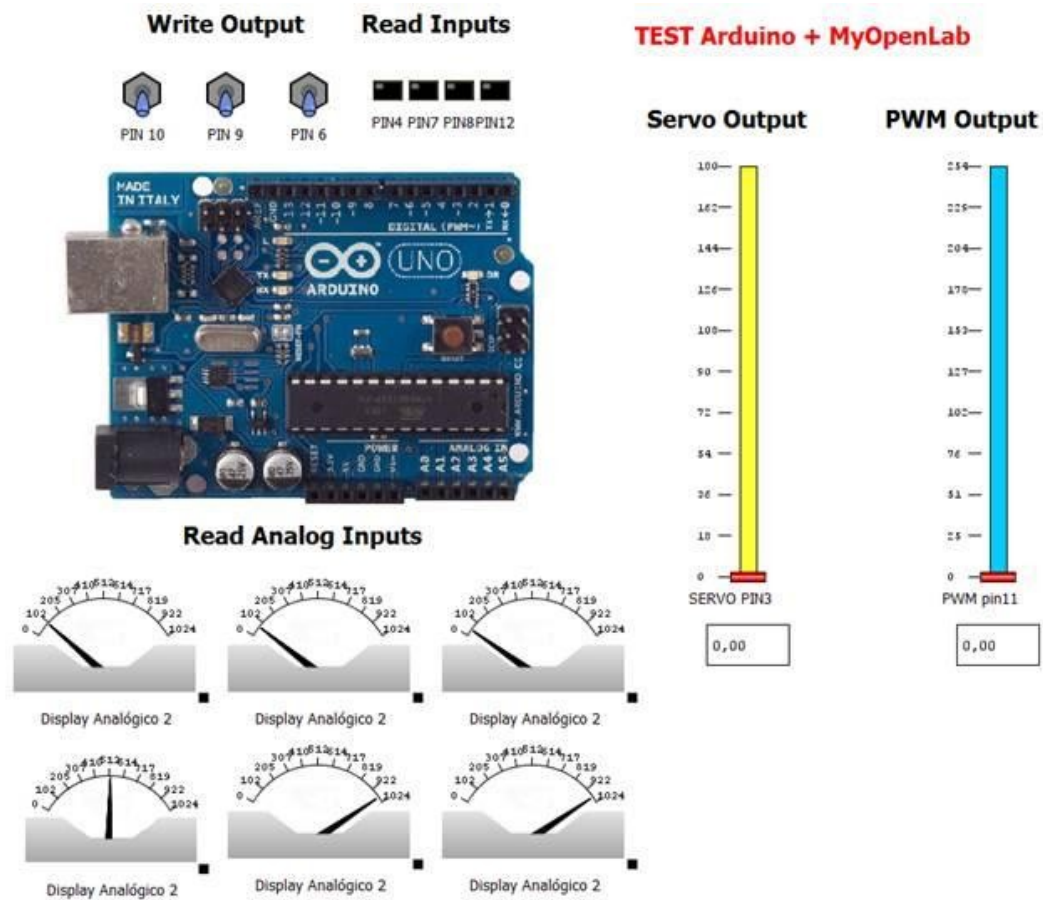
Para a designação de uma Saída PWM podemos colocar como parâmetros os valores de duração dos impulsos Min Pulse e Max Pulse



A seguir mostra-se o diagrama de Painel do Circuito de uma aplicação básica que serve para realizar o teste de comunicações desta biblioteca.

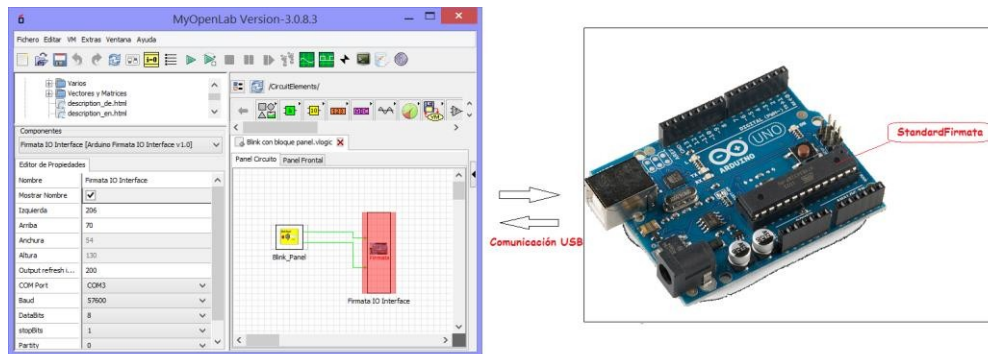


Este seria o desenho do Painel Frontal do programa, tal como se veria quando se coloca no modo RUN.

