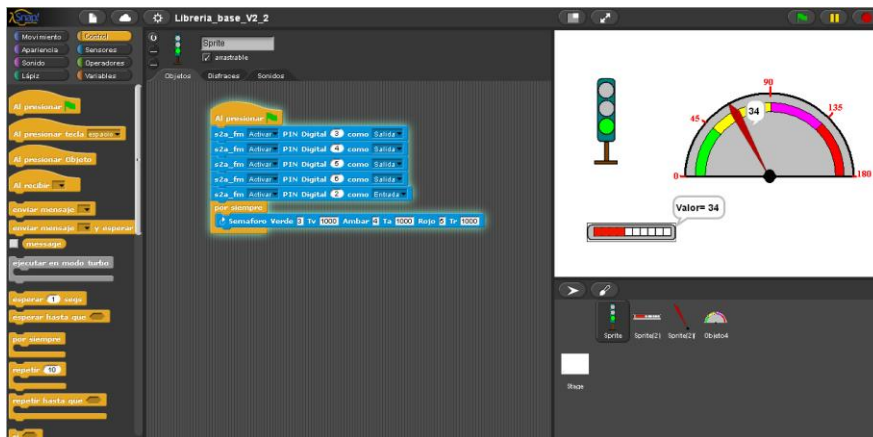


s2a_fm

ArduSnap! V 2.0

Ampliación de librerías para
Arduino+Snap!



Prof. José Manuel Ruiz Gutiérrez
Mayo 2014

Índice de bloques

Blink	 BLINK PIN Time(s)
Analog Regulator	 Analog Regulator A_Input Value Digital_Ouput
3State Regulator	 3State_Regulator PIN_A V1 V2 PINs P1 P2 P3
OR	 OR I1 I2 Out.
AND	 AND I1 I2 Out
NOT	 NOT I O
CLOCK	 CLOCK Out. Time_HIGH(ms) Time_LOW(ms)
Monoestable	 Monostable PIN Time(ms)
Biestable	 Biestable State_Inic Input_PIN Ouput_PIN
Mover Horizontal	 Mover PIN_Analog. Pos_Y Vmax Escala
Mover Vertical	 Mover Pin_Analog Pos_X Vmax Escala
Girar	 Girar Amax Vmx
Escalar Ent. Analógica	 Escalar Entrada Analógica PIN Vmax
Indicador Digital	 Indicador_digital 
Vumeter	 Vumeter Analog_PIN Tipo H Vmax
Secuencia Servo	 Secuencia Servo PIN T Secuencia 
Contador	 Contador PIN Reset PIN
Semáforo	 Semaforo Verde Tv Ambar Ta Rojo Tr
Slider Analog PWM	 Slider Pin_PWM Posición Vmax Incr. x
Key Control	 KEY_Control KEY Out_PIN
Biestable KEY	 Biestable_KEY State_Inic KEY Out_PIN
Secuenciador(1) 1 salida	 Secuenciador(1) Time s1 lista PIN_Out
Secuenciador(2) 2 salida	 Secuenciador(2) Time s1 s2 A B
Secuenciador(3) 3 salida	 Secuenciador(3) Time (ms) s1 s2 s3 A B C
Secuenciador(4) 4 salida	 Secuenciador(4) Time(ms) s1 s2 s3 s4 A B C D
Contador de 4 Bits	 Contador_Binario (4bits) Input Outputs A B C D
Trazador Gráfico	 Trazador_Grafico Analog_PIN Tmax(seg) Vmax Origen(X,Y)
Button Activation	 Button Activation PIN Color
Switch Activation	 Switch Activation PIN Color

Introducción:

El presente trabajo es continuidad del que he desarrollado con la traducción de las librerías [2sa_fm](#) creadas para [Snap!](#) y [Scratch 2.0](#) por [Alan Yorinks](#).

La aportación de Alan Yoeirik ha sido fundamental para que yo pueda realizar estas librerías, el ha ideado una librería escrita en Phyton a la que ha bautizado con el nombre de Pymata y con un sencillo protocolo de “telegramas” en forma de strings envía y recibe valores a través de la librería Firmata previamente instalada en la tarjeta Arduino como firmware de comunicación a través del puerto serie USB.

Basándome en este trabajo he realizado estas nuevas librerías que tienen como objetivo facilitar la tarea de realizar trabajos con Snap! Y Arduino.

Mi objetivo es animar a la utilización de Snap! + Arduino, por eso he querido “bautizar” estas librerías con el nombre de “**2sa_fm ArduSnap!**”

Desde aquí mi felicitación a Alan por su trabajo

Introduction.

The present work is a continuation of that I developed with the translation of the [2sa_fm](#) libraries created for [Snap!](#) and [Scratch 2.0](#) for [Alan Yorinks](#).

Yoeirik Alan's contribution has been crucial for me to make these libraries, he has devised a written in Python library that has given the name of Pymata and a simple protocol of "frames" in the form of strings sent and received values through the Firmata library previously installed in the firmware Arduino card as communication through the USB Serial port.

Based on this work I have made these new libraries that aim to facilitate the task of working with Snap! + Arduino.

My goal is to encourage the use of Snap! + Arduino, so I wanted to "baptize" these libraries with the name "2sa_fm ArduSnap!".

From here my congratulations to Alan for his work.

Librerías s2a_fm ArduSnap! V 2.0

Modo de trabajo.

- Las librerías invocan entradas o salidas de Arduino que previamente deben ser configuradas fuera de la librería.
- Las librerías que incluyen un Bucle deben ser ejecutadas fuera de un programa secuencial directamente activadas ejecutándose en un “hilo” a parte.
- La designación de Entradas/Salidas y los parámetros de las librerías deben ser colocados antes de ejecutarse la librería.
- Si una entrada o salida se utiliza en una librería no puede ser utilizada en otra porque pueden existir interferencias.
- Los bloques que tienen el símbolo  significa que incluyen un bloque de repetición , por lo tanto cuando se ejecutan por primera vez se quedan permanentemente ejecutándose

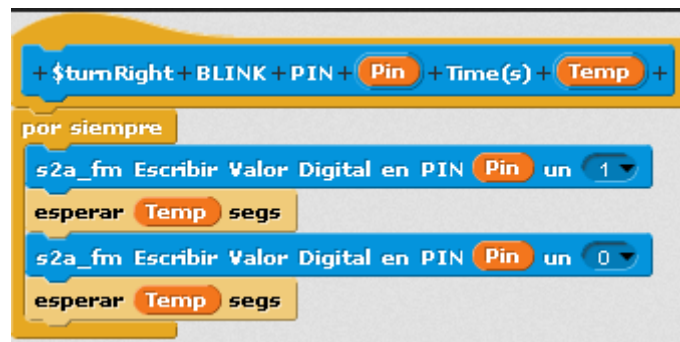
BLINK

Esta librería activa y desactiva una salida con intervalos de tiempo en ms.



