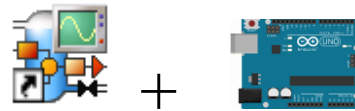
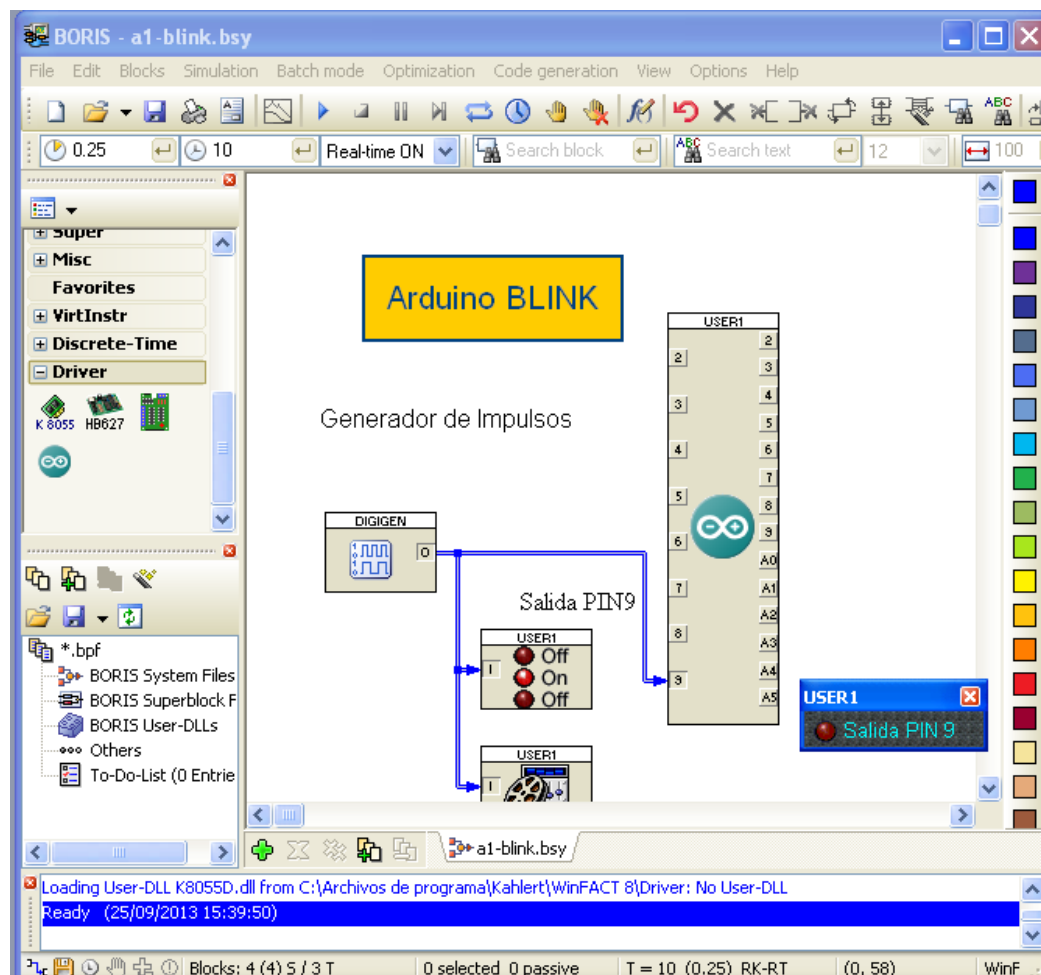


WinFACT + Arduino



Conexión y Diseño de Aplicaciones entre WinFACT y Arduino



Prof. José Manuel Ruiz Gutiérrez
Octubre 2013

INDICE

1. Introducción
2. La librería Arduino de WinFACT.
3. Configuración del objeto Arduino
4. Algunas cuestiones generales de WinFACT
5. Ejemplos prácticos.
 - 5.1. Test Arduino + WinFACT
 - 5.2. Control de una salida Digital
 - 5.3. Blink
 - 5.4. Función AND
 - 5.5. Contador de impulsos sencillo
 - 5.6. Semáforo Sencillo
 - 5.7. Semáforo en tiempo real
 - 5.8. Control de Iluminación de Escalera
 - 5.9. Salida PWM
 - 5.10. Encendido Progresivo de un LED
 - 5.11. Entrada analógica 1
 - 5.12. Entrada Analógica 2
 - 5.13. Climatización 1
 - 5.14. Climatización Banda Muerta

1. Introducción

La conexión de la tarjeta Arduino a distintas herramientas software para facilitar su manejo y potenciar sus posibilidades nos lleva a tomar en consideración la poderosa herramienta WinFACT bien conocida en los ambientes académicos y también profesionales.

WinFACT posee un conjunto de herramientas que facilitan el diseño de aplicaciones que permitirán la conexión con el mundo físico en el tratamiento de los datos y en el gobierno de actuadores y la lectura de sensores.

La forma de trabajo que presentamos en este tutorial es muy sencilla y solo requiere un mínimo conocimiento de conceptos de programación y un conocimiento básico del entorno WinFACT.

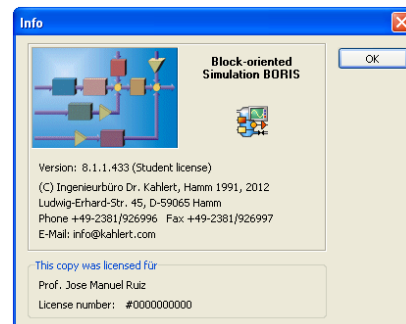


Las colecciones de librerías del software son muy amplias y podremos usarla en toda su gama, lo cual nos permitirá ir de lo sencillo a lo más complejo sin dificultad.

WinFACT incorpora un bloque de librería del tipo “USER” que se encarga de la comunicación con Arduino. Esta realizada en Delphi y se dispone del código fuente para poder realizar las modificaciones que nos parezcan bien.

La conexión se lleva a cabo depositando previamente un “firmware” en la tarjeta Arduino que se encargará del intercambio de información a través del puerto USB.

Se ha utilizado para este tutorial la versión **WinFACT 8**



<http://www.kahlert.com/web/home.php>

2. La librería Arduino de WinFACT.

A continuación se explican los ficheros que vienen incluidos en la librería:

El archivo ZIP **wfarduino.zip** que se puede descargar en la pagina de WinFACT www.kahlert.com contiene los siguientes componentes:

- El controlador Arduino para BORIS (WinFACT 6/7/8)
- El código fuente en DELPHI 6 del controlador
- El firmware Arduino para la comunicación a través de USB
- El firmware Arduino para la comunicación a través de Ethernet (UDP)

La manera de proceder para la instalación de a librería es la siguiente.

Descomprimir todos los archivos en la carpeta UserDLLs de la carpeta en dnde este instado WinFACT (normalmente **C: \ Archivos de programa \ Kahlert \ WinFACT8 \ userdlls**) Una vez realizado, cuando arranquemos la siguiente vez entonces usted puede utilizar el Drivers después próximo Inicio de BORIS en la paleta de objetos del usuario aplacerá el bloque de Arduino

La siguiente tabla ofrece una visión general de los archivos disponibles.

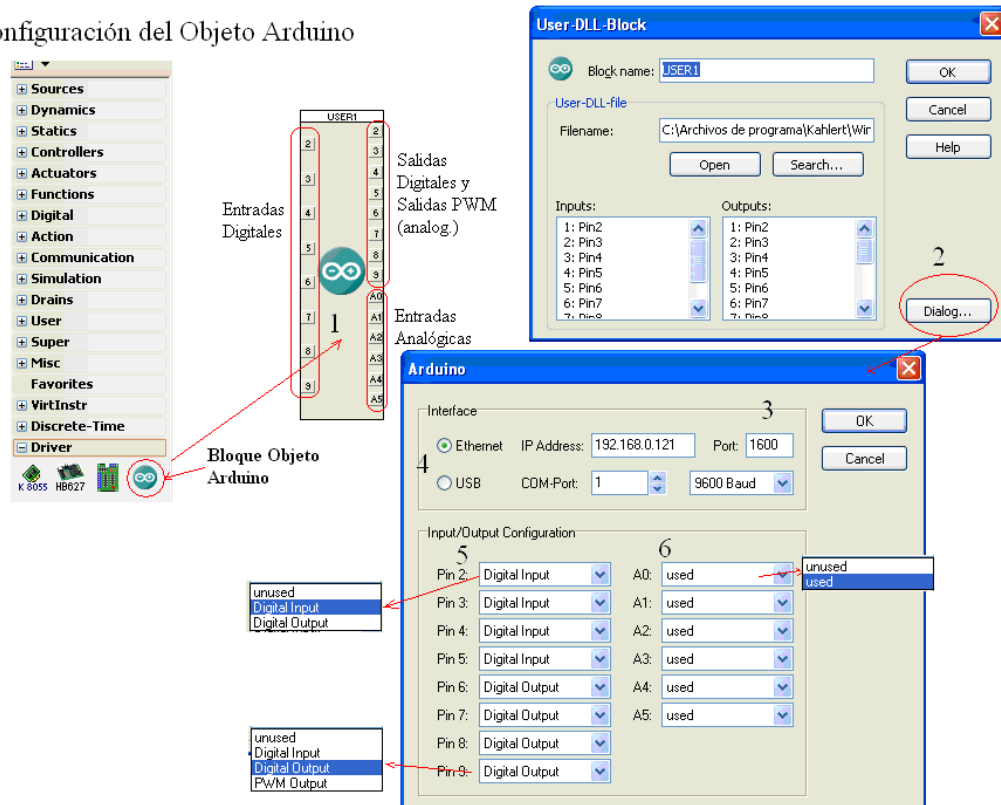
Descripción de archivos

<i>ArduinoTest.bsy</i>	Estructura de la prueba BORIS para Arduino (WinFACT 8!)
<i>Arduino.dll</i>	BORIS controlador para Arduino
<i>Arduino.bmp</i>	mapa de bits asociado
<i>Arduino_p.bmp</i>	mapa de bits asociado
<i>Arduino_t.bmp</i>	mapa de bits asociado
<i>Arduino.dpr</i>	DELPHI 6 archivo del controlador
<i>ArduinoDlg.ddp</i>	DELPHI 6 archivo del controlador
<i>ArduinoDlg.pas</i>	DELPHI 6 archivo del controlador
<i>ArduinoDlg.dfm</i>	DELPHI 6 archivo del controlador
<i>UDP.pde</i>	Arduino firmware (fuente) para la comunicación Ethernet
<i>SERIAL.pde</i>	Arduino firmware (fuente) para la comunicación USB

3. Configuración del objeto Arduino

Cuando colocamos el bloque (toma el valor por defecto USER1) **arduino** en el área de trabajo pulsando dos veces sobre el con el botón derecho del rato nos abra una ventana que permite la configuración del modo de trabajo de Arduino “**Dialog**”. Al pulsar este botón **Dialog** se abre la ventana (3) y en ella podemos configura:

Configuración del Objeto Arduino



Configuración de las Entradas/Salidas.