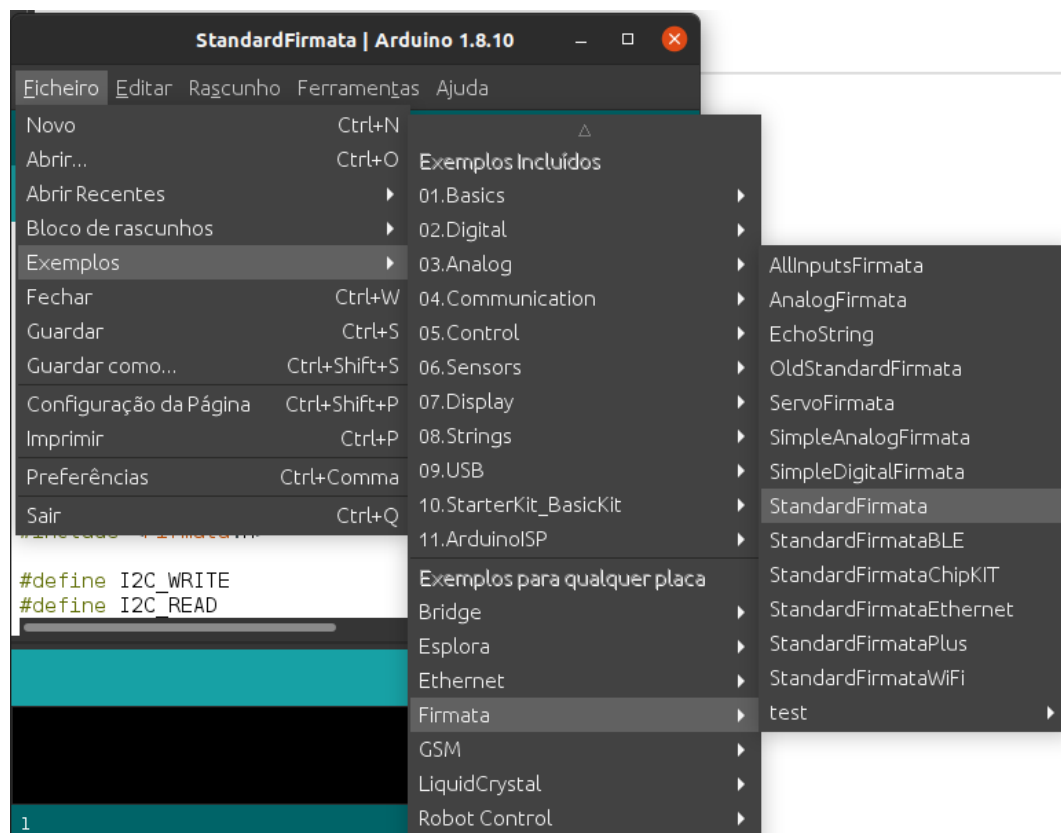


# Antes de começar, tenha em mente que:

A forma de trabalho de Arduino com MyOpenLab é em modo on-line, isto é, temos de ter a placa do Arduino sempre ligada à porta USB do PC.