Parámetros:

PIN: Numero de PIN de salida Digital
Time(s): Tiempo encendido y apagado en ms



Script del Bloque

ANALOG REGULATOR

Este bloque realiza la activación o desactivación de una salida digital en función de un valor de entrada analógico.



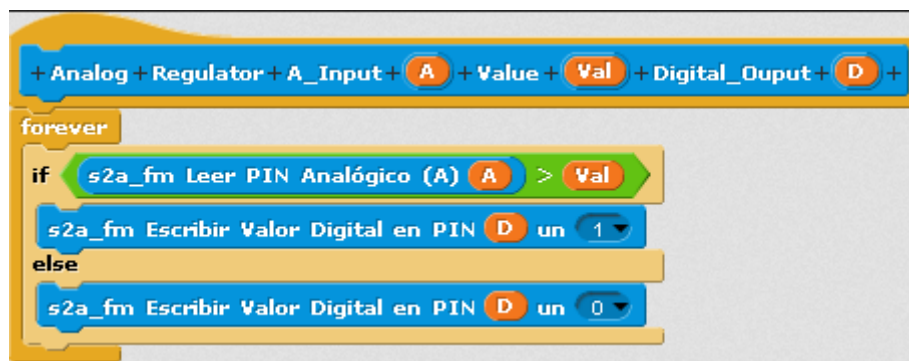
Parámetros:

A_Input: N° de Canal de Entrada Analogico de Arduino (0..5)
Value: Valor de consigna
Digital_Output: PIN de salida Digital de Arduino (0 .. 13)

Funncionamiento:

Si $A_input < 500$ then $Digital_Output = 0$

Si $A_input > 500$ then $Digital_Output = 1$



Script del Bloque

3STATE REGULATOR

Este bloque lee una señal analógica de entrada y en función de sus valores activa tres salidas digitales. Se trata de saber si una señal esta:

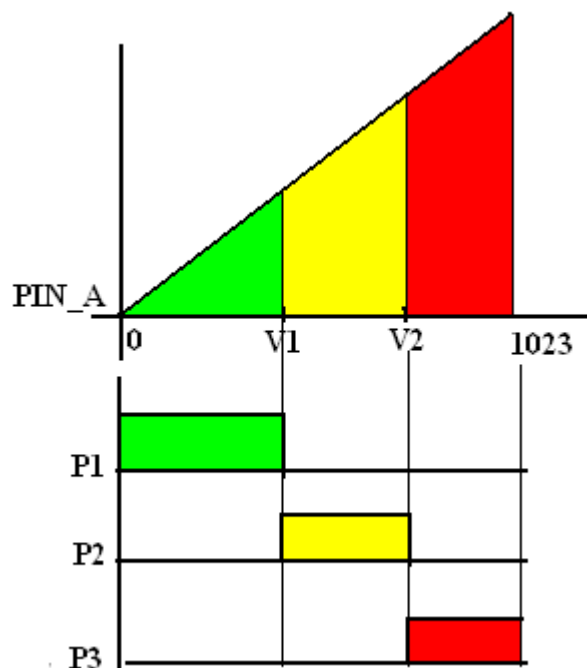
- Bajo un valor
- Dentro de un rango
- Por encima de un valor



Parámetros:

PIN_A: N° de canal analógico de entrada (0.. 5)
Val: V1 primer valor, V2 segundo valor del rango de control
P1: Salida1
P2: Salida2
P3: Salida3

Funcionamiento:



```
If PIN_A < V1 Then P1=1
Else P1=0
If PIN_A > V1 and PIN_A < V2 Then P1=1
Else P1=0
```

If PIN_A > V2 Then P3=1
 Else P3=0



Script del Bloque

OR

Esta función realiza la operación OR de dos entradas digitales de Arduino



Parámetros.

I1: Pin de Entrada I1
I2: Pin de Entrada I2
Out: Pin de Salida O

I1	I2	O
0	0	0
1	0	1
0	1	1
0	0	1



Script del Bloque

AND

Esta función realiza la operación AND de dos entradas digitales de Arduino



Parámetros.

I1: Pin de Entrada I1
I2: Pin de Entrada I2
Out: Pin de Salida O

I1	I2	O
0	0	0
1	0	0
0	1	0
0	0	1



Script del Bloque

NOT

Esta función realiza la operación AND de dos entradas digitales de Arduino



Parámetros.

I: Pin de Entrada I
O: Pin de Salida O

I	O
0	1
1	0



Script del Bloque

CLOCK

Esta función Genera impulsos de reloj de anchura configurable

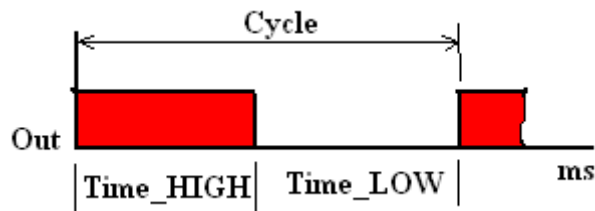


Parámetros:

Out: Pin digital de salida

Time_HIGH: Tiempo de duración en nivel “1”

Time:LOW: Tiempo de duración en nivel “0”



Script del Bloque

MONOESTABLE

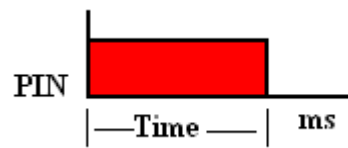
Este bloque representa un elemento monoestable.