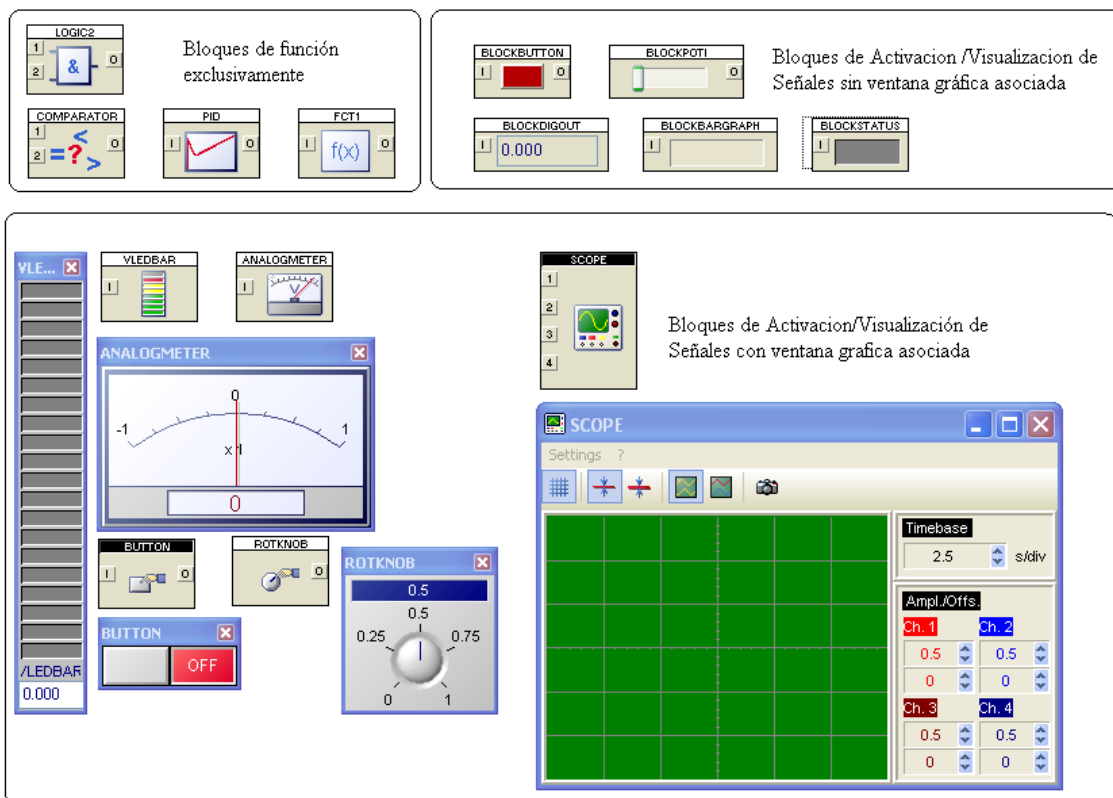
- El puerto de comunicación: Si nos comunicamos vía Ethernet o si nos comunicamos vía USB. Nosotros lo haremos vía USB. La configuración de la comunicación vía USB consiste en indicar el puerto por el que nos comunicaremos (podemos saberlo con la ayuda del IDE de Arduino) y la velocidad de comunicación (normalmente 9800 baudios).
- Seguidamente configuramos las entradas (inputs) y las salidas (outputs). Recordemos que en las entradas es donde colocaremos, por ejemplo los pulsadores y las salidas son en las que conectaremos los LEDs. Las entradas aparecerán en la parte derecha del bloque y las salidas en la parte izquierda.
- Cada Pin se configura pulsando sobre el y seleccionando el modo de trabajo. Recordemos que este firmware con el que trabajamos dispone de los pines 2,3,4,5,6,7,8,9 (8 pines) de los que son también salidas PWM los pines 3,5,6,9. Las entradas analógicas por defecto están en modo “used” (usadas) pero se pueden poner en modo “unused” (no usadas).

4. Algunas cuestiones generales de WinFACT

No es objetivo de este tutorial abordar el funcionamiento de la herramienta WinFACT. Este tema sería para dedicarle mucho más tiempo. Solo mencionaré algunas cuestiones muy básicas a tener en cuenta en relación con el entorno.

WinFACT tiene dos modos de trabajo: **Edición y Simulación**

En el **modo edición** colocamos los bloques de función que están incluidos en cada una de las librerías, los enlazamos y damos los parámetros correspondientes. Existen dos tipos de bloques: bloques que solo son de **función** (procesan datos de entrada y generan salidas) y bloques de **interacción y/o visualización**.



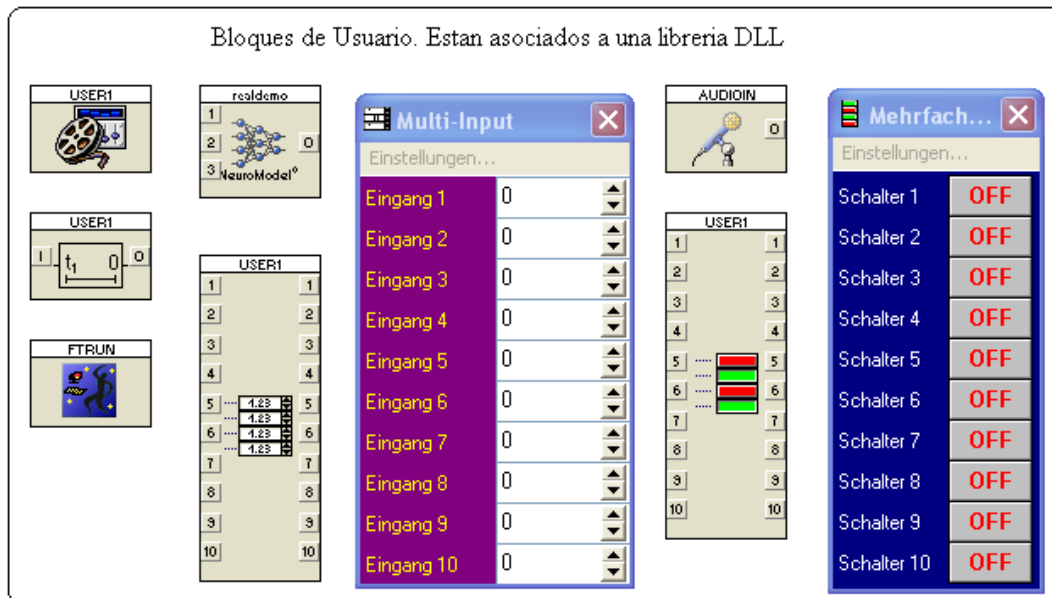
Los bloques unidos constituyen un modelo de algoritmo que es capaz de procesar datos de entrada y obtener salidas. Existen bloques que generan estímulos, es decir, señales de carácter continuo (analógicas) o discreto (digitales) que son las que se procesan y simulan el comportamiento de un sistema físico del tipo que sea.

El cableado se realiza de manera sencilla con la ayuda del ratón situando este sobre los terminales de entrada o de salida del bloque.

Hay bloques que llevan asociada una ventana o pantalla en la que aparecen elementos de tipo gráfico tanto para la generación como para la visualización de valores.

WinFact es un simulador de los llamados temporales en los que la variable principal es el tiempo.

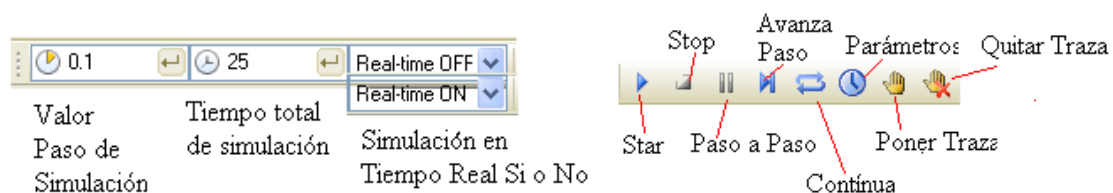
WinFACT tiene la posibilidad de incluirle librerías específicas que realizan operaciones concretas, esta librería se denominan de “usuario” y a continuación vemos algunas de ellas.



La simulación e un montaje se puede hacer atendiendo a dos modos fundamentales de trabajo: En Tiempo Real o en tiempo de máquina. El tiempo real significa que las unidades de tiempo serán mili segundos, segundos, minutos, etc.

La simulación se puede hacer en modo bucle (continuamente simulando) o acotarla en ciclos de simulación siendo cada unidad básica de tiempo lo que se denomina “paso de simulación”.

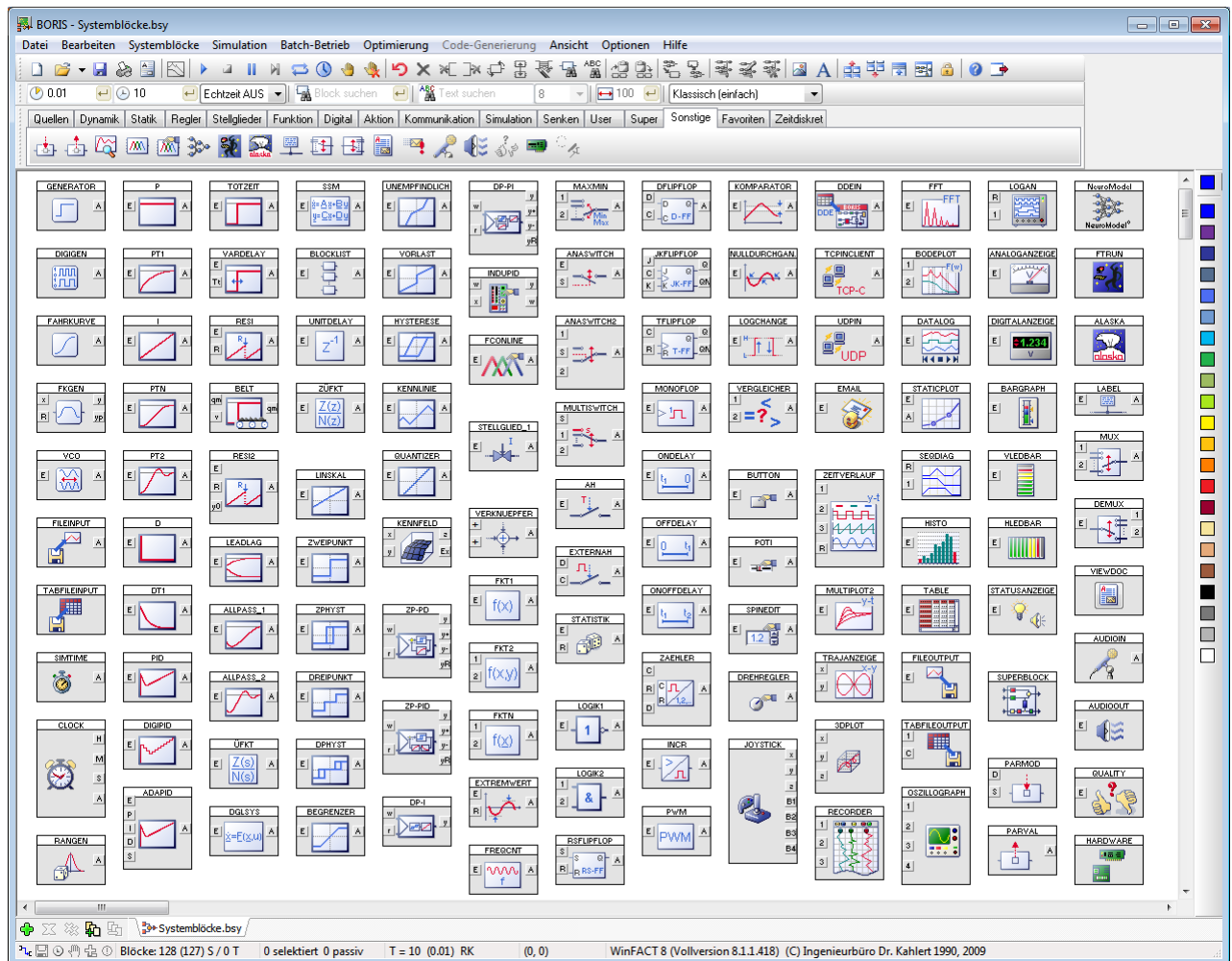
Gobierno de la Simulación



El gobierno de la simulación se realiza con una serie de botones de navegación que son los que mostramos en el dibujo anterior y que cada uno tiene un texto que define su función.

En a figura siguiente se muestra una pantalla con gran parte de las librerías de WinFACT.