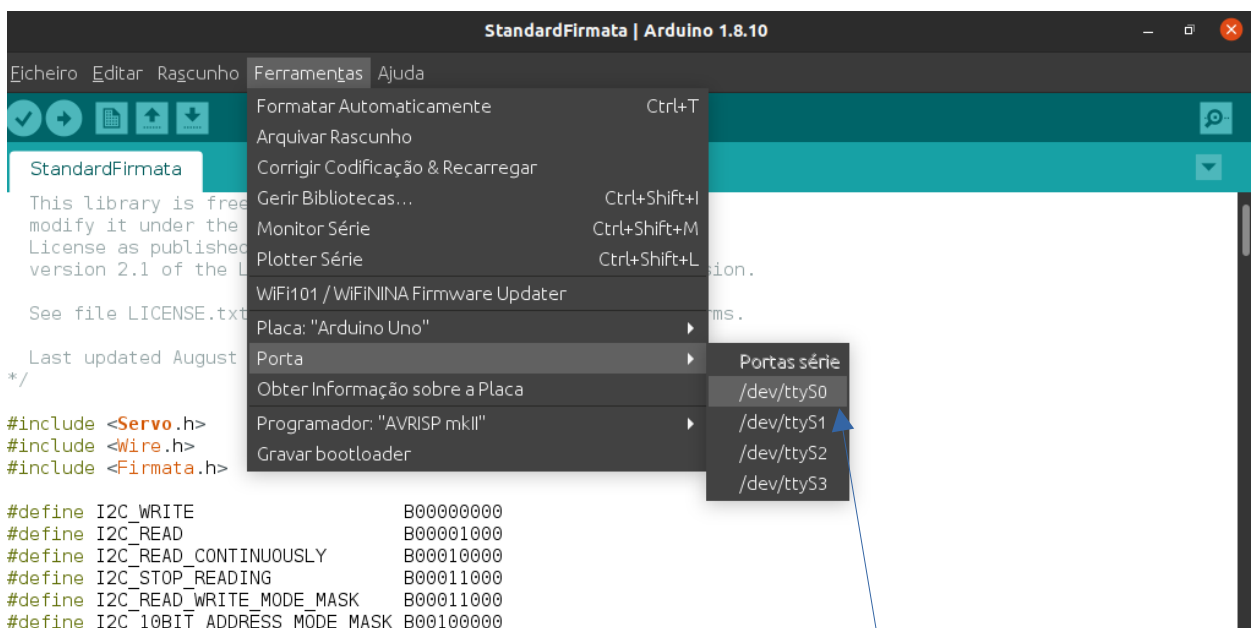


Para que MyOpenLab se possa comunicar com Arduino devemos ter previamente descarregar na placa do Arduino o firmware StandarFirmata que será o encarregado de gerir a comunicação com MyOpenLab. Na verdade, o que o MyOpenLab faz é ler e escrever dados na placa deixando o controlo das E/S ao firmware que previamente gravámos nela.



Como gravar o firmware StandardFirmata?

- Ligamos a placa Arduino ao PC
- Abrimos o software Arduino IDE.
- Procuramos no Menu Ficheiro a opção Exemplos e Seleccionamos o grupo de Exemplos "Firmata" e dentro de o veremos um ficheiro chamado "StandardFirmata". Abra o ficheiro.
- Seleccionamos a placa que temos na porta de comunicação, depois descarregamos o ficheiro StandardFirmata para a placa.
- Uma vez realizada o download podemos fechar o Arduino IDE e abrir MyOpenLab. Temos que manter ligada a placa Arduino ao abrir o MyOpenLab porque se isso não acontecer a biblioteca Firmata de Myopenlab no Panel Circuito na reconhecerá a porta de ligação.

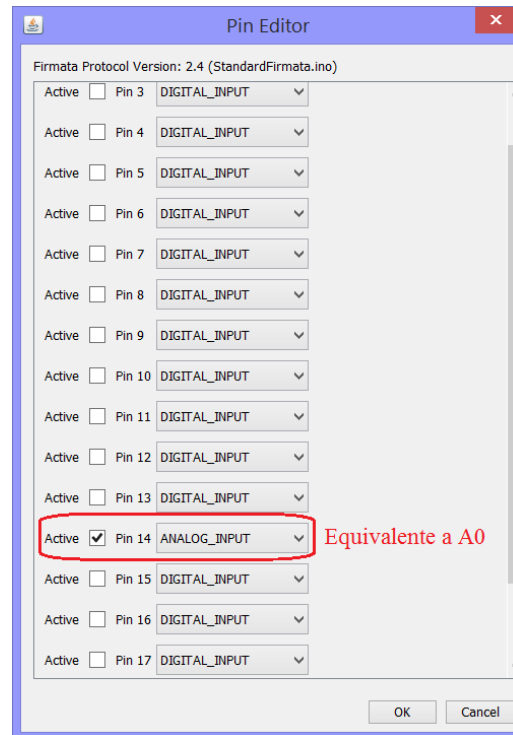


Em Windows, será por exemplo Com1

# Entrada Analógica

*Neste exemplo vamos realizar a leitura de um canal analógico do Arduino na porta A0 (PIN14) e vamos a representar o seu valor com a ajuda de um objeto "traçador gráfico" em tempo real.*

Configuração da ligação:

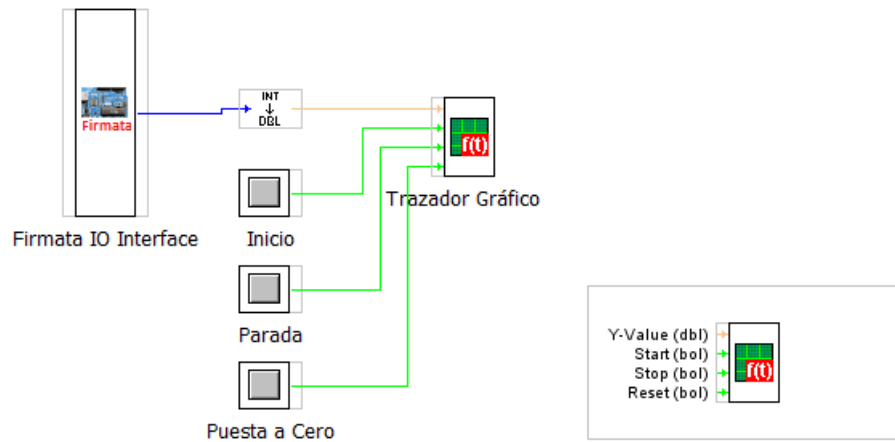


Na figura vemos o esquema montado no Panel Circuito no qual se distingue o bloco de comunicação com Arduino "*Firmata IO Interface*" de onde sai o sinal do canal A0 que é de tipo *Integer*, pelo que o devemos passar para o tipo *Double* (bloco *INT->DBL*).

>DBL"para levá-lo ao gráfico.

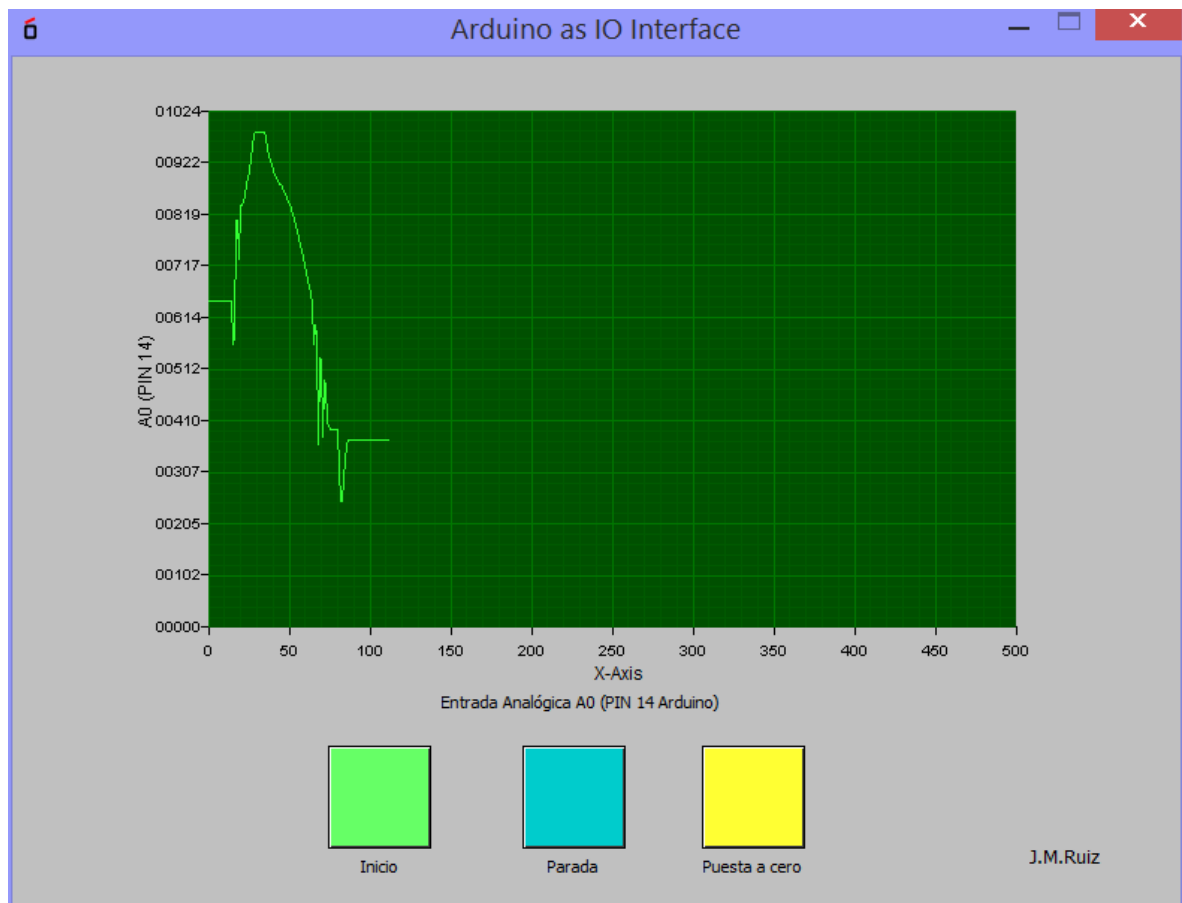






O gráfico tem três entradas de controlo do tipo booleano: Start, Stop e Reset.

O "Painel Frontal" da montagem ficará como se mostra na figura.




