



BASIC I/O

Manual de usuario

Enero 2018



MK Electrónica

c/ Ipergorta, 11 – 3ºA
48410 Orozko - Bizkaia
Tfno.: 34- 944 04 80 89
info@mkelectronica.com
www.mkelectronica.com



www.mkelectronica.com
info@mkelectronica.com

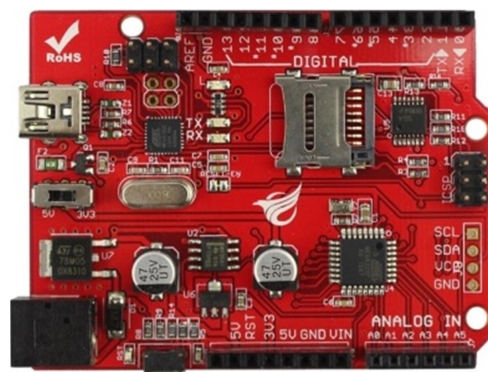
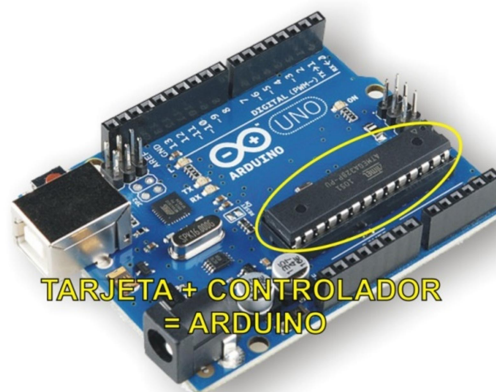
BASIC I/O

Arduino BASIC I/O

1 - INTRODUCCION

Desde que allá, por el año 2005, apareció la plataforma Arduino hasta nuestros días, su popularidad no ha hecho más que crecer de forma excepcional. Arduino consiste en una tarjeta electrónica sencilla y de muy bajo coste que incorpora un controlador que se programa mediante un lenguaje de alto nivel gratuito y fácil de aprender.

La plataforma Arduino puede ser usada por personas con escasos o ningún conocimiento técnico; desde estudiantes de ESO, Bachiller y ciclos formativos de grado medio y superior, hasta aficionados a la tecnología, la electrónica, la programación, “makers” y profesionales de todo tipo.

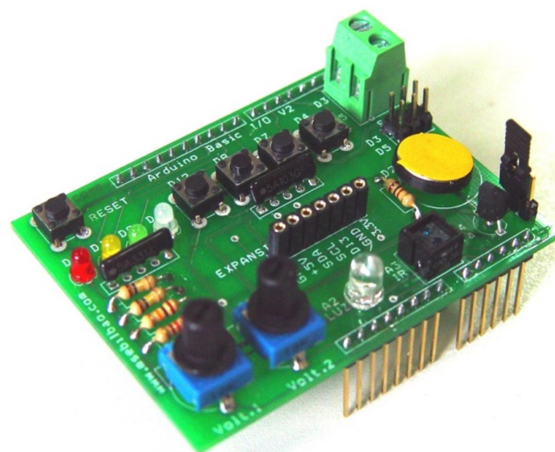


Es tal su popularidad que además de las tarjetas originales, en el mercado puedes encontrar desde simples copias o “clones” hasta tarjetas 100% compatibles con aquellas y que incluso incorporan características y prestaciones mejoradas. Como ejemplo puede servir la tarjeta Crowduino UNO de la imagen.

Atendiendo a las sugerencias de nuestros clientes y usuarios, entre los que se encuentran un buen número de profesionales de la