Aspecto de la mayor parte de las librerías de WinFACT



Inclusión de sinópticos de animación. Herramienta FAB

A la hora de realizar nuestras simulaciones disponemos de una herramienta llamada **FAB** (*Flexible Animation Builder*) que permite crear una pantalla sinóptico con elementos que intercambian variables con el montaje de WinFACT



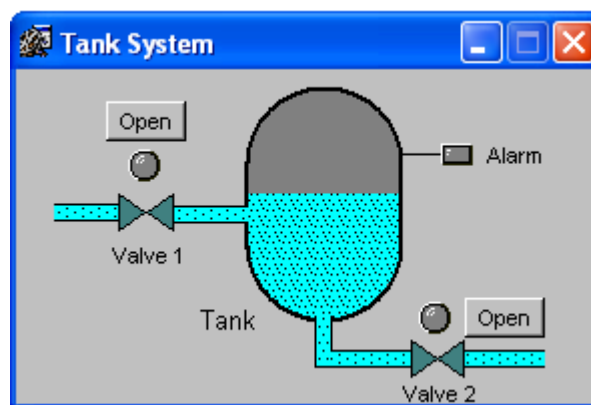
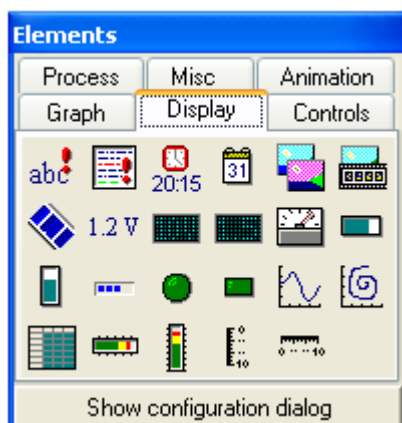
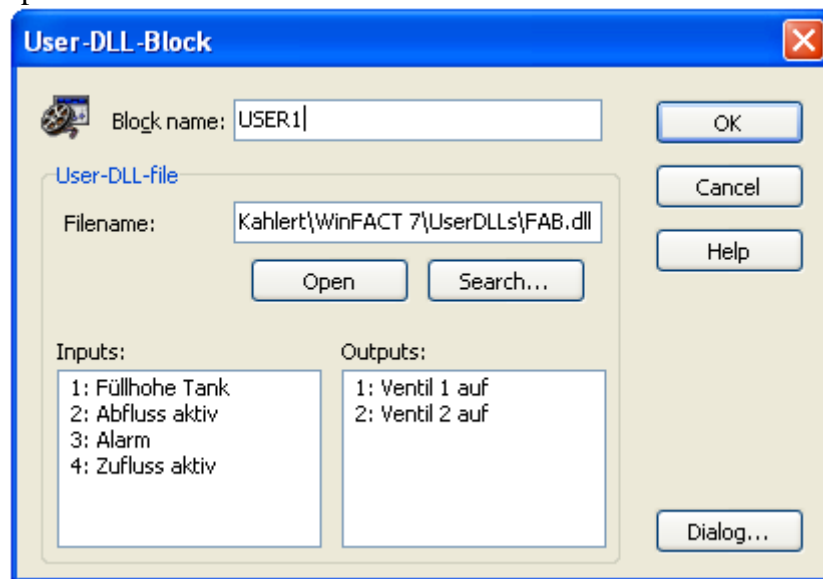
Para realizar una pantalla de interacción con FAB basta irse a la librería de bloques de

usuario y seleccionar un bloque de Usuario



en el que luego conectaremos

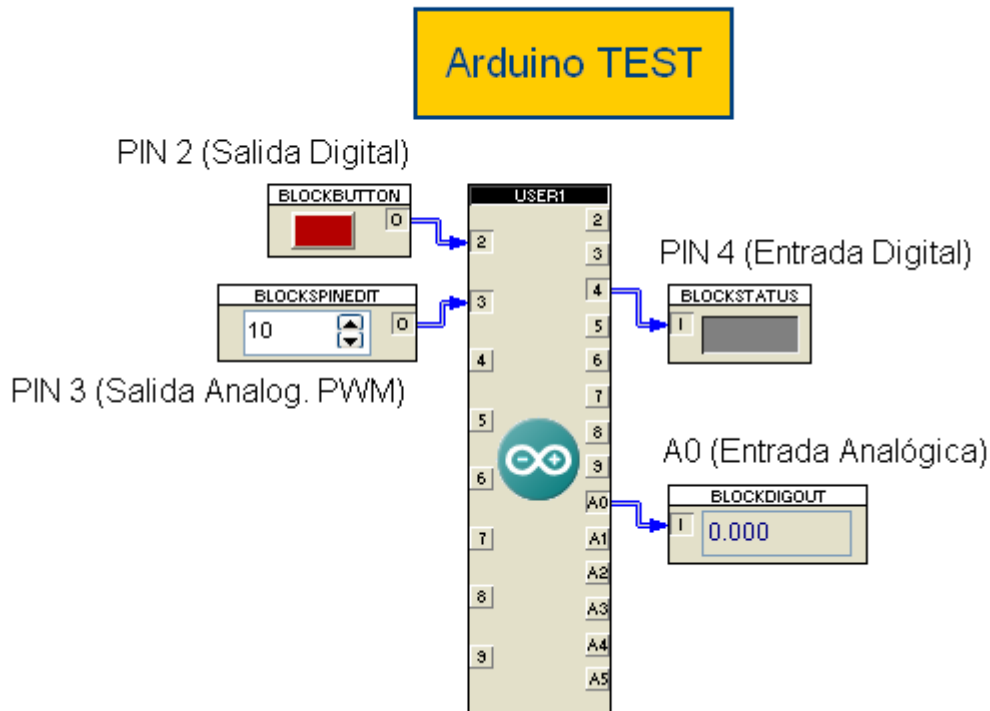
con la librería FAB.DLL que es la que invoca al entorno para realizar el diseño de la pantalla-sinóptico



5. Ejemplos prácticos.

5.1. Test Arduino + WinFACT

Empecemos por este sencillo ejemplo en el que reprobamos cada uno de los tipos de señales que pueden darse en la tarjeta Arduino: Salidas digitales, salidas analógicas, entradas digitales, entradas analógicas.

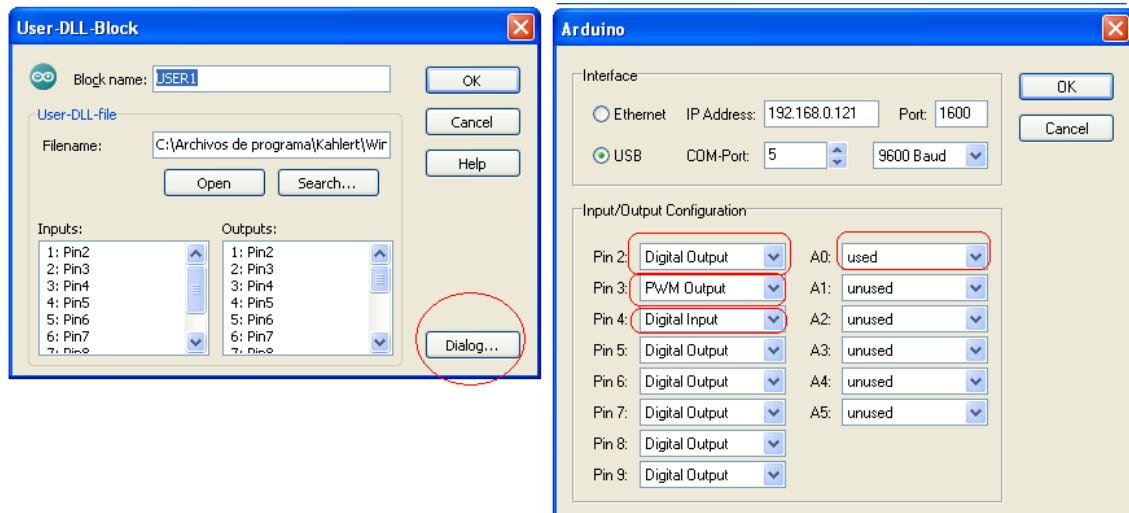




Se han colocado objetos de WinFACT para la generación de señal como un **BLOCKBUTTON** y un **BLOCKSPINEDIT** que introducen en Arduino un valor digital (0 o 1) y un valor analógico comprendido entre 0 y 255 para la salida **PWM**.

La salida entrada digital del **PIN4** se ha colocado un sencillo indicador de estado que es el bloque **BLOCKSTATUS** y para la entrada analógica se ha conectado una salida numérica que refleja el valor de la entrada analógica del canal A0 que es un valor comprendido entre 0 y 1024

No debemos olvidar configurar los bloques para los rangos de las señales que deben generar o leer.

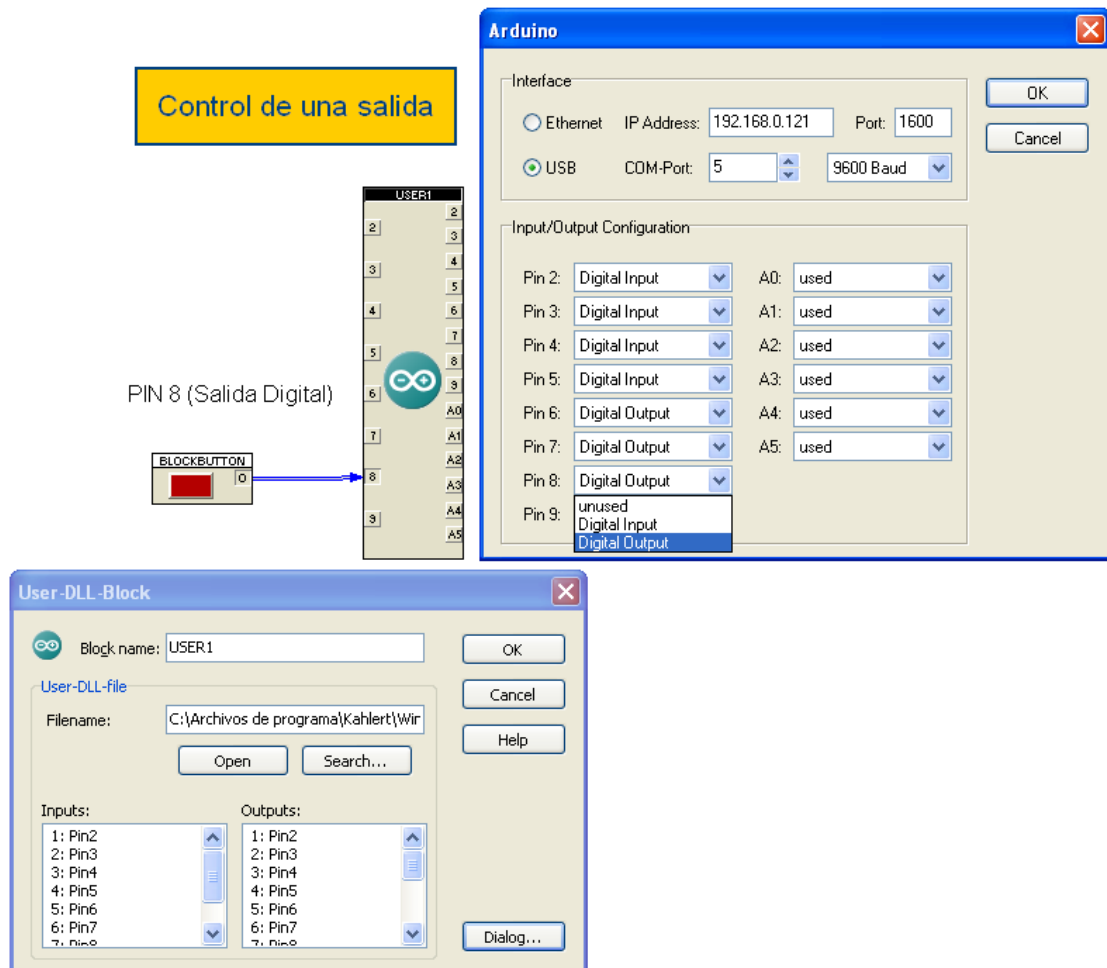
Otra parte fundamental de este ejercicio es realizar la configuración adecuada de la tarjeta Arduino, esto se hace pulsando dos veces sobre el bloque y aparecerá una ventana sobre a que seleccionaremos para nuestro caso la interface tipo USB, el puerto nº 5 y la velocidad de 9600 baudios. Seguidamente configuramos el modo de trabajo de los pines de la tarjeta: **PIN4 Digital Input**, **PIN2 Digital Output**, **PIN3 PWM Output** y **A0 Used**



Una vez realizada la configuración y el esquema con los bloques enlazados podremos realizar la simulación pulsando en el correspondiente botón pudiendo ser este para hacerla durante un tiempo establecido  o de manera continua .

5.2. Control de una salida Digital

Con este ejemplo realizaremos el gobierno de una salida digital, concretamente la salida correspondiente al PIN8.

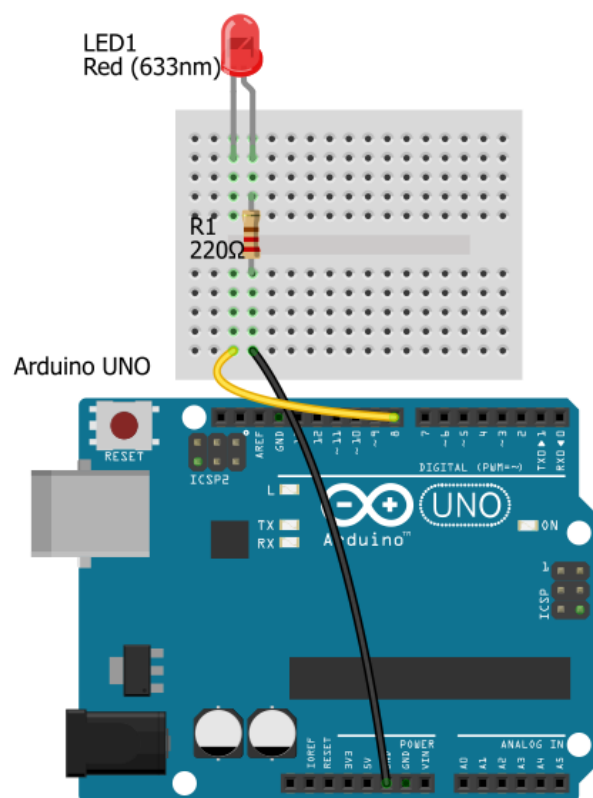


Una vez colocado el bloque de usuario Arduino en el área de trabajo pondremos los parámetros de funcionamiento y conexión. Por un lado designaremos como Interface el puerto **USB** (en nuestro caso será el 5 porque es al que conectamos Arduino) la velocidad la dejamos como está a **9600 baudios**.