É importante que configuremos os parâmetros do gráfico para termos uma visualização mais cómoda.

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 começar o programa pressionamos o botão "Arrancar VM"  O retorno depois de pressionar o botão "Inicio" veremos como se mostra o gráfico do sinal vindo do canal A0 (PIN 14 do Arduino)

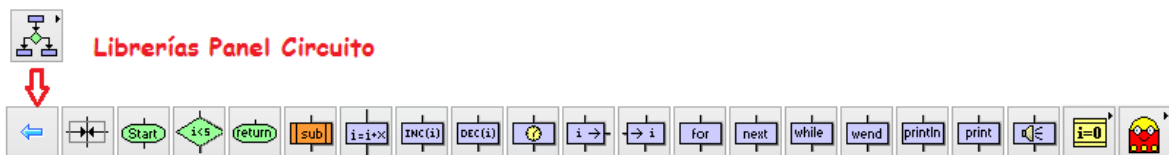
# Função AND



*Na montagem seguinte vamos ver como podemos implementar uma função lógica com o Arduino fazendo o uso da ferramenta "FlowChart". A entrada A e B da função serão formadas pelos pinos A=PIN4 e B=PIN 7 da placa Arduino e a Saída será o S= PIN6.*


Primeiro iremos configurar os pinos através da janela de configuração do bloco Firmata IO.



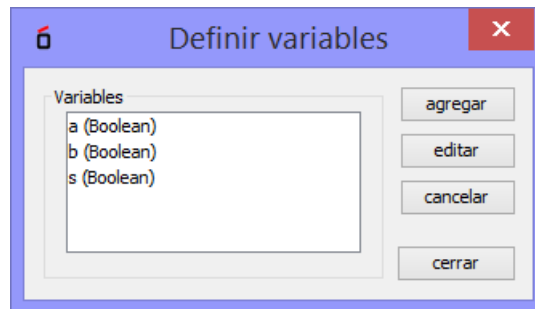
Utilizaremos a biblioteca correspondente de "Diagrama de fluxo" da que retiraremos os blocos para criar o programa.




A montagem do diagrama de fluxo do programa, como sempre começa pela colocação de um bloco "Start"  activado permanentemente com um bloco de "Constante Binária", dado digital , colocado com o valor "1".

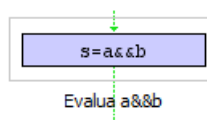
Como vamos trabalhar com a ferramenta de bibliotecas "**Diagramas de Fluxo**"  devemos definir as variáveis do nosso exemplo e o faremos na janela correspondente que se activa com


o botão "**Definir Variables**" .

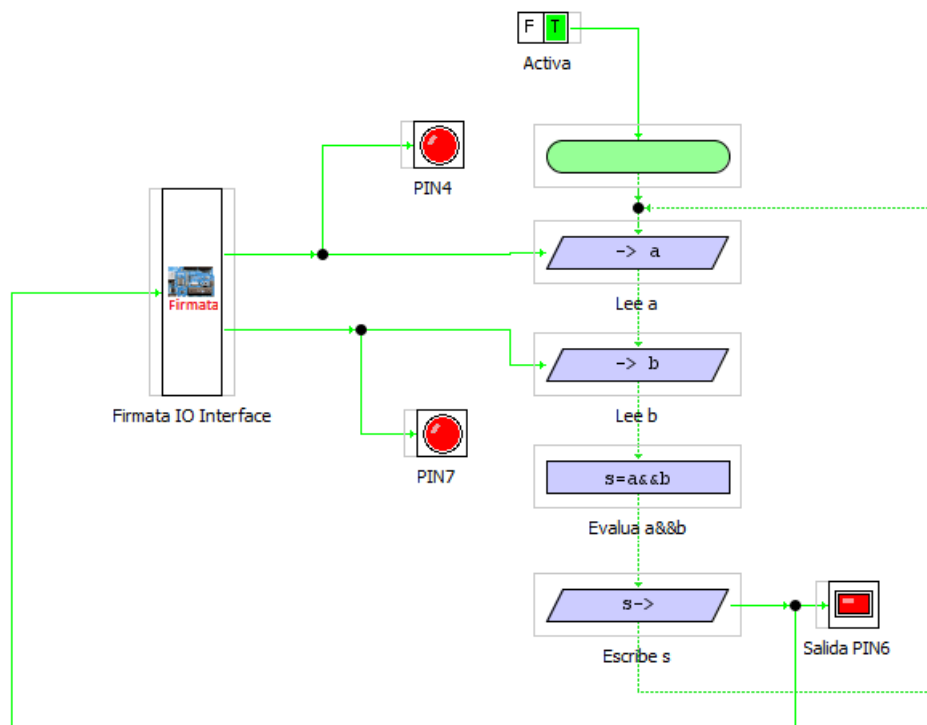


Seguidamente se recolherão os valores "a" e "b" das variáveis de entrada do PIN 4 e PIN7 com a ajuda dos blocos de tipo "Leer" .