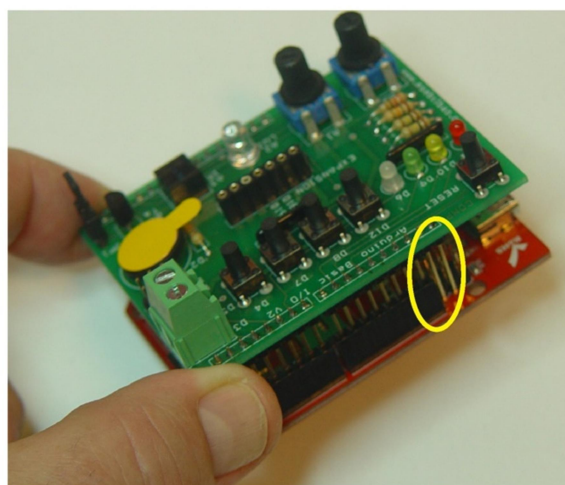
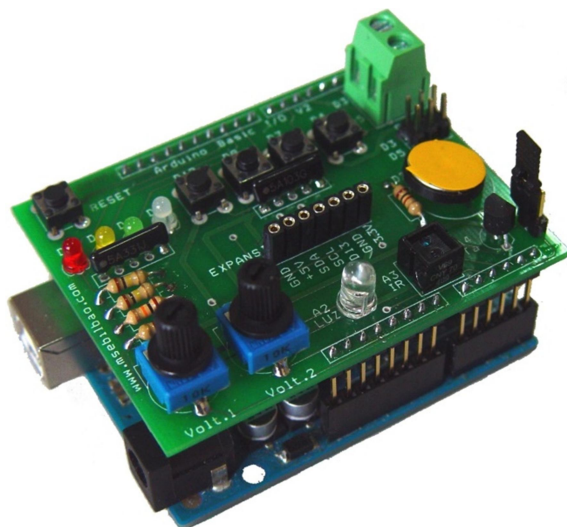
enseñanza, se nos propuso el diseño, fabricación y comercialización de una tarjeta electrónica que contuviera una serie de periféricos elementales. Esa tarjeta tenía que ser sencilla, económica y que se pudiera insertar sobre la plataforma Arduino UNO o cualquiera de las tarjetas compatibles existentes.

La idea es que el alumno o el usuario se centre, al menos inicialmente, en la programación para el control de los distintos periféricos que ya se encuentran debidamente conectados, comprobados y funcionando en nuestra tarjeta. El resultado es la tarjeta BASIC I/O.

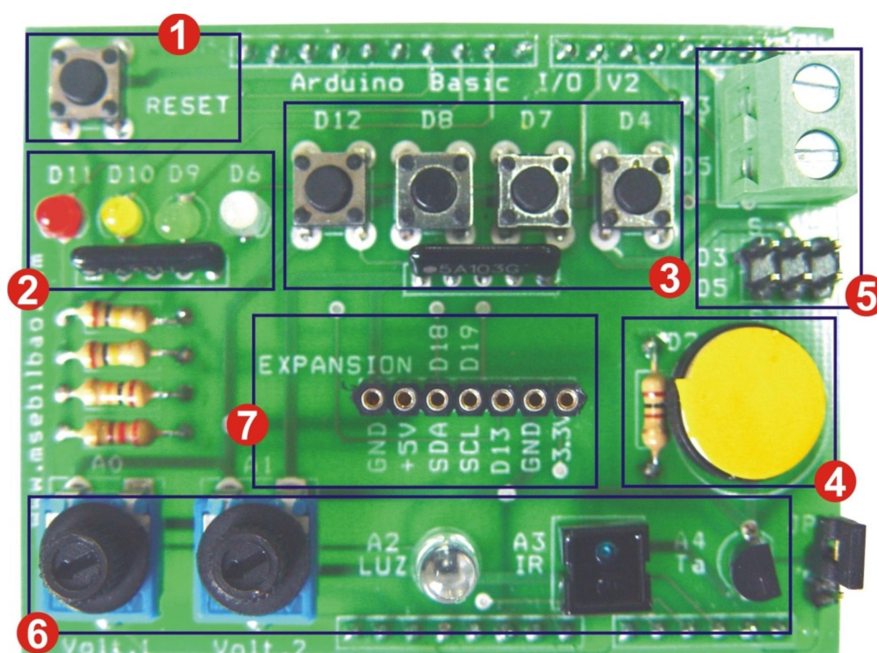


2 - CARACTERÍSTICAS

La tarjeta BASIC I/O se presenta totalmente montada y comprobada, lista para insertarse sobre la controladora Arduino y comenzar a programar. Es compatible con las plataformas Arduino UNO, LEONARDO, MEGA y compatibles. Basta insertarla con la orientación adecuada en cualquiera de ellas, prestando atención a que ninguna de las patillas se doble y quede sin conexión. En la figura se muestra BASIC I/O V2 insertada sobre Arduino UNO y sobre la compatible Crowduino UNO. En este caso quedan dos patillas sin insertar (no se usan).



En la siguiente figura puedes observar las diferentes secciones que componen la tarjeta BASIC I/O V2 y que se explican a continuación.