Parámetros

PIN: Pin de salida
Time(ms): Tiempo de activación de la salida

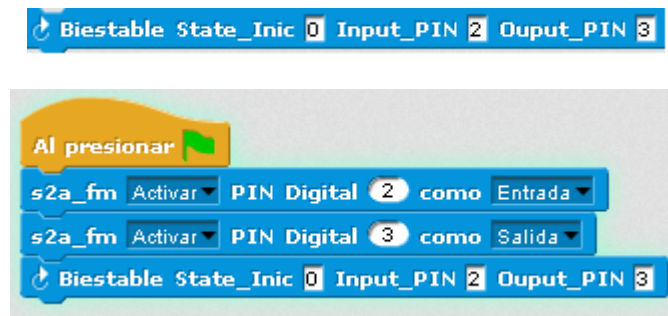
Cuando se ejecuta este bloque la salida PIN se activa durante un tiempo.



Script del Bloque

BIESTABLE

Este bloque representa un biestable



Parámetros:

- State_Inic: Estado inicial en el momento de lazar la función
- Input_PIN: Pin de entrada digital de entrada de gobierno del Biestable
- Output_PIN: Pin de salida digital del Biestable.



Script del Bloque

Mover Horizontal

Este bloque realiza el desplazamiento de un disfraz del objeto en el que se ejecute en sentido horizontal



Parámetros:

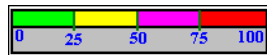
- PIN_Analog.:** Pin analógico de Entrada (0 a 5)
Pos_Y: Posición Y del cursor móvil
Vmax: Valor máximo de la medida que se corresponderá con 1023
Escala: Escala en el recorrido en pasos por cada unidad.



Script del Bloque

Ejemplo:

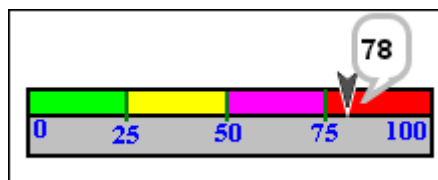
En el dibujo se ha colocado un Sprite que incluye un disfraz que representa una escala graduada, esta escala esta de fondo.



El Sprite en el que se ha incluido el bloque Mover Horizontal muestra el disfraz



“puntero” junto al que aparecerá el valor numérico de la posición.



El siguiente programa es un sencillo ejemplo en el que la lectura del PIN Analógico 0 configurado en un rango de 0 a100 y con Escala 1 mueve el cursor a lo largo de la regla graduada.



Mover Vertical

Este bloque realiza el desplazamiento de un disfraz del objeto en el que se ejecute en sentido vertical

↑ Mover Pin_Analog 0 Pos_X 0 Vmax 100 Escala 1

Parámetros:

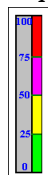
- PIN_Analog.:** Pin analógico de Entrada (0 a 5)
Pos_X: Posición X del cursor móvil
Vmax: Valor máximo de la medida que se corresponderá con 1023
Escala: Escala en el recorrido en pasos por cada unidad.



Script del Bloque

Ejemplo:

En el dibujo se ha colocado un Sprite que incluye un disfraz que representa una escala

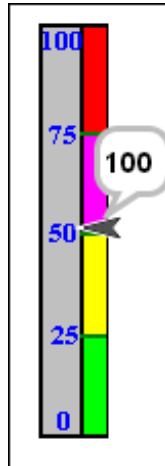


graduada, esta escala esta de fondo.

El Sprite en el que se ha incluido el bloque Mover Horizontal muestra el disfraz

“puntero” junto al que aparecerá el valor numérico de la posición.





El siguiente programa es un sencillo ejemplo en el que la lectura del PIN Analógico 0 configurado en un rango de 0 a100 y con Escala 1 mueve el cursor a lo largo de la regla graduada.



GIRAR

Este bloque realiza el giro de un disfraz del Sprite en el que se ejecute.

Girar 0 Amax 180 Vmx 1000

Parámetros:

Girar: Pin analógico de Entrada (0 a 5)
Amax Valor máximo en grados a girar
Vmax: Valor máximo de la medida que se corresponderá con 1023

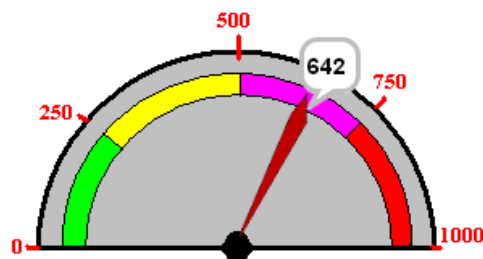
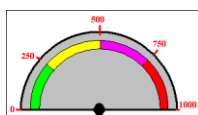


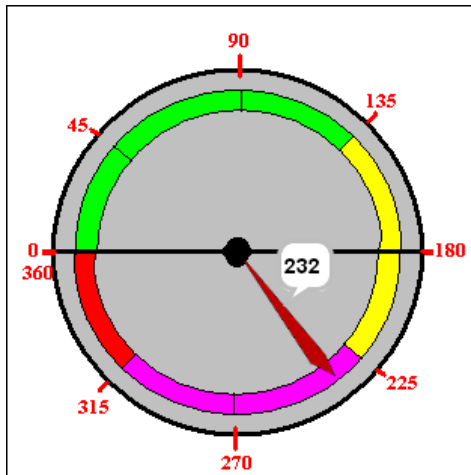
Script de Bloque

Ejemplos:

La imagen siguiente muestra el efecto del cueros giratorio (aguja) sobre un fondo que es una imagen de una escala semicircular en la que se han rotulado los valores de la variable numeric 0

a 1000 . El fondo es un Sprite que incluye como disfraz la escala semicircular.





Girar 0 Amax 360 Vmx 360



Escalar Entrada Analógica

Esta función lee un canal analógico de entrada y devuelve el valor leído escalado a un Valor máximo Vmax.

Los valores leídos por defecto de la tarjeta sabemos que son de **0 a 1023** pues en este caso los valores variarían de **0 a Vmax**. 

Parámetros:

PIN:	Pin digital (0 a 13)
Vmax:	Valor máximo de la medida que se corresponderá con 1023

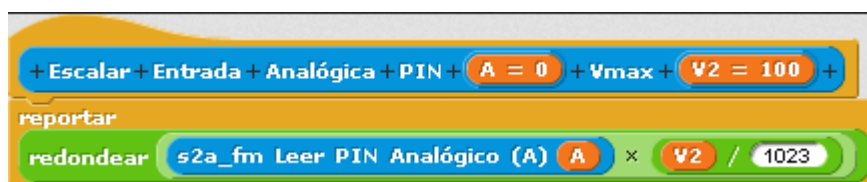


El valor leído lo muestra junto al display que tengamos seleccionado.