En segundo lugar deberemos seleccionar el PIN8 y seleccionar que trabajará en modo salida (Digital Output).

Para gobernar el estado de la salida hacemos uso de un Bloque Botón (BLOCKBUTTON) y eso sería todo

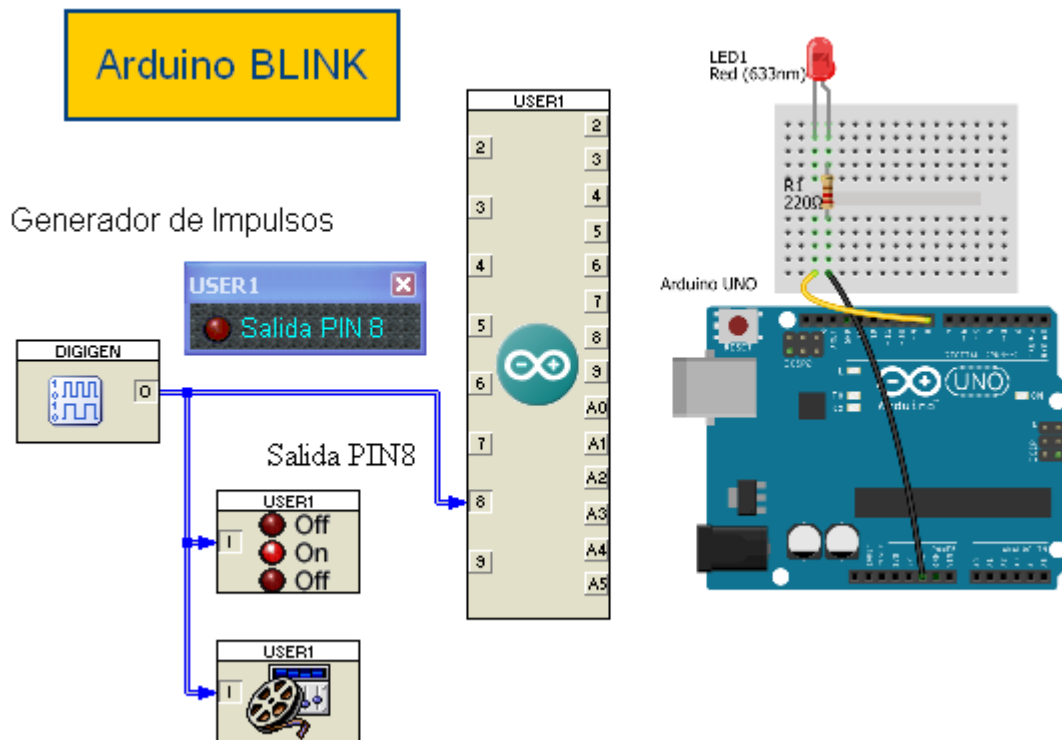
A continuación vemos el esquema de montaje de este ejemplo sobre una tarjeta protoboard.



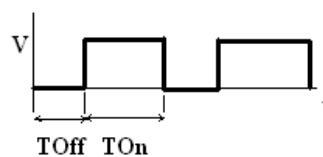
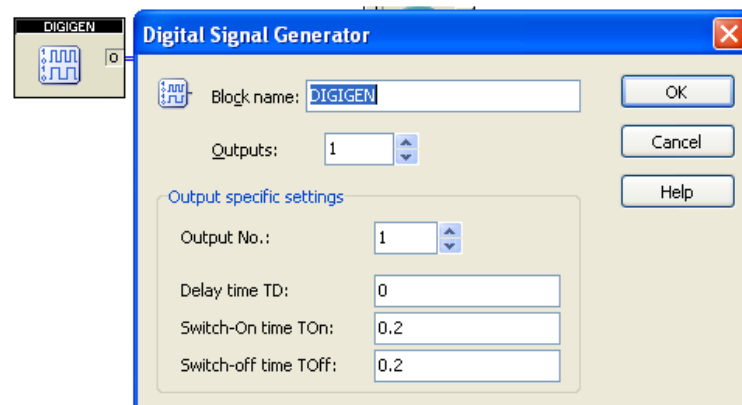
Montaje practico

5.3. Blink.

Si deseamos hacer que una salida se active y desactive de manera secuencial (intermitente) bastará que hagamos uso del bloque DIGIGEN que es un generador de señal digital cíclica (reloj) con impulso de duración configurable.

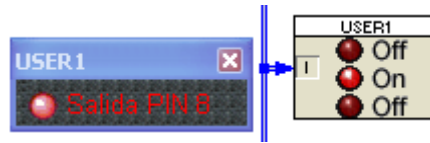


En la figura vemos la venta de designación de parámetros del bloque podremos modificar el valor del tiempo en estado ON y en estado OFF Ton y TOff

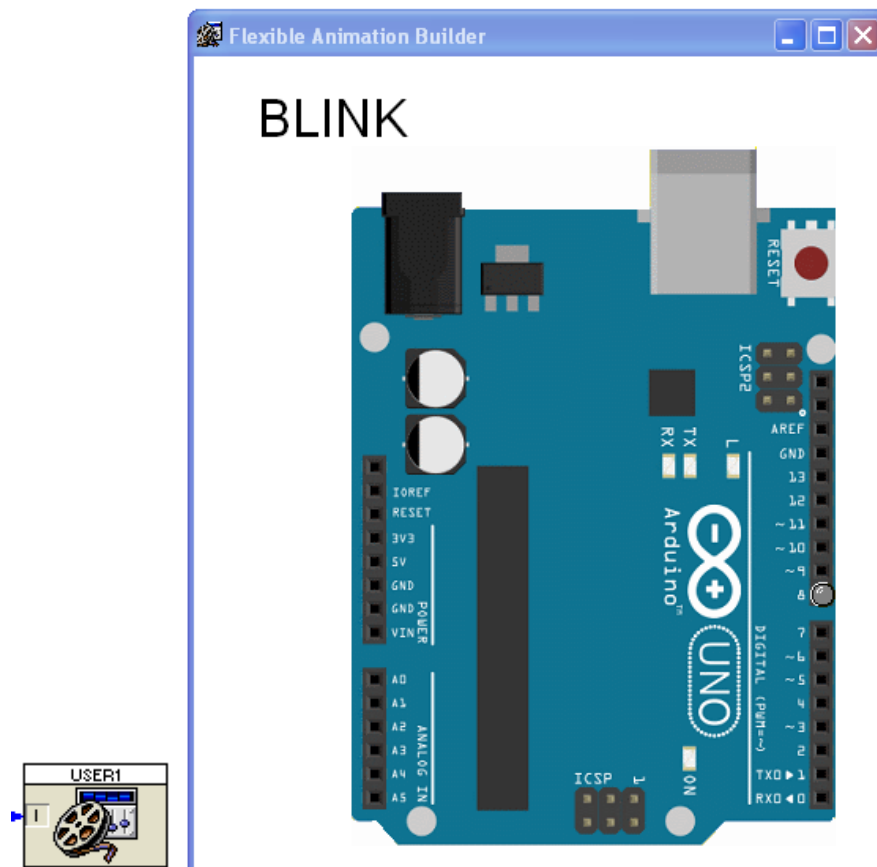


Se ha colocado también una imagen BMP en el área de trabajo con el montaje sobre protoboard del ejemplo.

Finalmente se ha colocado un bloque de visualización del valor que se envía a PIN8

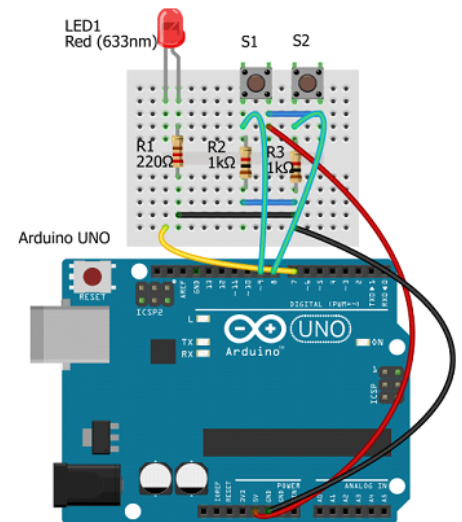
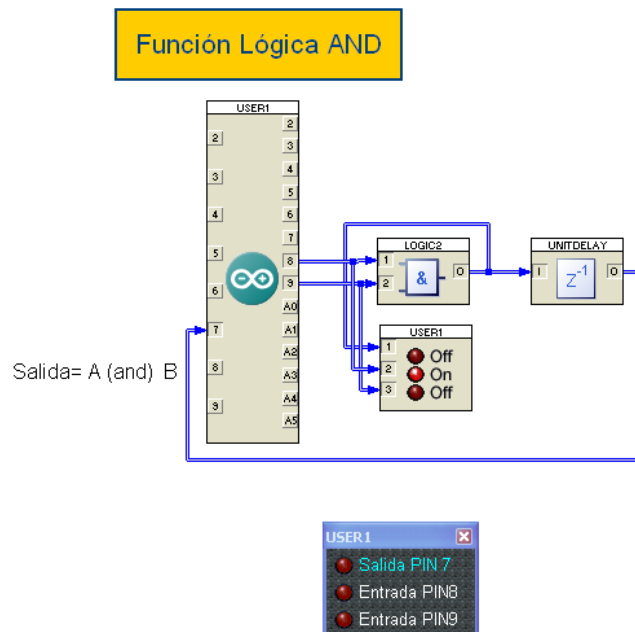


En el montaje se ha dispuesto un bloque de usuario para incorporar la herramienta **FAB** con la que se ha creado una pantalla en la que aparece un esquema de Arduino y un led que se sitúa en el PIN8 y que visualiza en tiempo real el estado de la salida física de Arduino



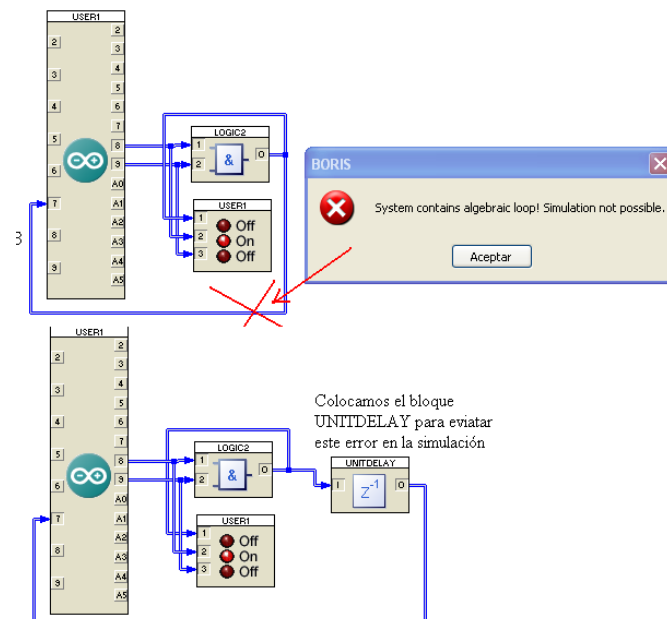
5.4. Función AND

En este ejemplo deseamos realizar una función AND de dos entradas cuyos valores recogemos de la tarjeta Arduino a través de los pines PI8 y PIN9 que actuaran como entadas digitales en la tarjeta Arduino a las que conectaremos unos pulsadores. La salida se realizara en el PIN7 que actuará como salida digital.



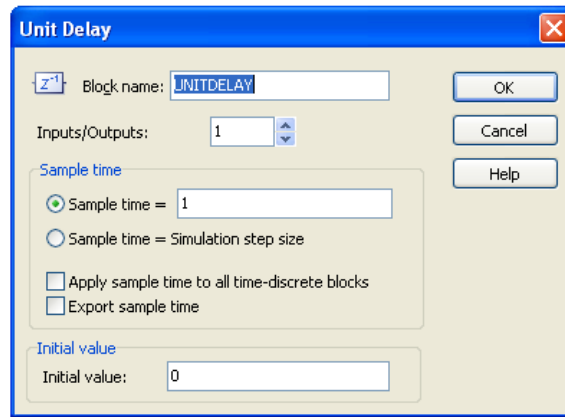
Esquema de montaje sobre protoboard.

En el montaje se ha colocado un bloque de retardo “UNITDELAY” que evita el error que se produce si cableamos directamente la función de salida del bloque LOGUC2 (función AND) hacia Arduino. Esto debemos hacerlo siempre que aparezca este mensaje.