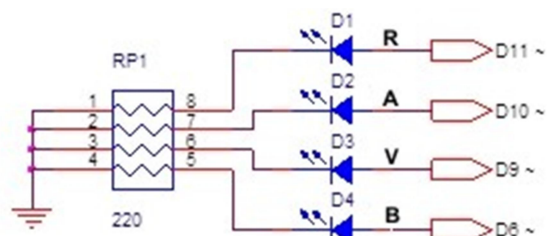
2.1- Pulsador RESET

Es equivalente al pulsador RESET de la tarjeta Arduino. Cada vez que se acciona se produce un reinicio del controlador. A efectos prácticos es similar a conectar nuevamente la alimentación del sistema. Su circuito eléctrico es el siguiente:



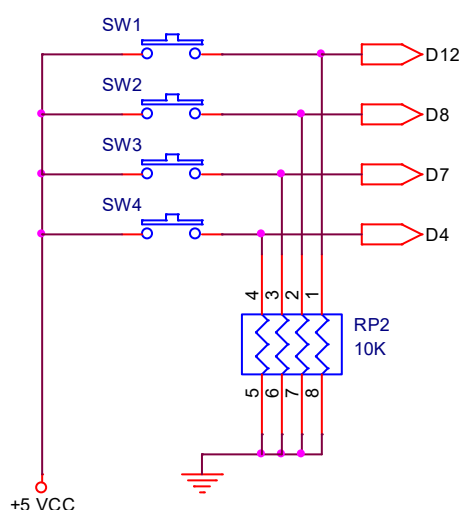
2.2 - Salidas Digitales

Según su esquema corresponden a las patillas D11, D10, D9 y D8 que están conectadas con los diodos leds rojo, ámbar, verde y blanco respectivamente. Cada led se puede activar (mediante nivel "1") o desactivar de forma individual bajo el control del programa. Piensa que de forma similar a cómo controlas un led, podrías controlar otros periféricos como relés, válvulas, motores, etc...



Esas patillas también pueden actuar, bajo el control de tu programa, como salidas de señal PWM. Puedes experimentar con ejemplos y aplicaciones de regulación de potencia e iluminación.

2.3 - Entradas digitales



Están formadas por cuatro pulsadores de contactos normalmente abiertos y conectados con las patillas D12, D8, D7 y D4. Mira el esquema de la figura.

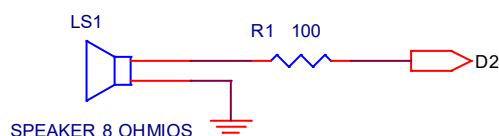
En estado de reposo, sin pulsar, mantienen a nivel "0" cada entrada. Cuando se acciona y pulsa cualquiera de ellos, la patilla correspondiente pasa a nivel lógico "1".

Mediante las funciones apropiadas del lenguaje de programación, puedes conocer el estado de cada una de estas entradas y actuar en consecuencia.

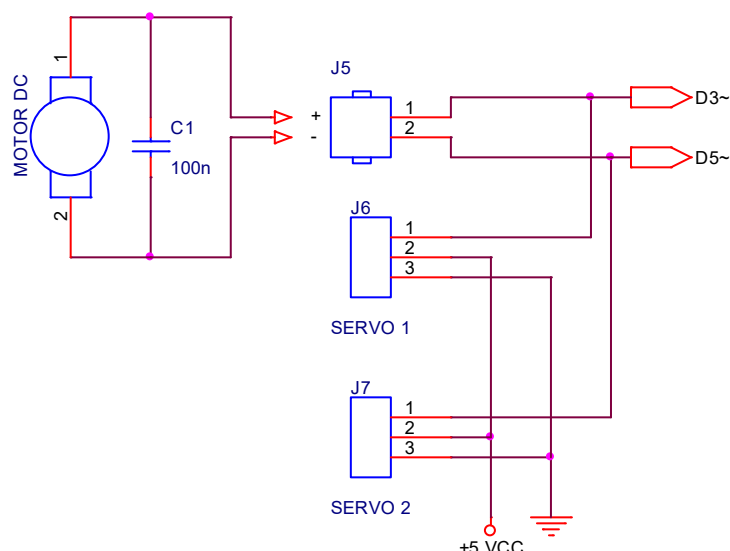
Si eres capaz de conocer el estado de un pulsador, también podrás conocer el estado de un sensor digital, un final de carrera o cualquier otro periférico conectado a cualquiera de las patillas digitales de la tarjeta controladora Arduino.

2.4 - Salidas al Altavoz

La patilla D2 del controlador Arduino es empleada por la tarjeta BASIC I/O V2 para actuar sobre un pequeño altavoz. Gracias a él puedes dotar a tus programas de música, sonidos, señales acústicas de aviso, de alarma, etc...



2.5 - Salidas a Motores/Servomotores



La tarjeta BASIC I/O utiliza las patillas D3 y D5 para controlar a un pequeño motor DC. Como estas patillas pueden actuar como salidas de señal PWM, también puedes conectarlas a dos servomotores diferentes. Observa el esquema.

Por un lado, en el borne J5 puedes conectar un motor DC. Con las salidas D3 y D5 puedes controlar el arranque o parada del mismo, así como el sentido de giro. Como estas salidas también son PWM, puedes regular su velocidad.