Vemos un sencillo ejemplo de esta instrucción. La función no se ejecuta de modo cíclico, lo cual significa que para ejecutarla permanentemente debemos meterla dentro de un bucle “por siempre”.



Este es el script del contenido del bloque



INDICADOR DIGITAL

Con este bloque podemos mostrar en la ventana de ejecución un “led” con dos estado: encendido y apagado que se corresponderan con el estado del PIN digital que leamos

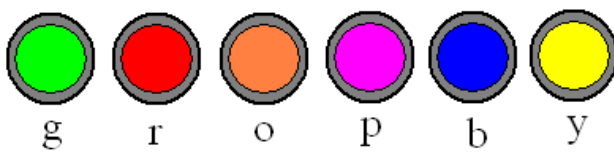


Parámetros:

Indicador_digital: Pin digital (0 a 13)



: Color del Led (g,r,o,p,b,y) El color por defecto es el rojo (r)



Un ejemplo sencillo de utilización es el siguiente.

Vemos que primero configuramos el PIN 2 como Entrada y después colocamos el bloque dentro de un bucle “por siempre”



El script del bloque es el mostrada en la figura siguiente



VUMETER

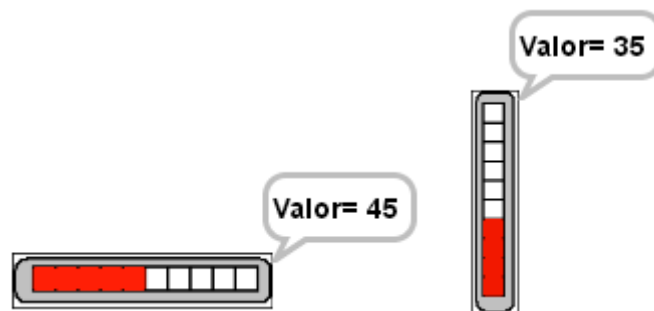
Este bloque de función permite mostrar en la pantalla un “vumeter” de color rojo que se activa en función del valor leído



Parámetros:

PIN:	Pin digital (0 a 13)
Tipo:	Posición vertical u horizontal (H horizontal por defecto)
Vmax:	Valor máximo de la medida que se corresponderá con 1023

En la siguiente figura vemos los vumeters en las posiciones posibles.



El siguiente esquema muestra un ejemplo basico de utilización.

Se configura el PIN A(0) como entrada que leeremos y despues, dentro de un bloque “por siempre” colocamos el bloque del Vumeter.



Omito el script de este bloque porque es demasiad extenso.

Secuencia Servo

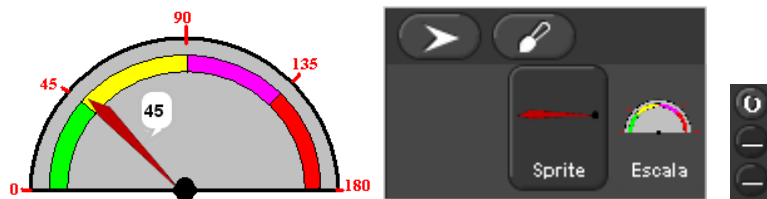
Este bloque sirve para manejar un servo conectado a una de los pines digitales de arduino.

Secuencia Servo PIN 0 T 1 Secuencia

Parámetros:

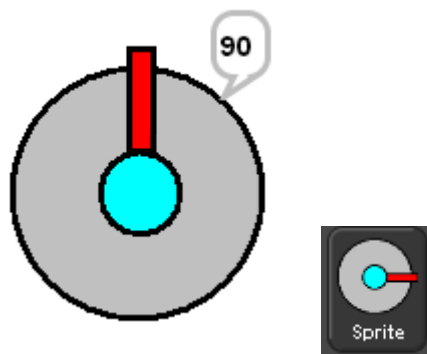
PIN: Pin digital (0 a 13)
T: Intervalo de tiempo entre cada valor de ángulo
Secuencia: Lista de valores de posiciones que deseamos ejecutar

Si deseamos monitorizar los valores de las posiciones de giro del servo (0 a 180°) podemos hacerlo añadiendo un objeto con el disfraz de la escala y seleccionando en el Objeto en donde tenemos el bloque el disfraz de “aguja”



No olvidemos seleccionar la opción de giro en el disfraz de la aguja del objeto Sprite para que esta pueda girar.

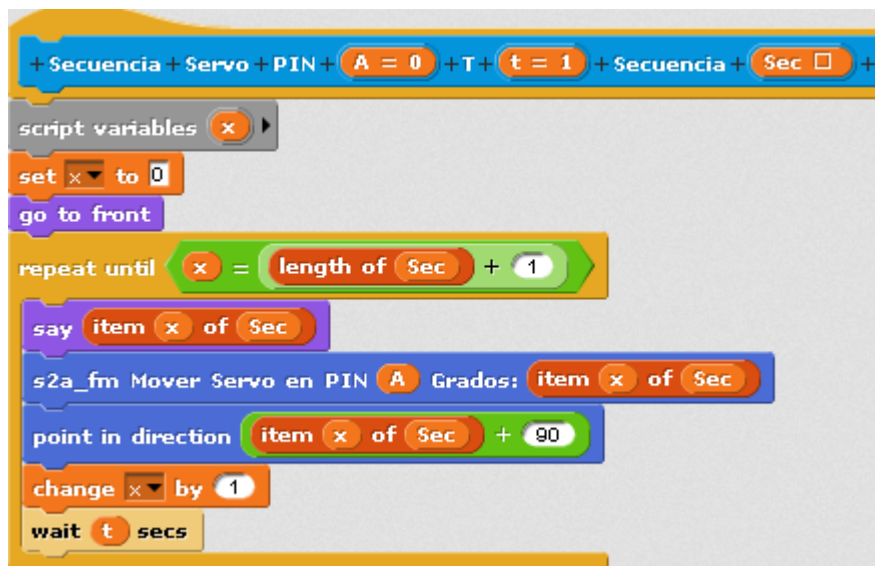
Podríamos utilizar solamente un objeto si en el objeto “Escala” seleccionamos el disfraz “servo” y seleccionando también la opción giro del disfraz como en el caso anterior



El programa sería el siguiente.