Se ha colocado un bloque de monitorización del estado de las variables con el fin de poder ver como se comporta el sistema.

En el bloque de retardo debemos ajustar el tiempo de retardo a un valor que no desvirtúe el comportamiento de las señales en lo que se refiere al tiempo.



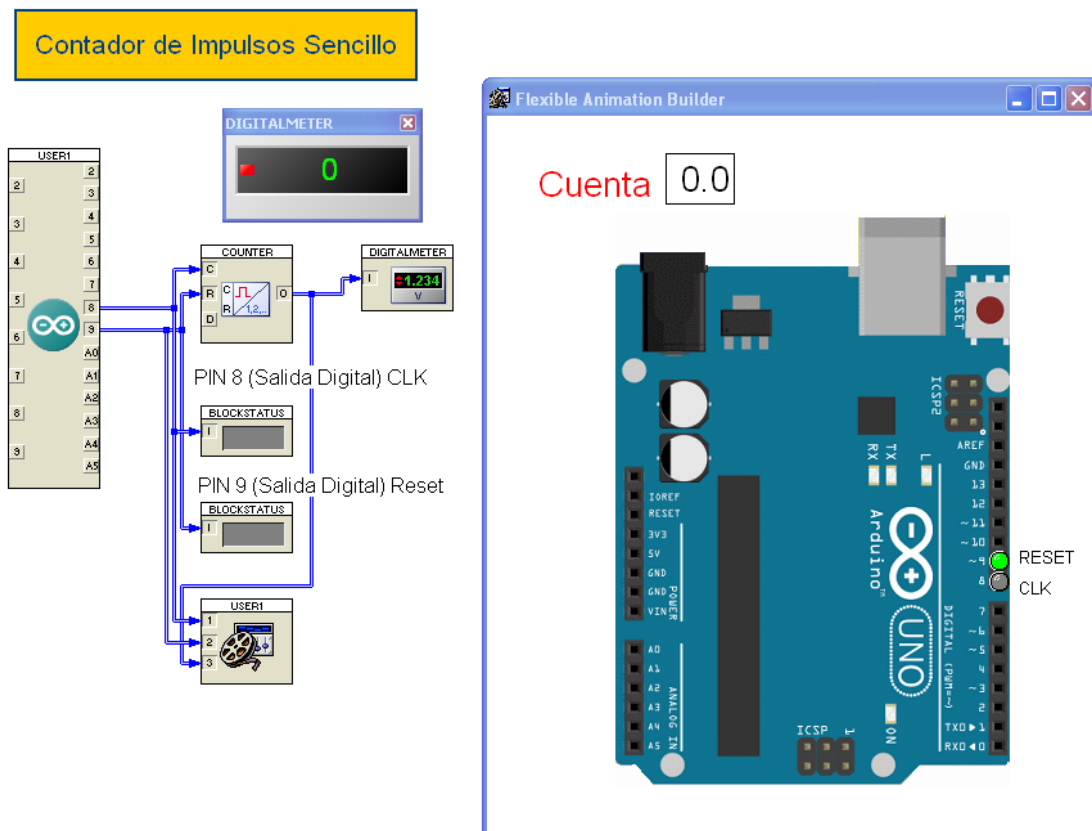
Se ha colocado un bloque para la visualización de la salida PIN7

5.5. Contador de impulsos sencillo

En este ejemplo utilizamos un bloque contador “COUNTER” que se alimentará de la señal digital de salida del PIN8 de Arduino. Para el Reset utilizaremos la señal del PIN9.

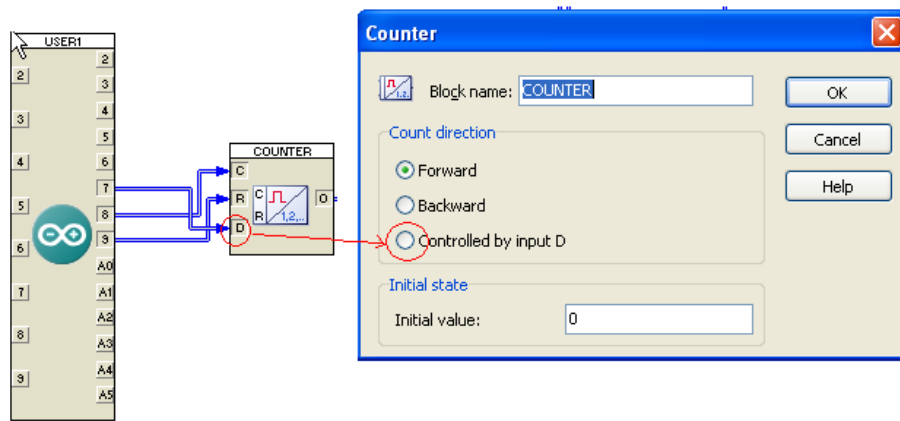
La monitorización del estado de los pines 8 y 9 se realiza con a ayuda de dos bloques “BLOCKSTATUS” y la salida del valor del contador se lleva aun bloque de visualización de valor digital “DIGITALMETER”

En esta aplicación se utiliza un bloque de Usuario para monitoriza los valores de las señales y además se ha colocado una imagen de Arduino. El bloque de Usuario FAB presenta tres entradas: CLK, Reset y Cuenta que representan los valores de entradas de impulso de cuenta y reset y el valor del contador.

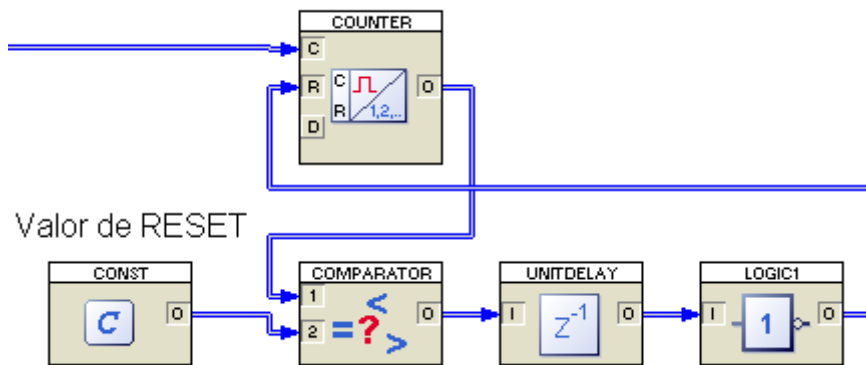


Para la simulación basta observar como dando impulsos en pulsador de CLK el contador se incrementa y pulsando en el pulsador del PIN9 se pone a cero.

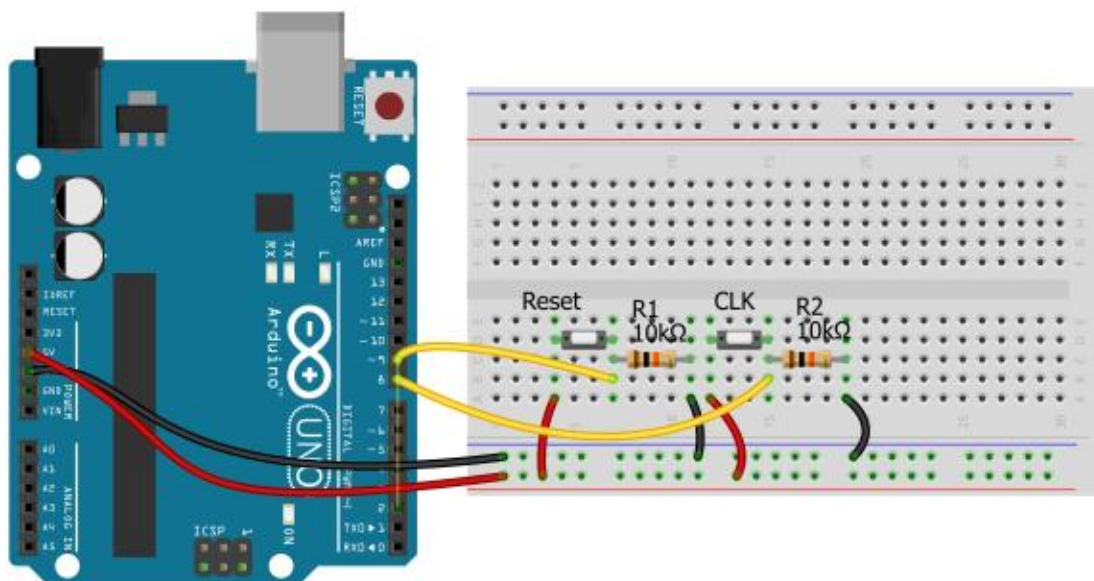
Como variantes en este ejemplo podríamos poner una señal que nos permitiese la cuenta hacia delante y otro hacia atrás (entrada D del contador) Bastaría con modificar el parámetro en la configuración del contador



Otra posibilidad es poner un valor de reset en el contador para que este se ponga a cero de manera automática. El montaje sería el siguiente.



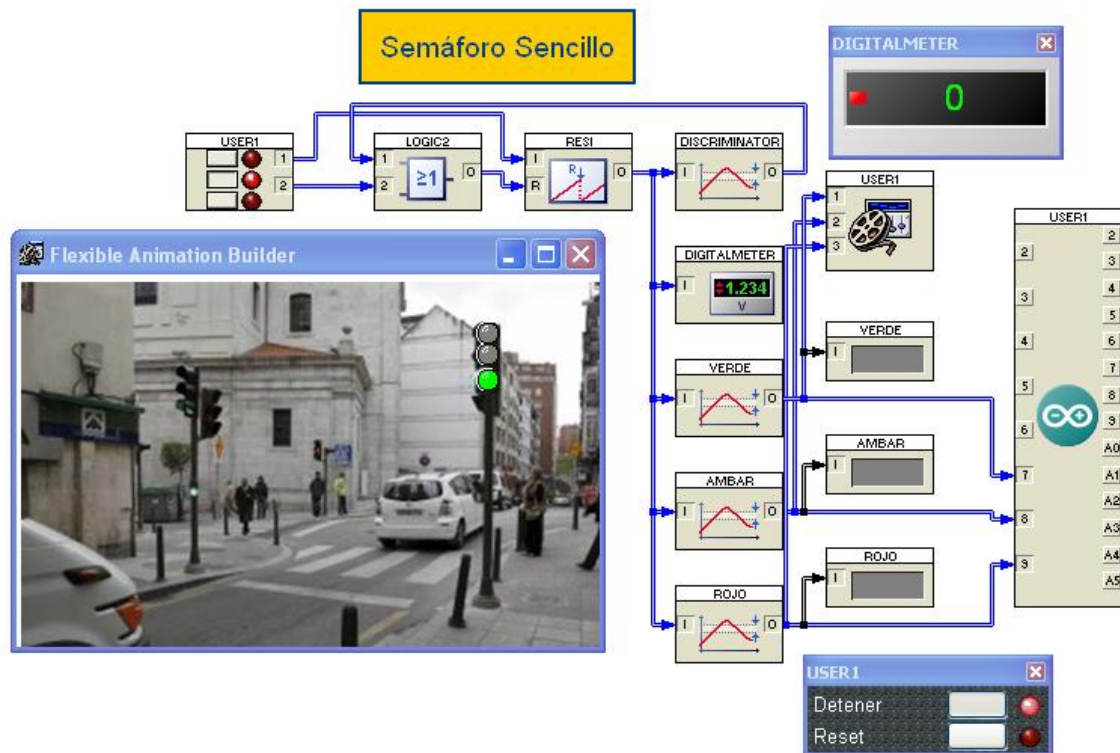
Montaje practico



5.6. Semáforo Sencillo


En el siguiente montaje se realiza la simulación del funcionamiento de un semáforo. Se recurre a un bloque RESI que lo que hace es contar en sentido ascendente hasta que se activa su entrada R.

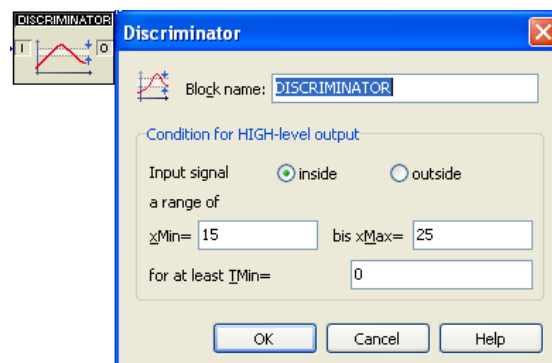
Se establecen los tiempos de encendido de cada **LED** en base a la respuesta de los bloques **DISCRIMINATOR** que mantiene su salida activa mientras el valor de sus entradas permanece en un rango que definimos nosotros en su ventana de parámetros.