Por otro lado dispones de los conectores J6 y J7 de tres pines y donde puedes conectar hasta dos servomotores controlados por las señales PWM disponibles en las patillas D3 y D5. No solo puedes controlar el sentido de giro sino también el posicionamiento de los ejes de esos servos.

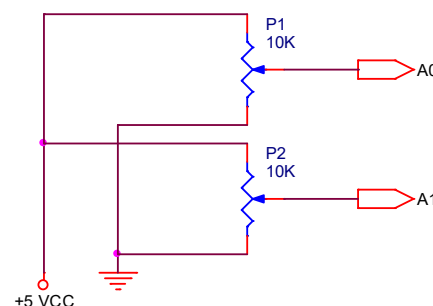
IMPORTANTE: Las salidas D3 y D5 de Arduino proporcionan una corriente máxima de 40 mA. No debes conectar ningún dispositivo cuyo consumo sea superior. Puedes estropear el controlador.

2.6 - Entradas analógicas

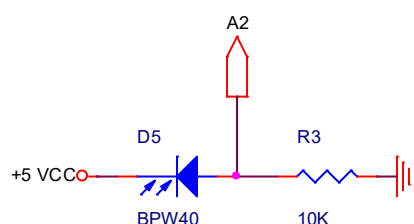
Arduino es capaz de recoger y procesar información procedente de periféricos analógicos. Normalmente estos periféricos suministran una tensión que varía en función de la luz, temperatura, presión, velocidad, etc... En el mercado hay un gran número de dispositivos y sensores analógicos capaces de recoger prácticamente cualquier magnitud física. Nuestra tarjeta BASIC I/O incorpora algunos de ellos. Los puedes ver de izquierda a derecha.

Potenciómetros

Hay dos. Están conectados con las patillas A0 y A1 del controlador Arduino. Cuando mueves el eje de cualquiera de ellos varías la tensión presente en esas patillas desde un mínimo (0V) hasta un máximo (5V).



Sensor de luz visible



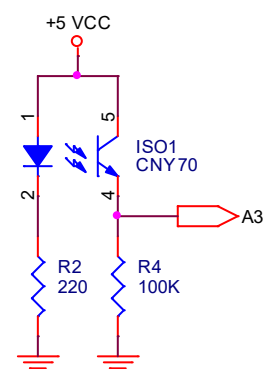
Está basado en el fototransistor BPW40 (D5) que deja pasar una corriente proporcional a la luz visible que incide sobre él. Esta corriente provoca una tensión en la resistencia R3 que se aplica a la patilla A2 del controlador Arduino. En definitiva, cuanto mayor sea la cantidad de luz que incide sobre el sensor, mayor es la tensión en la entrada analógica A2.

Imagina aplicaciones de control de luz ambiente, iluminación, regulación, alarmas, etc...

Sensor de reflexión

Se trata del modelo CNY70. Consiste en un diodo emisor de luz infrarroja (IR) y de un fototransistor que detecta ese tipo de luz, encerrados en una misma cápsula. El dispositivo emite luz IR de forma constante. Si se pone un objeto frente a él, esa luz rebota (reflexión) y retorna al fototransistor. Este produce una tensión proporcional a esa luz que se aplica a la entrada analógica A3 del controlador.

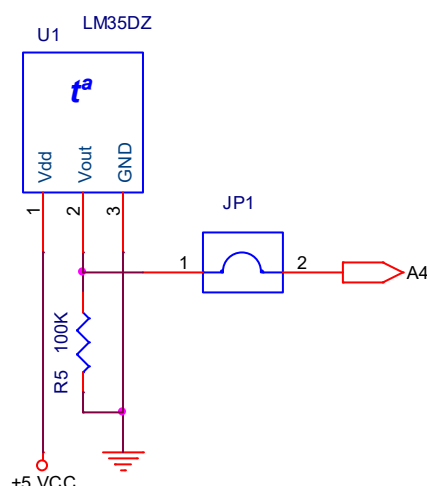
Cuanto mayor sea la luz reflejada mayor será la tensión en A3. Según esto, si colocas un objeto alejado del sensor (más de 1 cm aprox.) la luz se dispersa y no reflexiona. La tensión analógica será mínima, igual que si colocas un objeto que absorbe la luz (color oscuro). Puedes por tanto experimentar a detectar la proximidad y los colores de los distintos objetos que pongas frente al sensor.



NOTA: La luz IR no es visible por el ojo humano.

Sensor de temperatura

Es el último de los sensores disponibles y consiste en el modelo LM35DZ. Puede trabajar con diferentes configuraciones aunque en BASIC I/O se emplea la más sencilla. En este caso el sensor ofrece una tensión de 10mV por cada °C.