"Evaluar Expresión"  onde escrevemos a expressão da função "s=a&b".



A obtenção do resultado e a sua entrega a um indicador LED se realizará através de um bloco do tipo "Escribir" .



Poderemos colocar etiquetas nos elementos do "Panel Circuito" e os posicionamos da forma que acharmos mais interessante.



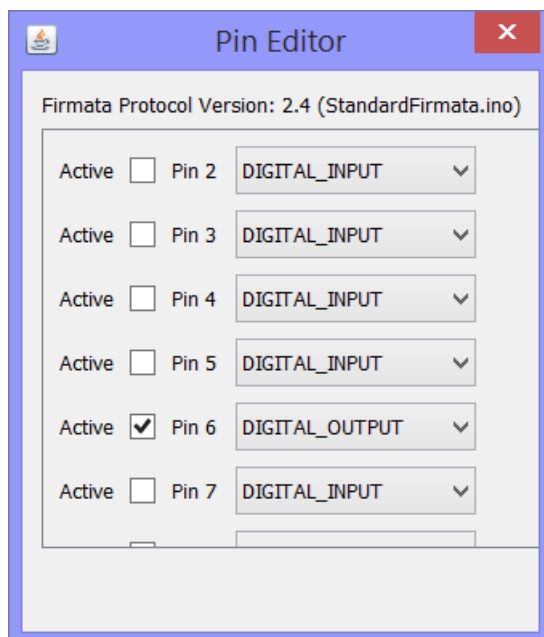
Finalmente iniciaremos a simulação e se testarão as distintas combinações das variáveis de entrada "Entrada A Pin 4" e "Entrada B Pin 7" observando o estado do Led "Saída PIN 6" que corresponde à Saída S.

# Saída Intermitente

*O programa seguinte é o mais sensível de todos e apenas vai servir para provar que as coisas funcionam. Trata de acender e apagar a Saída digital estabelecida no pin PIN 6.*

Para se fazer isto selecciona-se o bloco de função da biblioteca correspondente **Arduino IO Interface v1.11** e procede-se à ligação de um elemento de entrada do valor do tipo Integer para a seleção da porta COM de comunicação e um botão para iniciar a comunicação com a placa Arduino. Esta operação será comum e necessária em todas as aplicações que montemos.

Definiremos também o PIN6 como Saída na janela de designação de E/S.



O sinal que temos de enviar na forma de impulsos obtém-se de um bloco de função chamado **Temporizador** da biblioteca **Utilidades do Panel de Circuito**. A este bloco damos os valores correspondentes ao tempo de aceso (Tempo activado) e tempo apagado (Tempo desactivado).

Colocamos um led para mostrar-mos a informação do estado do sinal no lado físico da placa Arduino (Pin Digital 13)

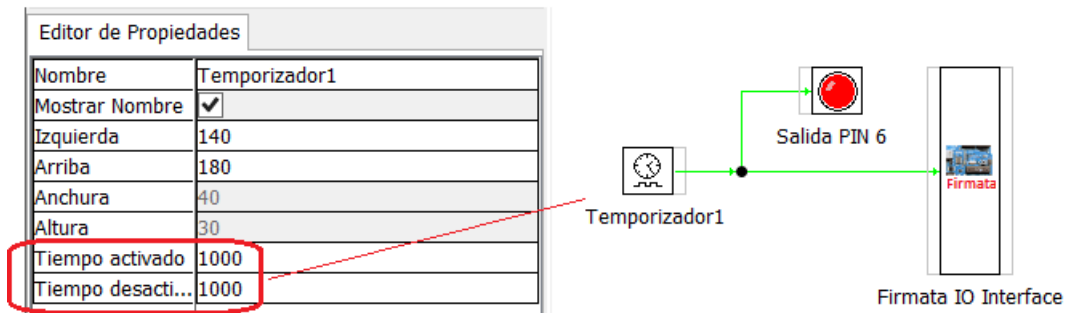


Figura do Panel Circuito

No Painei Frontal podemos ver uma imagem do Arduino fazendo uso do objeto **Raster Imagen** da biblioteca **Decoración** pertencente ao Panel Frontal.

Colocaremos também um texto através do objeto **Label** da mesma biblioteca.