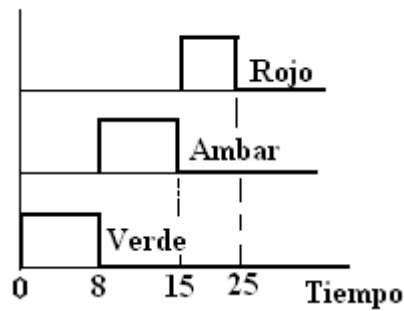


Se establecen los tiempos de cada salida y se colocó un discriminador mas para que



su salida ponga a cero el contador **RESI** (entrada R)  También se ha puesto un botón por si queremos reiniciar la secuencia estando esta a mitad de ejecución.





En la figura se ve la designación de tiempos de cada uno de los discriminadores para cada salida del semáforo.

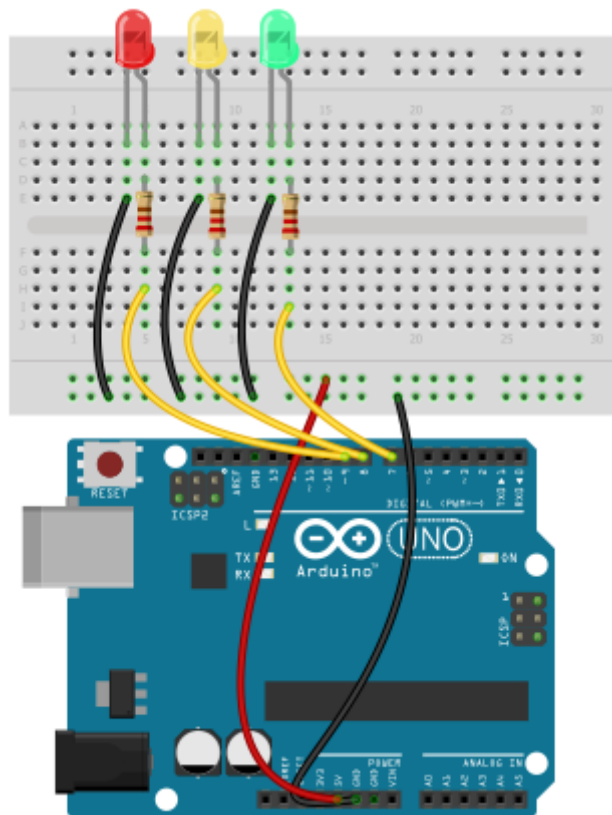
Las señales de cada lámpara se llevan a las salidas PIN7 PIN8 PIN9

Se han colocado bloques de visualización del estado de las lámparas “BLOCKSTATUS”.

Se ha colocado un bloque de usuario de tipo FAB en el que se muestra sobre una imagen de una calle con tres indicadores leds uno para cada color.

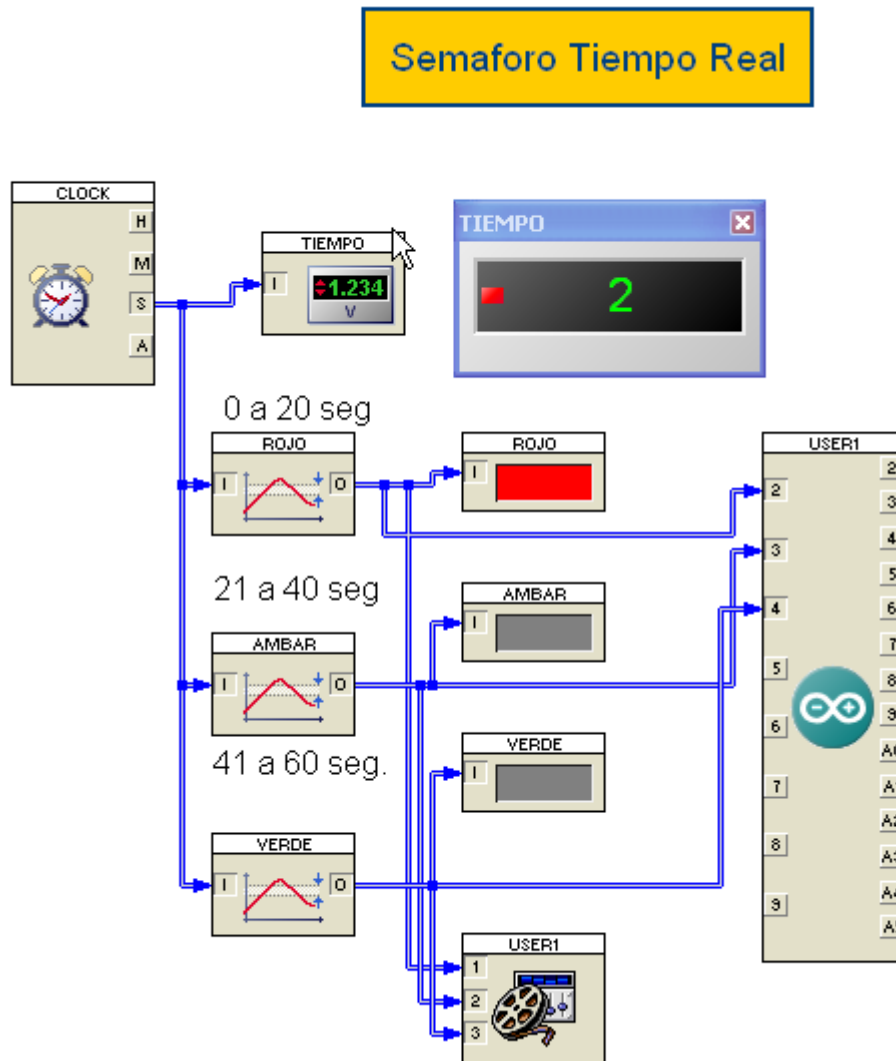
Se muestra también el estado del contador mediante un bloque “DIGITALMETER”.

Montaje en protoboard.



5.7. Semáforo en tiempo real

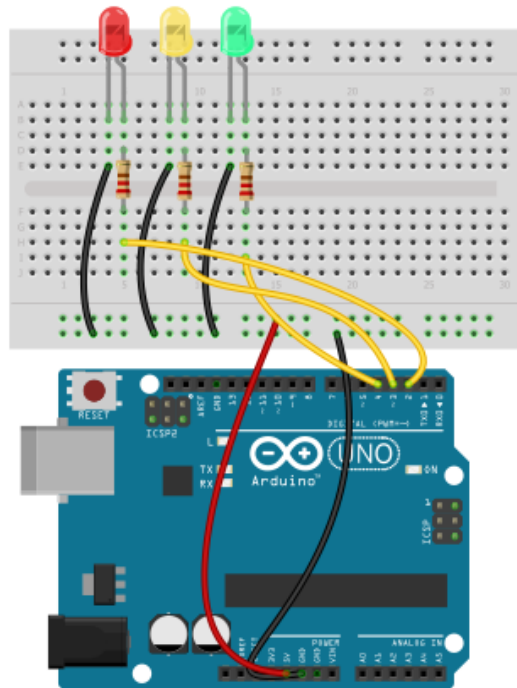
Este ejemplo es parecido al anterior pero en este caso el patrón de tiempo son los segundos que genera el bloque "CLOCK" (0 a 60). El valor se lleva a tres bloques "DISCRIMINATOR" que se encargan de activar las salidas de cada lámpara (Rojo, Ámbar y Verde)



Los tiempos de activación de cada una de las salidas están indicados en la imagen.

Las señales de cada lámpara se llevan a las salidas **PIN7 PIN8 PIN9**

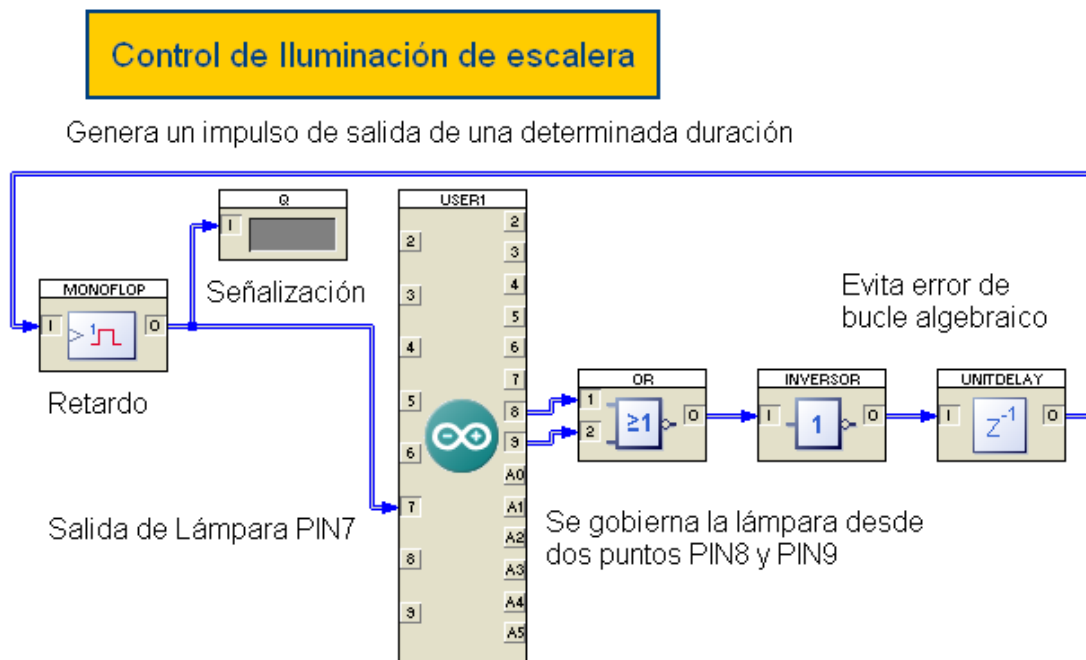
Montaje en protoboard



5.8. Control de Iluminación de Escalera

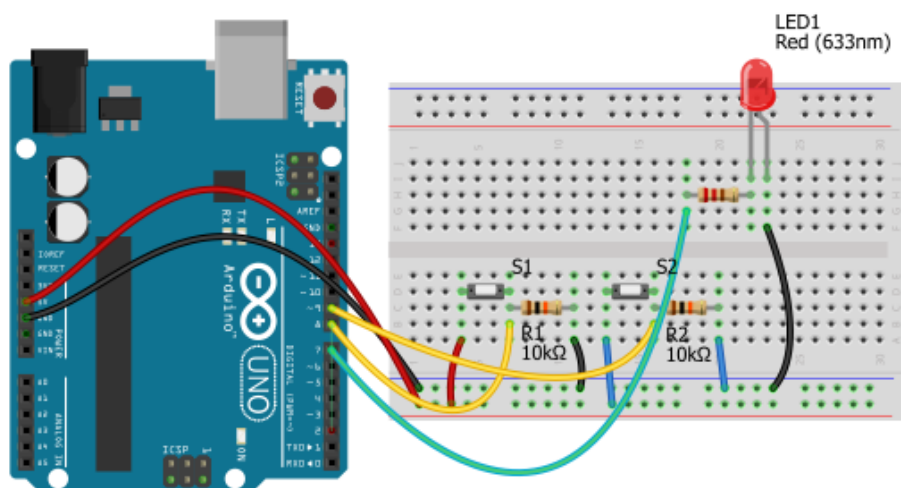
Este ejemplo sirve para simular el encendido de la lámpara de una escalera que puede ser gobernado mediante dos puntos.