La salida del sensor se conecta con la entrada analógica A4 a través del puente JP1. El número de proyectos y aplicaciones que puedes realizar con este sensor son numerosos: control de temperatura, control ambiental, termostatos, alarmas, etc...

IMPORTANTE: La patilla A4 del controlador Arduino se puede configurar para que actúe como señal SDA del bus I2C o señal D18 de E/S digital. Si la vas a usar en estos modos debes dejar abierto el puente JP1.

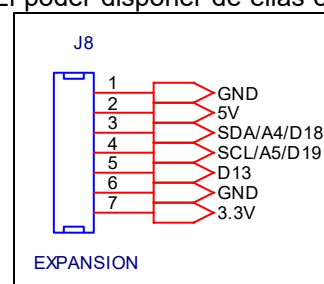
2.7 - Conector de Expansión

Consiste en un conector, el J8, que transporta las señales del controlador Arduino mostradas en la figura. Además de las señales de alimentación GND, 5V y 3.3V transporta las señales SDA/A4/D18, SCL/A5/D19 y D13.

Las dos primeras, SDA y SCL, son muy empleadas por el conocido bus I2C. El poder disponer de ellas en la tarjeta BASIC I/O te va a permitir insertar y utilizar multitud de dispositivos que existen en el mercado y que se gobiernan según este protocolo I2C.

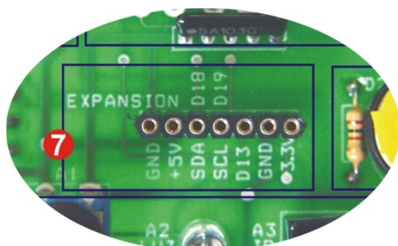
También puedes usarlas, junto con D13, como líneas D18 y D19 de E/S de propósito general. En ellas puedes conectar cualquier circuito o periférico que diseñes por tu cuenta.

En resumidas cuentas, gracias a este conector puedes insertar en BASIC I/O otros circuitos que quedarán conectados con el controlador Arduino. En este documento se proponen algunos ejemplos.



3 – CONEXIÓN DE DISPOSITIVOS

Gracias al conector de expansión J8 de 7 patillas podemos conectar diferentes dispositivos a la tarjeta BASIC I/O V2 y por lo tanto controlarlos desde el Arduino. En la figura se muestra las señales que transporta y cómo están distribuidas.



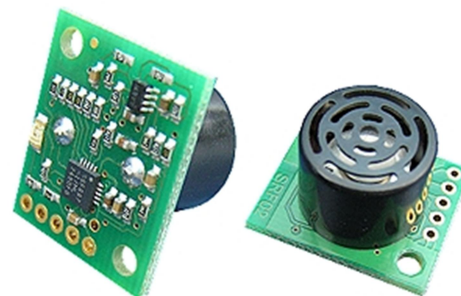
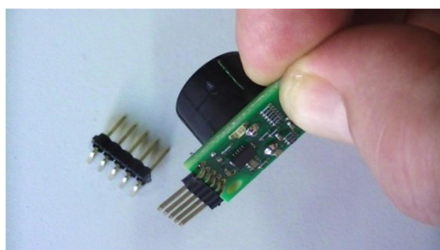
Patilla N°	NOMBRE	DESCRIPCIÓN
1	GND	Señal de tierra de alimentación 0V.
2	+5V	Señal de alimentación de +5V.
3	SDA/A4/D18	Señal de datos SDA del protocolo I2C, entrada analógica A4 y E/S digital D18 de propósito general.
4	SCL/A5/D19	Señal de reloj del protocolo I2C, entrada analógica A5 y E/S digital D19 de propósito general.
5	D13	E/S digital D13 de propósito general.
6	GND	Señal de tierra de alimentación 0V.
7	3.3V	Señal de alimentación de 3.3V

Ya sabes que la patilla SDA/A4/D18 está también conectada con el sensor de temperatura LM35 DZ incorporado en BASIC I/O. Recuerda abrir el jumper JP1 para desconectarlo y evitar que influya en su funcionamiento como señal SDA/A4/D18.

En este apartado se va a proponer la conexión de varios de estos dispositivos que pueden ser comerciales, o diseñadas y fabricadas por cada uno.

3.1 – Medidor ultrasónico de distancias SRF02

Se trata de un sensor capaz de medir, mediante ultrasonidos, la distancia de un objeto respecto a él. Está fabricado por la firma inglesa Devantech Ltd: <http://www.robot-electronics.co.uk/htm/srf02tech.htm>.