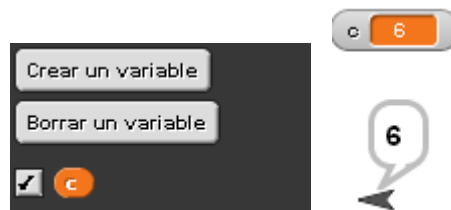
El contenido del bloque seria el siguiente script. Observamos que es un objeto que se ejecuta con una secuencia que recorre cada uno de los valores de las posiciones que colocamos en en la lista de parámatros.



Contador

En ocasiones tenemos necesidad de contar eventos. Para esto hemos creado el bloque Contador que recoge las pulsaciones que generamos en un PIN digital en modo entrada y las contamos y entregamos a una variable que previamente se debiera crear de nombre “c”. Si no se crea esta variable el contador no funciona y da un error

Contador PIN 2 Reset PIN 3



Creación de la variable c

Parámetros:

Contador PIN: Pin digital (0 a 13) para la entrada de impulsos
Reset PINT: Pin digital (0 a 13) de puesta a cero del contador

El siguientes es un sencillo ejemplo de su uso. Tengase en cuenta que el bloque no es ciclico por lo tanto debe insertarse dentro de un bloque “por siempre”



El contenido del bloque es el de la figura.

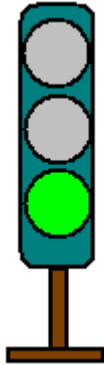


Semáforo

Este bloque implementa un semáforo elemental



Parámetros:



Verde:	PIN asociado a la lámpara verde
Tv:	Tiempo de encendido de la lámpara verde
Ambar:	PIN asociado a la lámpara ámbar
Ta:	Tiempo de encendido de la lámpara ámbar
Rojo:	PIN asociado a la lámpara roja
Tr:	Tiempo de encendido de la lámpara roja

El siguiente ejemplo muestra como implementar el semáforo. Se definen los pines de salida PIN3, PIN4 y PIN5 para cada lampara y despues se coloca el bloque Semáforo que al ser de tipo ciclioc sempore se estar ejecutando (no será necesario colocar un bloque “por siempre”



Se omite el contenido del bloque por ser demasiado extenso y muy sencillo de interpretar.

Slider Analog_PWM

Esta función permite, mediante el uso de las teclas de cursor-arriba y cursor-abajo realizar el incremento o decremento de un valor que se asocia a un canal analógico de salida PWM pudiendo hacer variable una señal analógica de salida



Parámetros:

Pin_PWM:	PIN asociado a la salida de valor analogico (3,5,6,9,10 y11)
V:	Valor maximo de la variable
Incr.x:	Valor del incemento por cada pulsacion de tecla

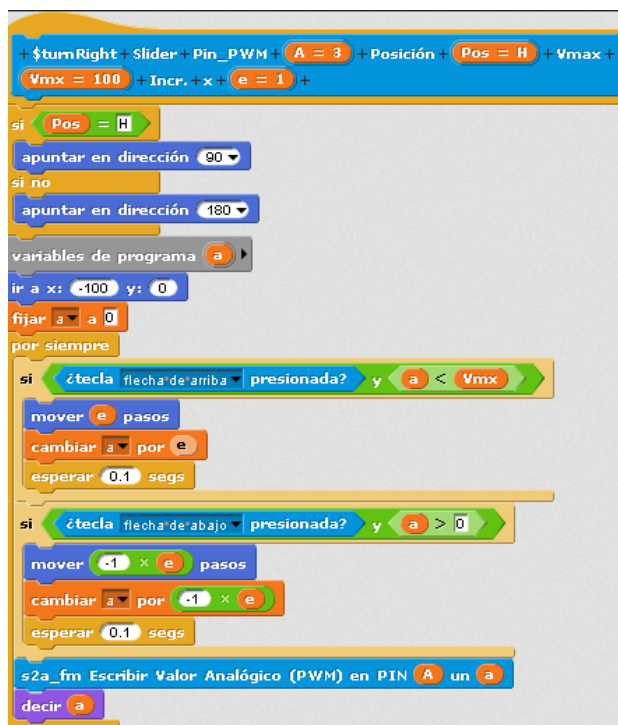
El valor se muestra en el disfraz activo del Objeto en el que utilizemos la función.



Este es un sencillo ejemplo de uso de la función.



El siguientes es el contenido del script del bloque.



KEY CONTROL

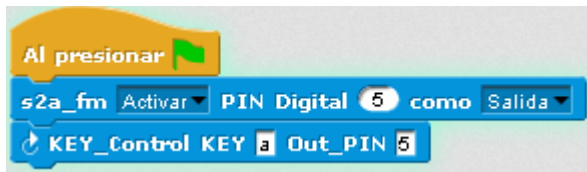
Esta función permite el gobierno de una salida digital mediante la pulsación de una tecla del teclado. Al pulsar se activara la salida y al soltar se desactivara.



Parámetros:

Key: Tecla a pulsar para e control de la salida (por defecto será a)
Out_PIN: Pin de la salida que queremos gobernar.

Ejemplo de utilización. Obsérvese que el bucle es de tipo cíclico ya que incluye un componente “por siempre”



Este es el contenido del script del bloque



BIESTABLE KEY

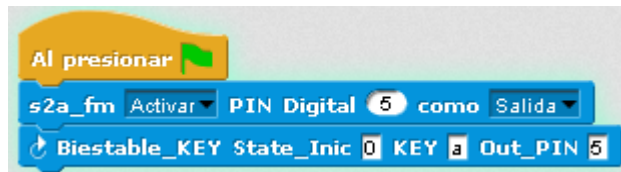
Este bloque es muy parecido al anterior. En este caso la función del control, desde una tecla del teclado es de tipo biestable (memoria). Pulsamos y se activa , volvemos a pulsar y se desactiva.



Parámetros:

- State_Inc.:** Estado inicial de la salida digital
- Key:** Tecla a pulsar para el control de la salida (por defecto será a)
- Out_PIN:** Pin de la salida que queremos gobernar.

Ejemplo de utilización. Obsérvese que el bucle es de tipo cíclico ya que incluye un componente “por siempre”



Este es el contenido del script del bloque



SECUENCIADOR(1) 1 SALIDA

Este bloque permite generar una secuencia de activaciones/desactivaciones de una salida que se recoge en una lista

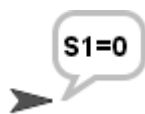


Parámetros:

Time:	Tiempo entre valores de la secuencia
S1:	Lista de valores de activación (0 desactiva, 1 activa)
PIN_Out:	Pin de salida

Lista de la secuencia

El programa básico para la prueba de este bloque es el siguiente.