La activación se realiza mediante las entadas digitales PIN8 y PIN9 y la salida de la lámpara se realiza en la salida digital PIN7



El intervalo de tiempo de activación se implementa mediante un bloque de tipo MONOFLOP. Las entradas PIN8 y PIN9 se llevan a un bloque de función LOGIC2 (OR) cuya salida se lleva a un inversor LOGIC1 y como no se puede realimentar directamente la señal al bloque Arduino se utiliza un bloque de retardo “UNITDELAY”

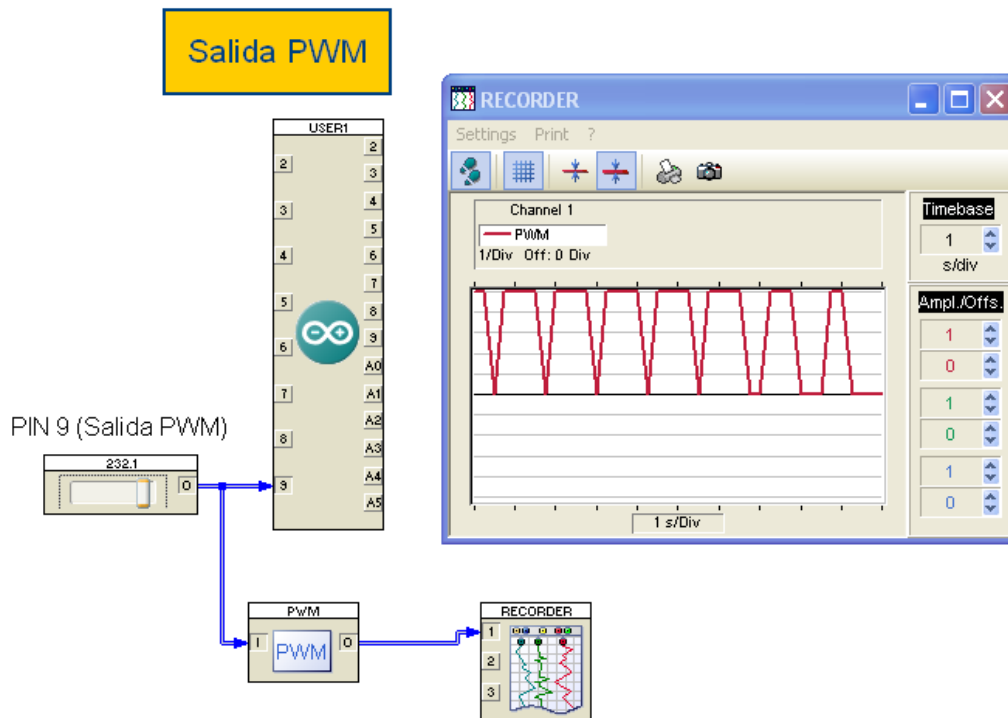
Montaje en protoboard.



5.9. Salida PWM

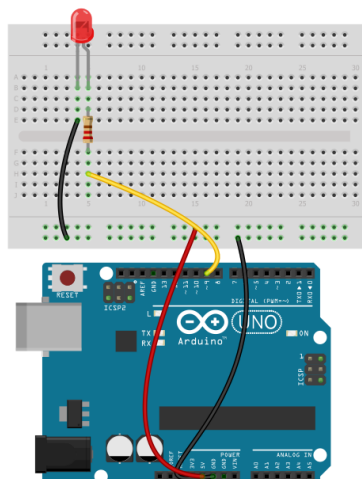
Las salidas analógicas en Arduino se realizan mediante un sistema de Modulación de Anchura de Pulsos PWM.

En este montaje configuramos el PIN9 como salida PWM y conectamos a este salida un Led que veremos iluminarse paulatinamente gobernado por un nivel de tensión de 0 a 255 que generamos de manera manual con un sencillo bloque del tipo “BLOCKPOTI”



Se ha colocado un bloque PWM que nos permite a través de un bloque trazador grafico “**RECORDER**” ver el tipo de señal. Estos dos bloques se han colocado sencillamente para poder mostrar un ejemplo grafico de lo que es una señal de esta naturaleza ya que para los efectos de control del Led del PIN9 ya el propio driver de comunicación se encarga de generar esta señal

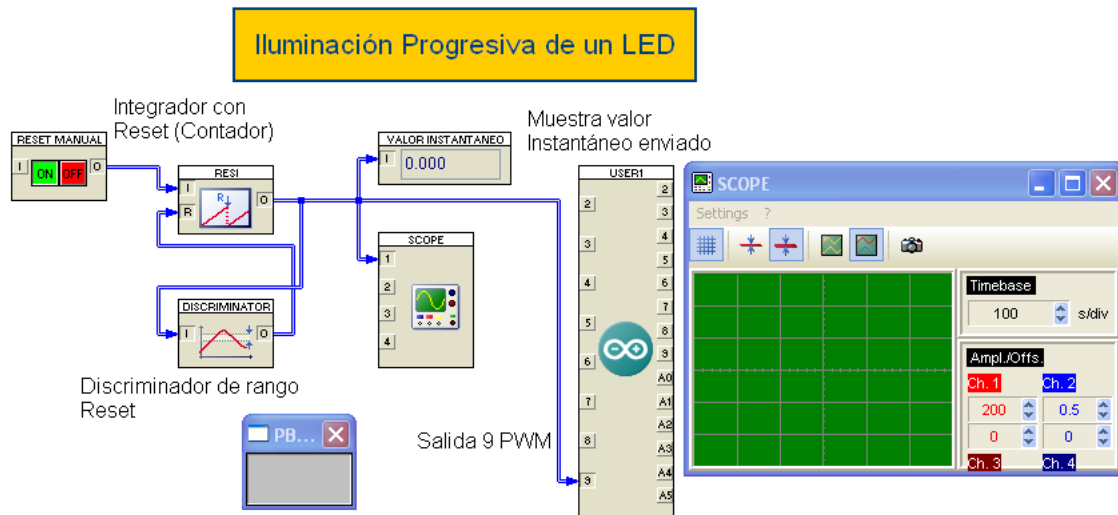
Montaje en protoboard.



5.10. Encendido Progresivo de un LED

En este ejemplo se realizará el gobierno de una salida tipo **PWM** en el **PIN9**.

La señal es una rampa generada por un integrador con entrada de Reset, bloque **RESI**. El reset lo realiza un bloque **DISCRIMINATOR** al que le establecemos una banda dentro de la cual se realiza la rampa.

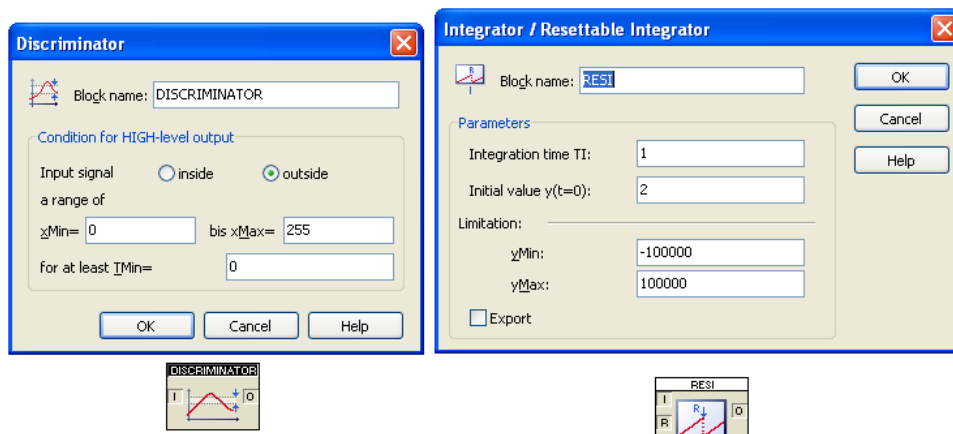


Se ha colocado un bloque de visualización y trazado **SCOPE** de la señal para ver la naturaleza de esta.

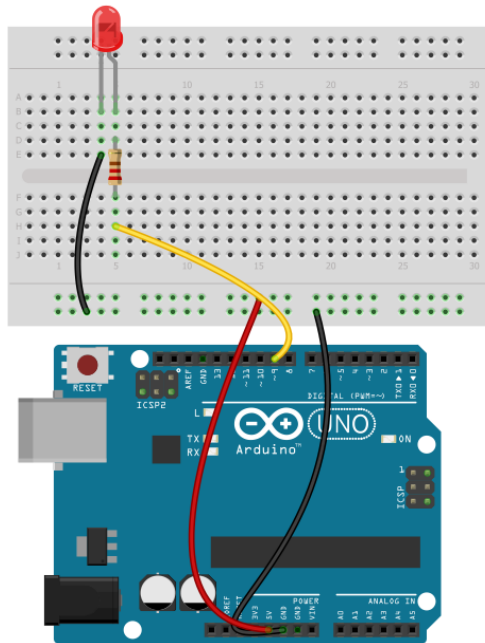
Se ha puesto un bloque para visualizar el valor numérico de la señal variable que se traslada al **PIN9**.

Para el reinicio manual de la rampa se ha colocado un botón, bloque **RESET MANUAL**

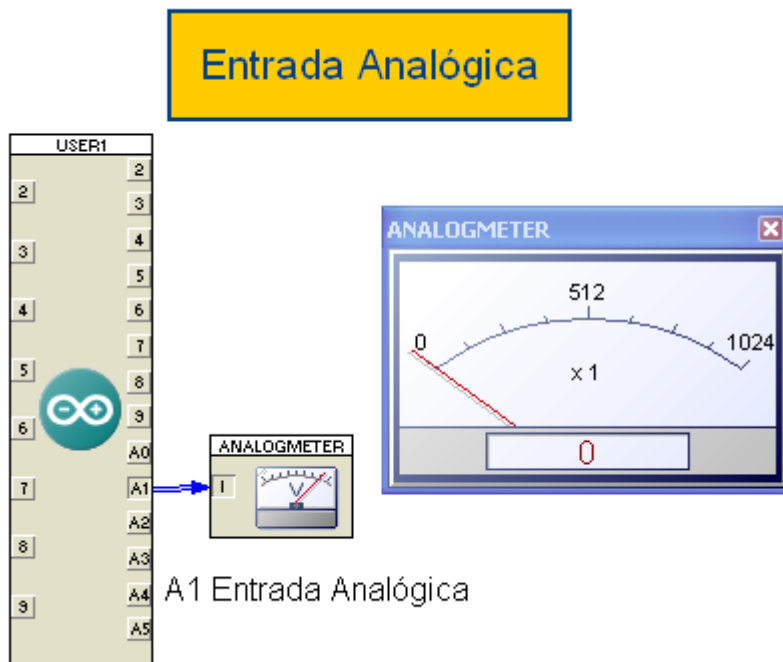
En la siguiente figura se observan las ventanas de configuración de los dos bloques que conforman el generador de la señal en rampa que variara de **0 a 255**.



Montaje en protoboard



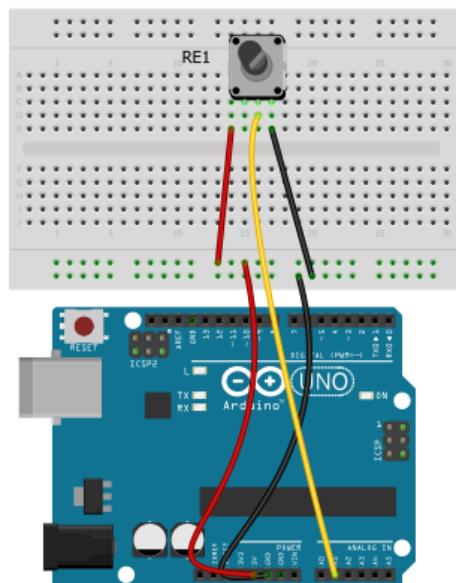
5.11. Entrada analógica 1



En este ejemplo vamos a leer un canal analógico de entrada y representaremos el valor en un instrumento de panel.

Los valores analógicos se encuentran en un margen comprendido entre 0 y 1024 por lo que el instrumento deberá tener ese rango de escala.

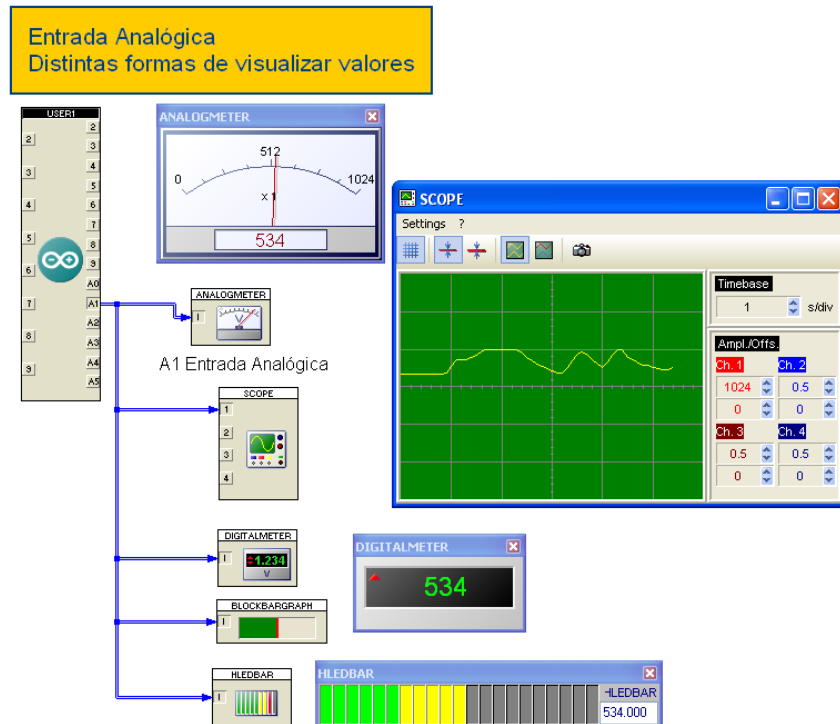
Podríamos también utilizar otros bloques de función de WinFACT que permitan la visualización e los valores.