O resto dos objetos, o botão e o Spinner (INT) aparecem, porque se puseram no Panel Circuito.

Blink Salida PIN 6



Para iniciar o programa pressionamos o botão Arrancar VM. Seleccionamos a porta com o selector e pressionamos o Botão “Inicia comunicação com Arduino”.

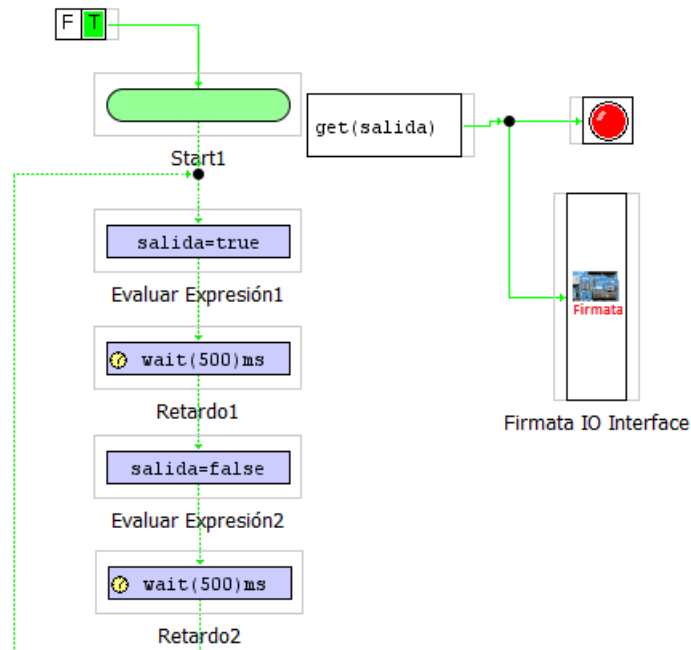
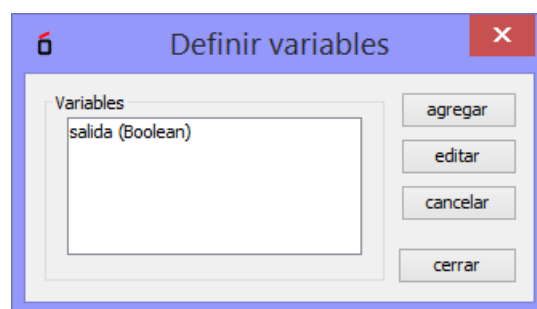
Se tudo estiver bem, veremos a Saída 13 a piscar (cintilar) do Arduino de acordo com a lógica que teremos colocado no led de acordo com a figura seguinte.



## Saída Intermitente (outra opção)

*Trata-se de realizar um programa fazendo o uso da biblioteca de programação de Diagramas de Fluxo do Painel de Circuito do MyOpenLab. Cria-se uma variável que se chama “Saída” e se activa e desactiva cada 500 ms de acordo com o valor que se estabelece no correspondente bloco de atraso da biblioteca.*

A variável de Saída faz-se através do bloco Recoge Dato e a leva à Saída do PIN13 do bloco da biblioteca de Arduino e a um Led, que serve para indicar o estado do sinal da janela.





# Comparador de Entrada analógica com uma constante.

*Neste programa trata-se de realizar a comparação de um dos sinais de entrada analógica “i” da placa Arduino (procedente de um sensor) com um valor ajustável através de um Slider “P” na janela do ordenador. Em função do resultado da comparação se activará a Saída PIN 13 de Arduino (0 se  $i < p$  e 1 caso contrário)*