Los valores de salida se muestran en el disfraz por defecto.

El contenido del bloque es el siguiente



SECUENCIADOR(2) 2 SALIDAS

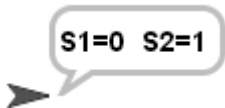
Este bloque permite generar una secuencia de activaciones/desactivaciones de dos salidas que se recoge en dos listas S1 y S2.



Parámetros:

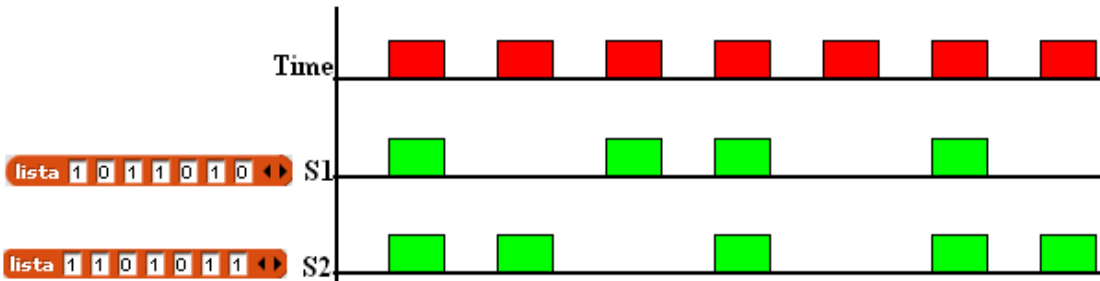
- Time:** Tiempo entre valores de la secuencia
- S1:** Lista de valores de activación (0 desactiva, 1 activa)
- S2:** Lista de valores de activación (0 desactiva, 1 activa)
- A:** Pin de salida A
- B:** Pin de salida B

El programa básico para la prueba de este bloque es el siguiente.



Los valores se muestran en la pantalla de la forma indicada.

En el cronograma siguiente se muestra la secuencia de valores que salen por las salidas S1 S2 correspondientes a los pines PIN4 y PIN5



Este es el contenido del script de bloque.



SECUENCIADOR(3) 3 SALIDAS

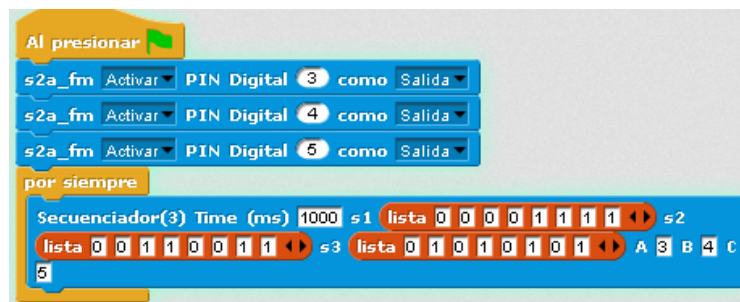
Este bloque permite generar una secuencia de activaciones/desactivaciones de tres salidas que se recoge en tres listas S1, S2 y S3



Parámetros:

Time:	Tiempo entre valores de la secuencia
S1:	Lista de valores de activación (0 desactiva, 1 activa)
S2:	Lista de valores de activación (0 desactiva, 1 activa)
S3:	Lista de valores de activación (0 desactiva, 1 activa)
A:	Pin de salida A
B:	Pin de salida B
C:	Pin de salida C

El programa básico para la prueba de este bloque es el siguiente.



S1=1 S2=0 S3=1

Esta es la información que se muestra en la pantalla de los valores que van saliendo por las salidas.

El contenido del script del bloque es el siguiente.



SECUENCIADOR(4) 4 SALIDAS

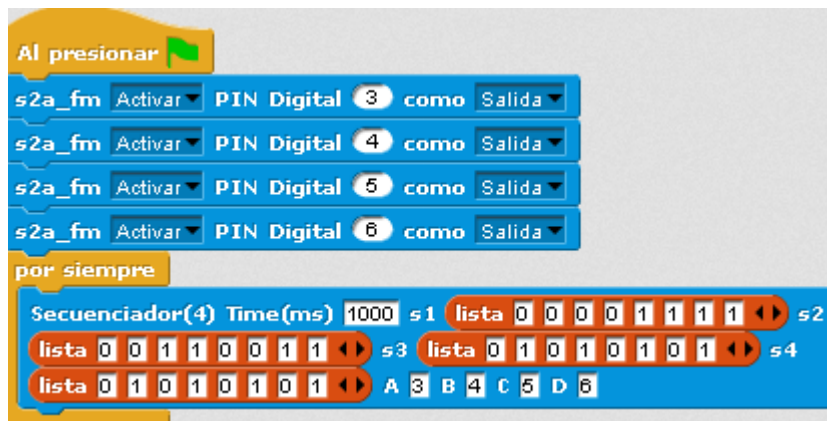
Este bloque permite generar una secuencia de activaciones/desactivaciones de cuatro salidas que se recoge en tres listas S1, S2, S3 y S4



Parámetros:

Time:	Tiempo entre valores de la secuencia
S1:	Lista de valores de activación (0 desactiva, 1 activa)
S2:	Lista de valores de activación (0 desactiva, 1 activa)
S3:	Lista de valores de activación (0 desactiva, 1 activa)
S4:	Lista de valores de activación (0 desactiva, 1 activa)
A:	Pin de salida A
B:	Pin de salida B
C:	Pin de salida C
D:	Pin de salida D

El programa básico para la prueba de este bloque es el siguiente.



S1=1 S2=0 S3=0
S4= 0

Esta es la información que se muestra en la pantalla de los valores que van saliendo por las salidas.

El contenido del script del bloque es el siguiente.



CONTADOR BINARIO DE 4 BITS

Este bloque implementa un contador binario de 4 bits.