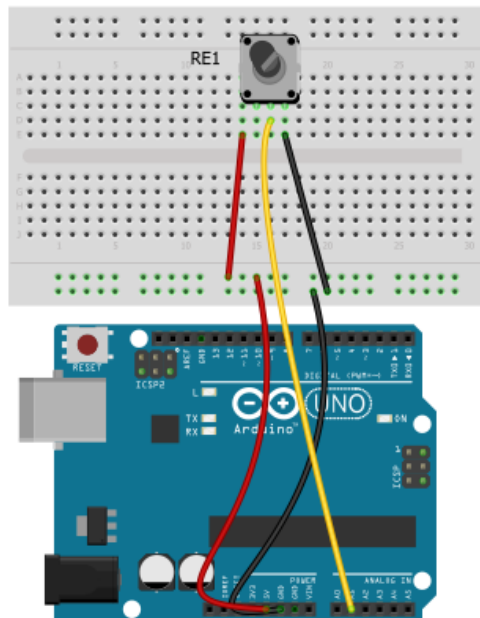
Montaje sobre protoboard

5.12. Entrada Analógica 2

En el siguiente ejemplo se lee igualmente una señal de entrada analógica y se han colocado distintos instrumentos para su visualización.



Montaje en protoboard



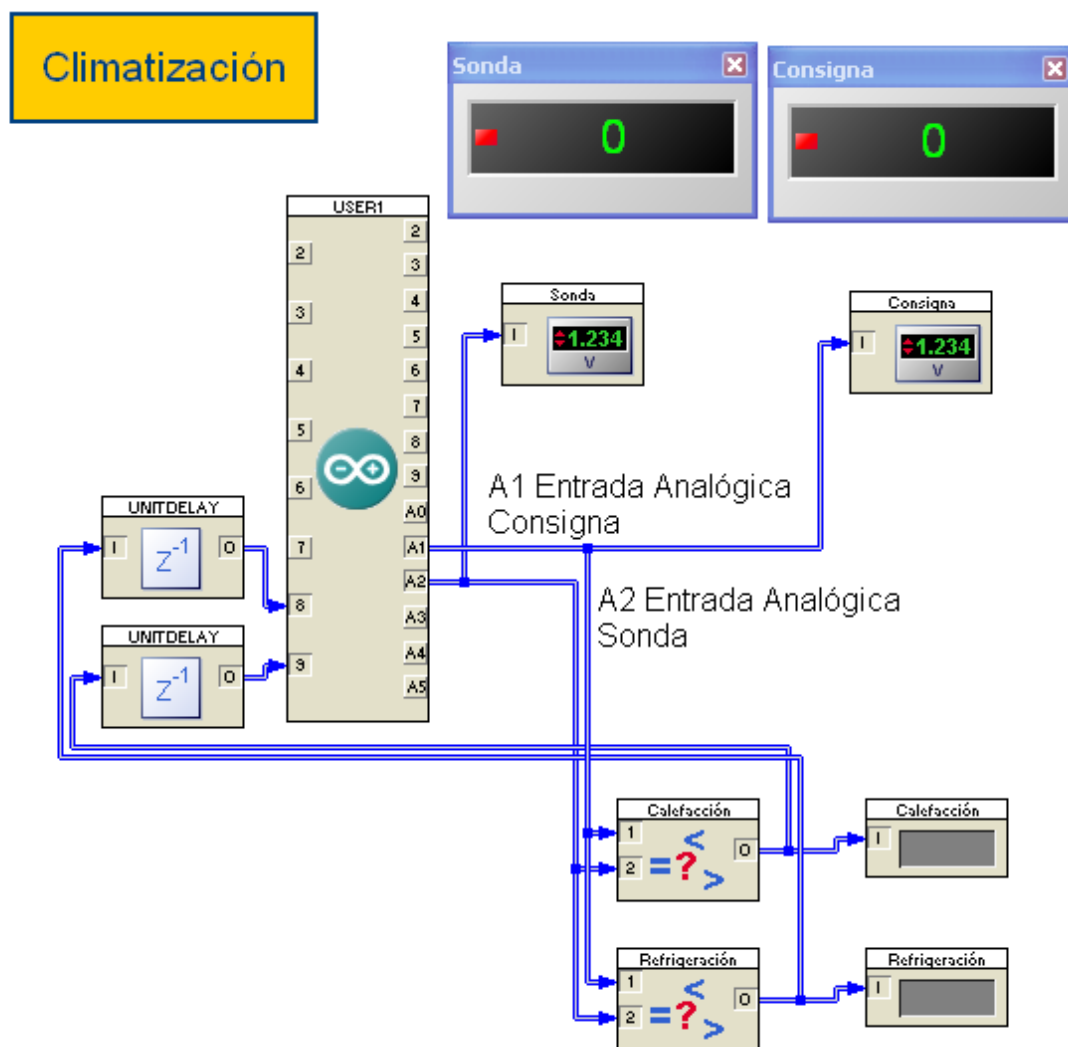
5.13. Climatización 1

Este ejemplo implementa un sistema básico de control de los equipos de calefacción y refrigeración en un local.

Las variables que se recogen de Arduino son dos de los canales analógicos A1 y A2.

El valor de consigna se recoge del canal A1 y será básicamente un potenciómetro con el que marcamos el valor sobre el que queremos realizar las conmutaciones para activar la calefacción o el aire acondicionado.

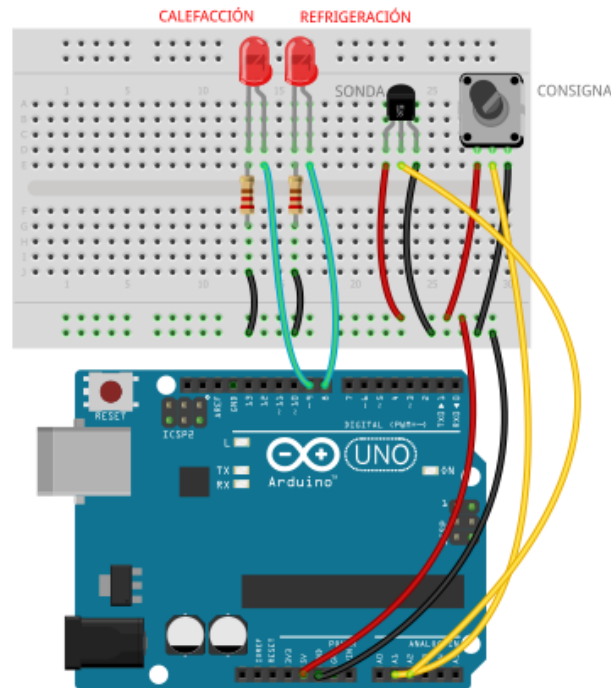
El segundo canal analógico es A2 y es el que a través de un sensor de temperatura nos indica el valor de esta.



Las salidas digitales PIN8 y PIN9 activan los elementos que gobiernan la calefacción y el aire acondicionado. Salida PIN8 activa la Refrigeración y salida PIN9 activa la calefacción.

Los elementos encargados del gobierno de las salidas son bloques de comparación (COMPARATOR) a los que se les ha consignado una banda de trabajo en función de los parámetros de comparación.

Esquema de montaje en protoboard



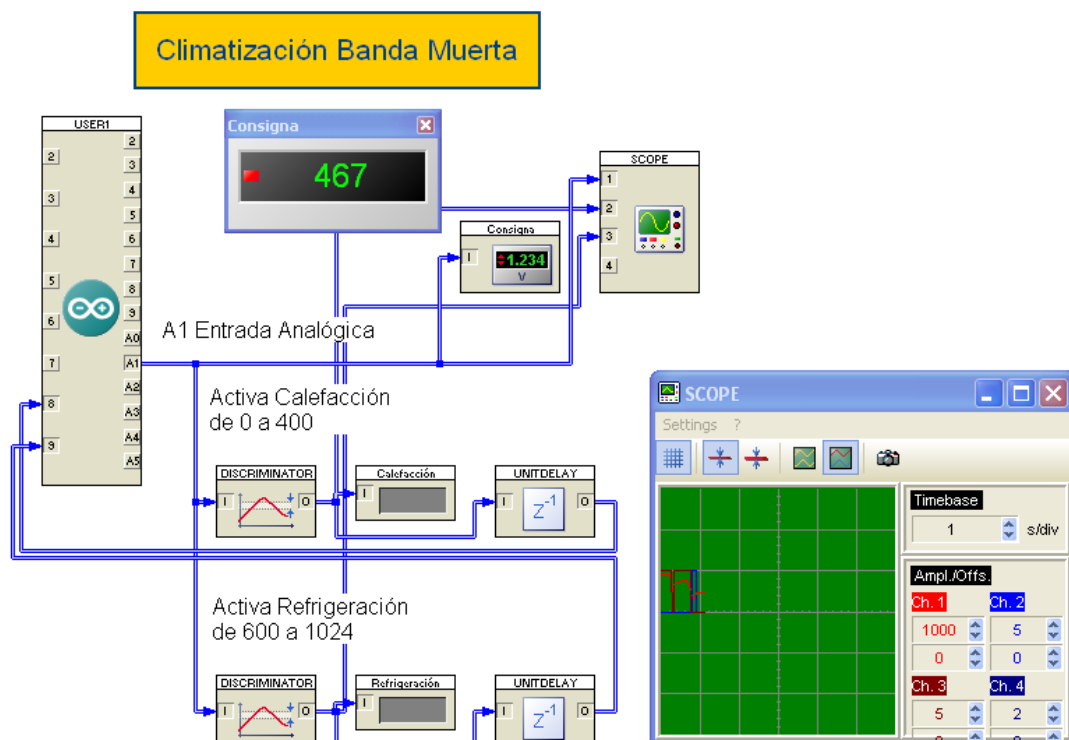
5.14. Climatización Banda Muerta

Con este ejemplo deseamos controlar la climatización de una sala.

Utilizaremos un sensor conectado a la entrada analógica de Arduino A1.

Queremos gobernar un aparato de calefacción y otro de aire acondicionado.

Para el gobierno de estos aparatos recurrimos a las salidas digitales de Arduino PIN8 (Calefacción) y PIN9 (Refrigeración)

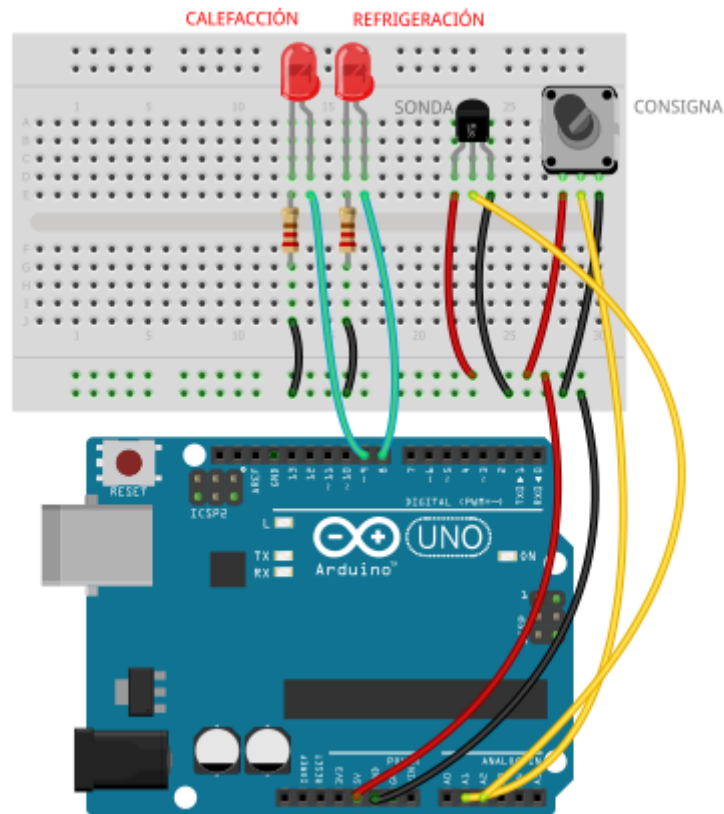


El control se realiza mediante dos bloques de tipo “**DISCRIMINATOR**” en los que definimos la banda de trabajo tanto del elemento calefactor como del refrigerador.

Se han colocado dos bloques de retardo “**UNITDELAY**” que evitan el error de bucle.

Montaje en protoboard

Esquema de montaje en protoboard

**AGRADECIMIENTO:**

Agradezco al *Dr.-Ing. Jörg Kahlert* de la firma *Ingenieurbüro Dr. Kahlert* la ayuda y buena disposición que me ha prestado para poder elaborar este manual.

Profesor:
José Manuel Ruiz Gutiérrez
Octubre 2013