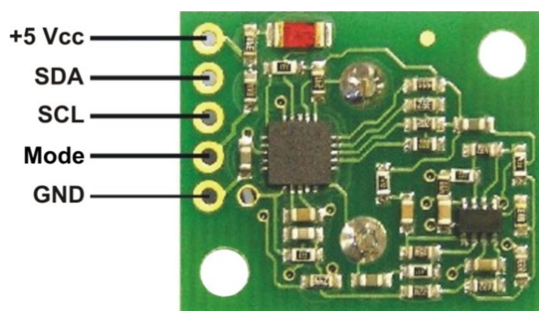
Deberás empezar soldando una tira de 5 pines macho y acodada tal y como se muestra en la figura.

Las características más relevantes se pueden resumir en:

- Rango de medida de 16 cm a 6 m (en teoría)
- Alimentación: +5V a 4 m A
- Frecuencia ultrasónica: 40 MHz
- Dimensiones: 24 mm x 20 mm x 17 mm
- Control automático de ganancia
- Comunicación con el controlador principal o Host: vía serie o vía I2C
- No precisa calibración
- Resultado de la medida en: μ S, centímetros o pulgadas
- Dirección del SRF02 como Slave y por defecto 224 (0xE0). Se puede configurar entre otras 15 direcciones posibles.

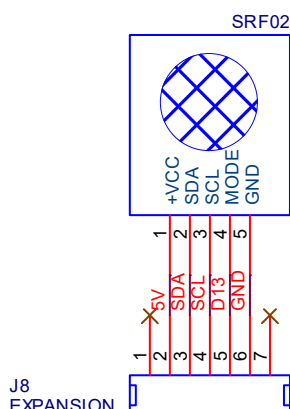
BASIC I/O

El medidor ultrasónico SRF02 actúa como un dispositivo I2C Slave e incluye su propio controlador encargado de realizar las medidas y calibraciones oportunas, para luego transmitir los resultados a l controlador Host (Arduino, PIC, Basic Stamp, PICaxe, etc... Estas son sus conexiones eléctricas:



PIN	NOMBRE	DESCRIPCION
1	+5Vcc	Tensión de alimentación de + 5V
2	SDA	Señal de datos del bus I2C
3	SCL	Línea de reloj del bus I2C
4	Mode	Comunicación con el Host. A "0" comunicación serie tipo USART. A nivel "1" o sin conexión comunicación tipo I2C.
5	GND	Tierra de alimentación

Tal y como se muestra en la figura, utilizamos el conector de expansión de la tarjeta BASIC I/O para conectarlo con el controlador Arduino. El sensor trabaja con una alimentación de +5Vcc y emplea el protocolo I2C para la transferencia de datos con el Arduino. Para ello utiliza las señales SDA y SCL disponibles en el conector. Opcionalmente también puede trabajar en modo USART serie. Observa que las dos patillas extremas del conector de expansión quedan libres, sin conexión.

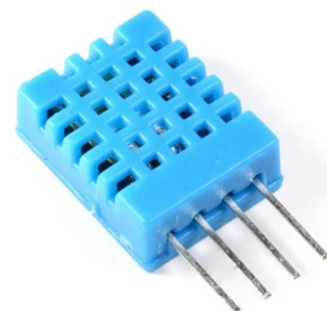


3.2 – Módulo MR003-005.1

Es un dispositivo basado en el sensor DHT11 desarrollado por Aosong, www.aosong.com, que integra un sensor de temperatura basado en una resistencia NTC y un sensor de humedad del tipo resistivo. Está fabricado por la firma Italiana Microbot, www.microbot.it.

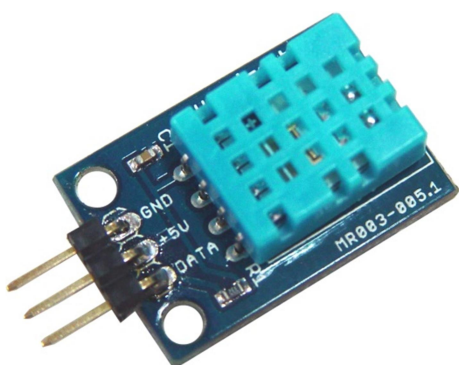
Sus características son:

- Tensión de alimentación: 3.5 – 5V
- Consumo: 0.3 mA durante la medida y 60 μ A en reposo
- Periodo de muestreo: más de 2 segundos