Parámetros:

Input: Pin de entrada de impulsos de cuenta

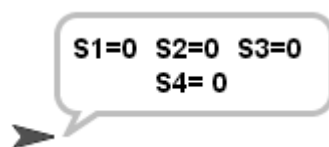
A: Salida A

B: Salida B

C: Salida B

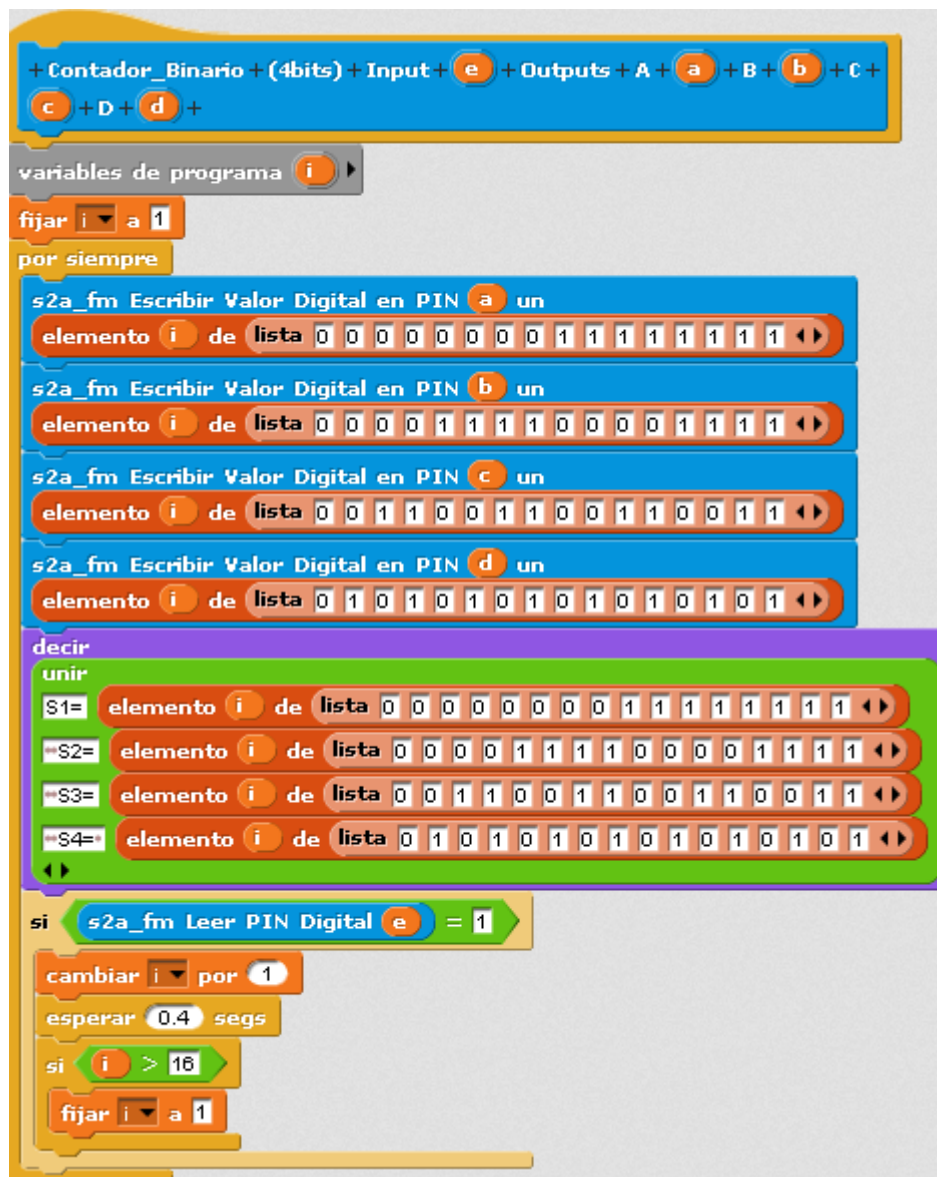
D: Salida C

Ejemplo básico de prueba.



Valores que se visualizan

El siguiente es el esquema del script de este bloque.



TRAZADOR GRAFICO

Este bloque realiza el trazado grafico de una señal analógica recogida de un canal de Arduino



Parámetros:

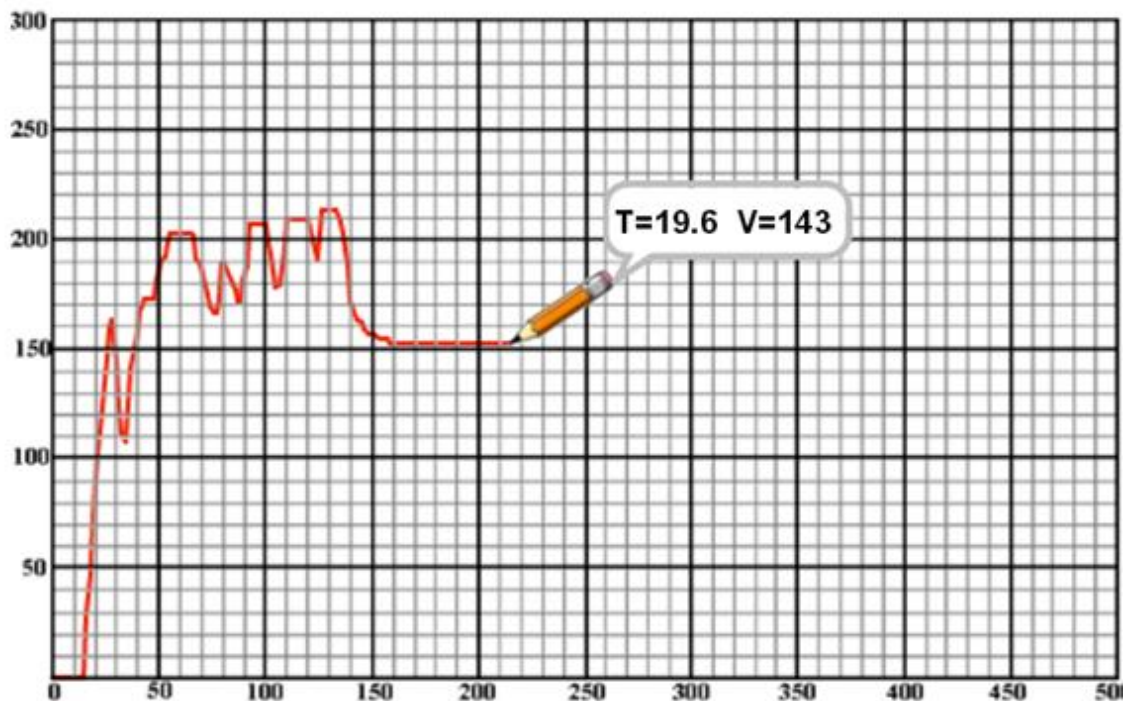
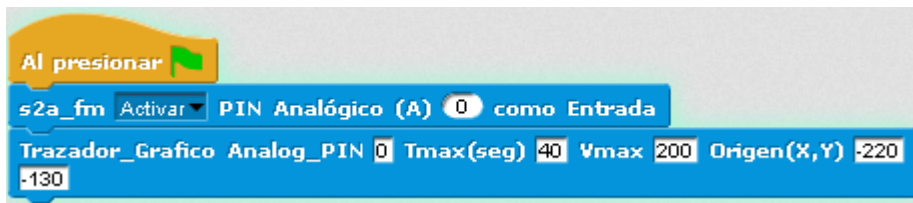
Analog_PIN: Pin de entrada analógica (0..5)