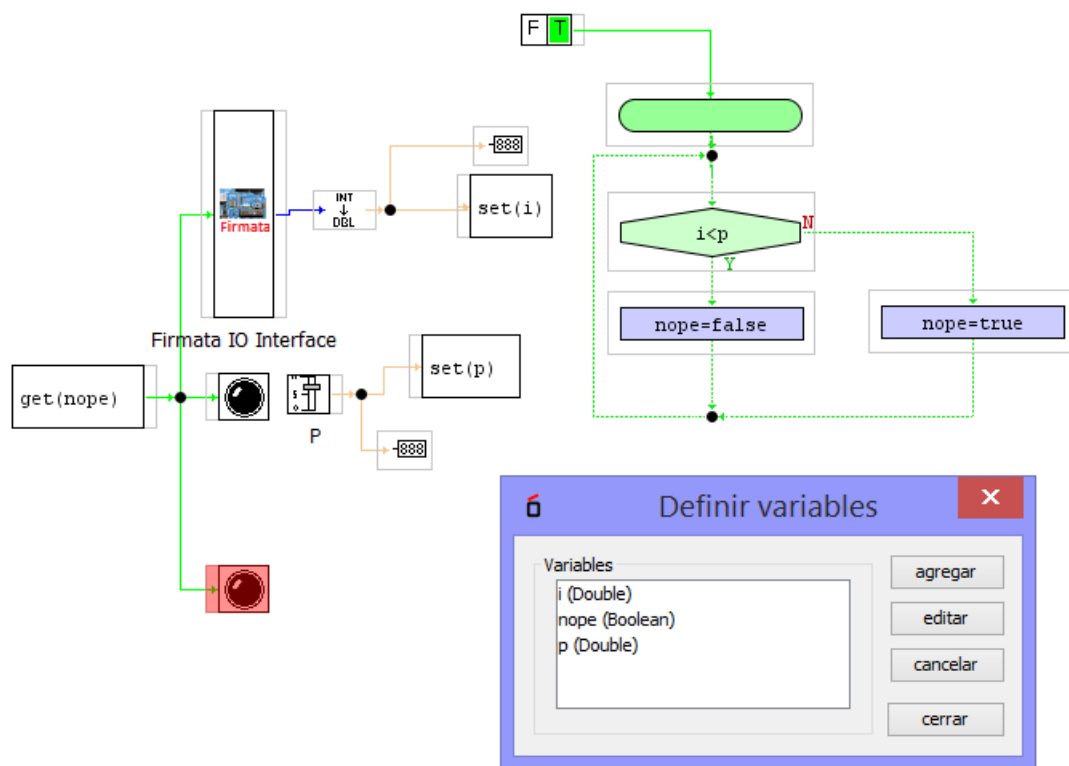
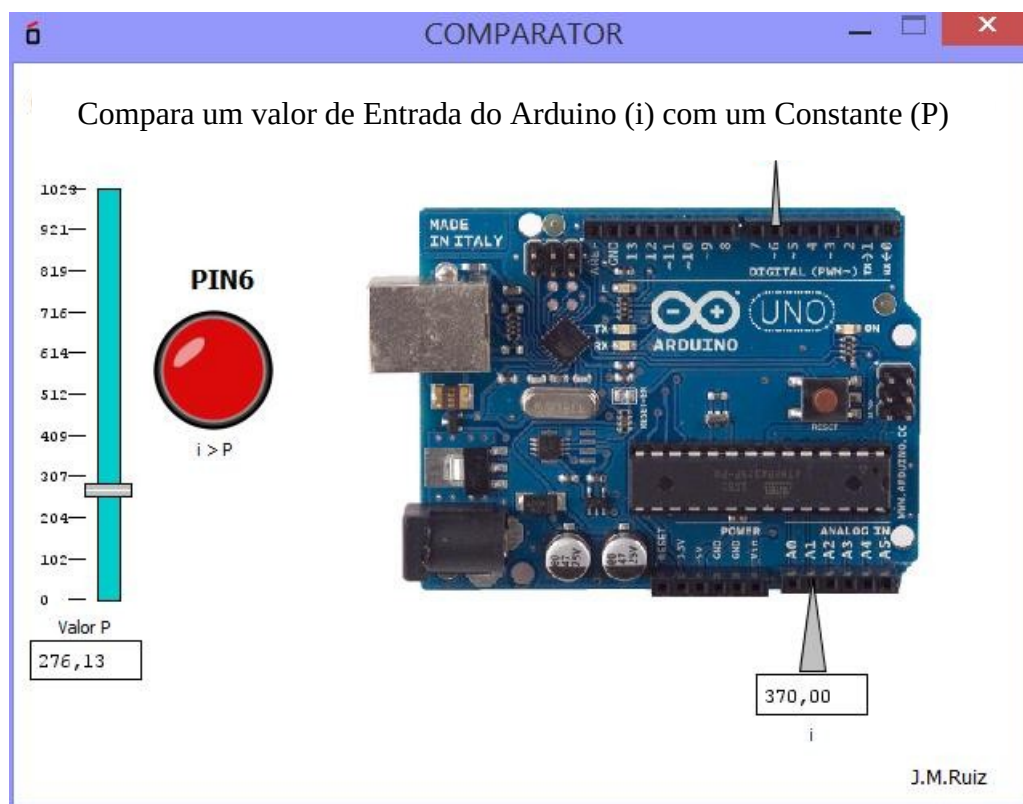


Diagrama funcional da aplicação conteúdo no Panel circuito de MyOpenLab

As variáveis que devemos definir na aplicação são mostradas na janela de definição de variáveis de MyOpenLab.



Na figura mostra-se o Painei Frontal da aplicação