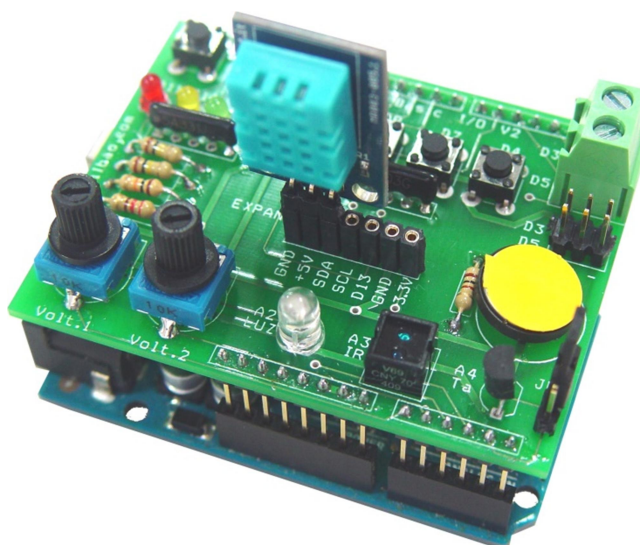
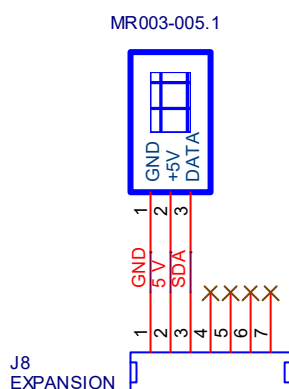
PARÁMETRO	HUMEDAD	TEMPERATURA
Resolución	8 bits	8 bits
Repetitividad	± 1% RH	± 0.2 °C
Precisión	± 5 % a 25 ° C	± 1 °C
Rango	20 % - 90 % de RH a 25 °C	0 °C – 50 °C

El módulo consta de tres únicas patillas de conexión con el controlador Host (Arduino, PIC, AVR, Stamp, etc...). Deberás soldar una tira de 3 pines macho y acodada. Mira la figura y la tabla que se adjunta.



PIN	NOMBRE	DESCRIPCION
1	GND	Tierra de alimentación
2	+5Vcc	Tensión de alimentación de + 5V
3	DATA	Salida de información (humedad relativa, RH, y temperatura)

Tal y como se muestra en la figura, se utiliza el conector de expansión de la tarjeta BASIC I/O V2 para conectarlo con el controlador Arduino. El sensor trabaja con una alimentación de +5Vcc y emplea una única línea, DATA, para la transferencia de datos con el Arduino. Se conecta con la señal SDA/A4/D18 que se emplea como línea de entrada digital. Observa que el módulo se alinea con la patilla 1 quedando libres las cuatro patillas de la derecha de ese conector de expansión.

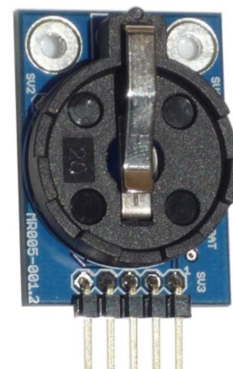


3.3 – Módulo MR005-001.3

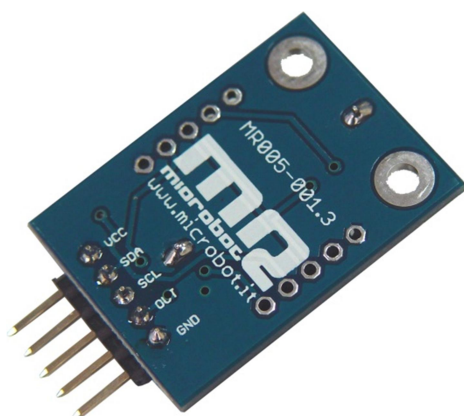
Es un módulo basado en el circuito integrado DS1307 de Dallas Semiconductor que implementa todas las funciones de un reloj/calendario en tiempo real más una pequeña área de memoria RAM. Está fabricado por la firma Italiana Microbot, www.microbot.it.

Sus características más relevantes son:

- Reloj en tiempo real con segundos, minutos, horas, día de la semana, día del mes, mes y año.
- Ajuste automático de años bisiestos hasta el año 2099.
- 52 bytes de memoria RAM alimentada por pila o batería externa (opcional tipo CR2032).
- Interface I2C con el controlador Host o Master.
- Señal de salida de onda cuadrada programable.
- Detección de fallo de alimentación y conmutación automática con la pila o batería externa opcional.
- Alimentado desde la pila o batería opcional externa (tipo CR2032) el consumo es menor de 500 nA. El reloj sigue su curso y los datos de la memoria RAM se mantienen.