Tmax(seg): Tiempo máximo en la escala

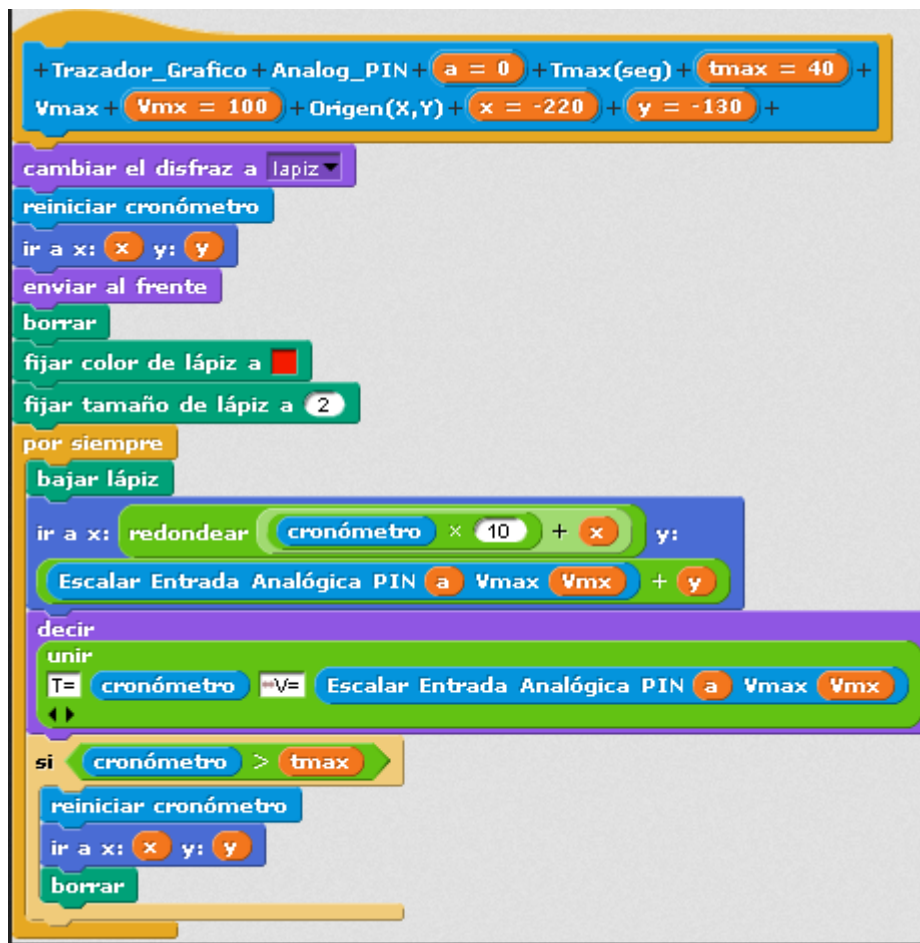
Vmax: Valor máximo en la escala

Origen(X,Y): Situación del origen de coordenadas en el grafico.

Ejemplo de programa básico.



Contenido del script del bloque.



BUTTON ACTIVATION



Este bloque permite activar o desactivar en modo “botón” o pulsador una de las salida que indiquemos. Al poner sobre el botón el cursor del ratón se activa la salida y al retirarlo se desactiva



Cursor del ratón fuera del botón Salida Desactivada



Cursor del ratón encima del botón. Salida Activada

Parámetros:

PIN: Numero de Pin de salida

Color: Color del botón al estar activa la salida (ON)

Colores de botón estado ON:

r: Rojo

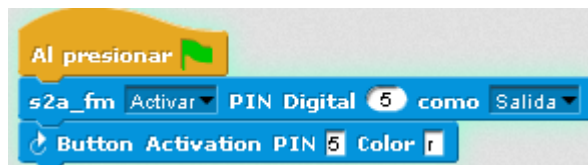
g: Verde

y: Amarillo

b: Azul



Este es un sencillo ejemplo de uso. Configuramos el PIN5 como salida y designamos en el bloque el PIN y el color. Por defecto trae el color rojo “r”



Podríamos poner tantos botones como queramos pero sabiendo que cada uno debe estar adscrito a un Objeto distinto en el que se colocará el correspondiente programa.

SWITCH ACTIVACION

Este bloque permite activar o desactivar en modo “Switch” o biestable una de las salidas que indiquemos. Al poner sobre el botón el cursor del ratón se activa la salida y al retirarlo se queda activado. Se debe retirar el cursor del ratón antes de un segundo de lo contrario, si lo dejamos sobre el botón, esta salida se pondrá a parpadear. Una vez activado si volvemos a acercar el ratón sobre el objeto se desactivará y quedará desactivado al retirarlo.



Salida desactivada



Salida Activada

Parámetros:

PIN: Numero de Pin de salida

Color: Color del botón al estar activa la salida (ON)

Colores de botón estado ON:

r: Rojo

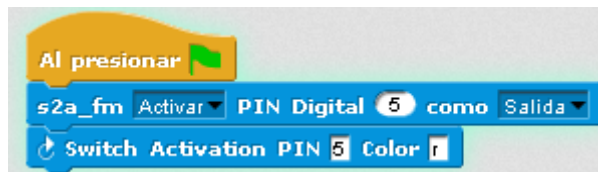
g: Verde

y: Amarillo

b: Azul



Este es un sencillo ejemplo de uso. Configuramos el PIN5 como salida y designamos en el bloque el PIN y el color. Por defecto trae el color rojo “r”



Podríamos poner tantos switches como queramos pero sabiendo que cada uno debe estar adscrito a un Objeto distinto en el que se colocará el correspondiente programa.