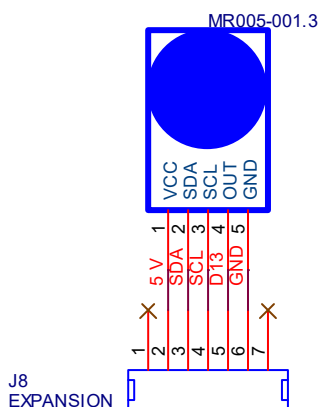


En la figura y en la tabla adjunta tienes una descripción de las patillas de conexión:



PIN	NOMBRE	DESCRIPCION
1	+5Vcc	Tensión de alimentación de + 5V
2	SDA	Línea de datos del bus I2C
3	SCL	Línea de reloj del bus I2C
4	OUT	Salida de señal de onda cuadrada
5	GND	Tierra de alimentación

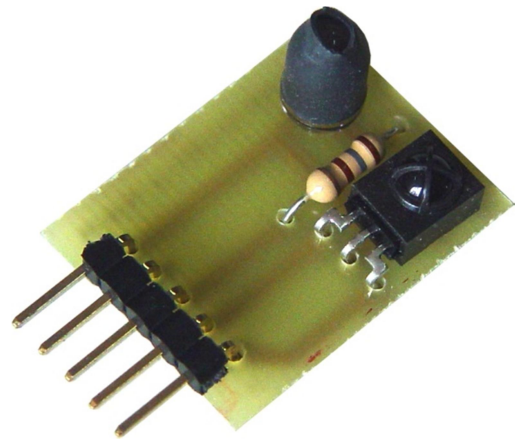
Como puedes apreciar en la figura, se utiliza el conector de expansión de la tarjeta BASIC I/O V2 para conectarlo con el controlador Arduino. Quedan libres las dos patillas extremas de dicho conector: GND y 3.3 V.



3.4 – Diseños propios

Sobre el conector de expansión se puede insertar cualquier tipo de circuito electrónico diseñado según las necesidades. A fin de cuentas dicho conector transporta las señales de alimentación de +5 V, GND y 3.3V así como tres señales de E/S de propósito general: SDA/A4/D18, SCL/A5/D19 y D13.

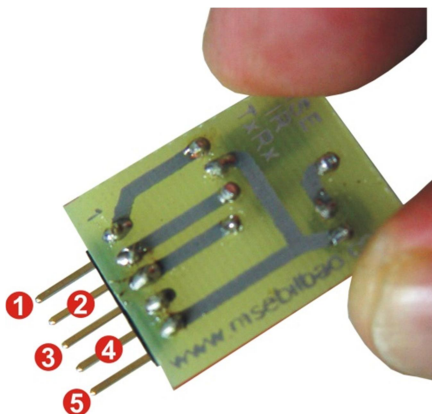
En la figura se muestra el caso de un sencillo circuito para la transmisión/recepción de luz infrarroja (IR).



Sus características más relevantes son:

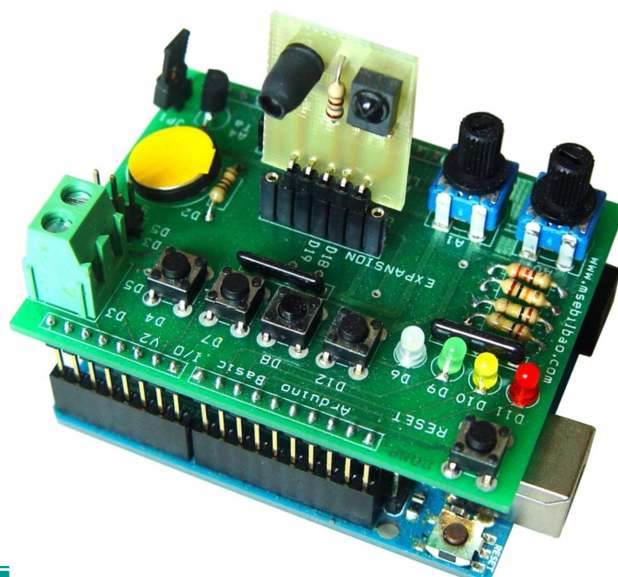
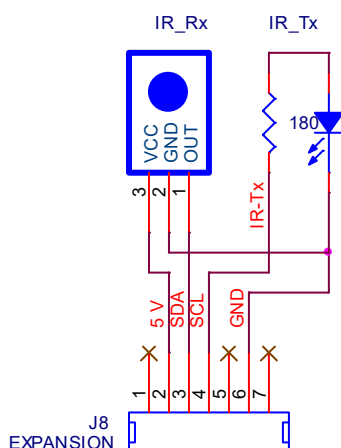
- Alimentación: 5 V
- Led emisor de infrarrojos de 950 nm.
- Receptor/detector de infrarrojos con filtro 38 KHz

En la siguiente figura, junto con la tabla adjunta, se muestra la situación y descripción de las patillas.

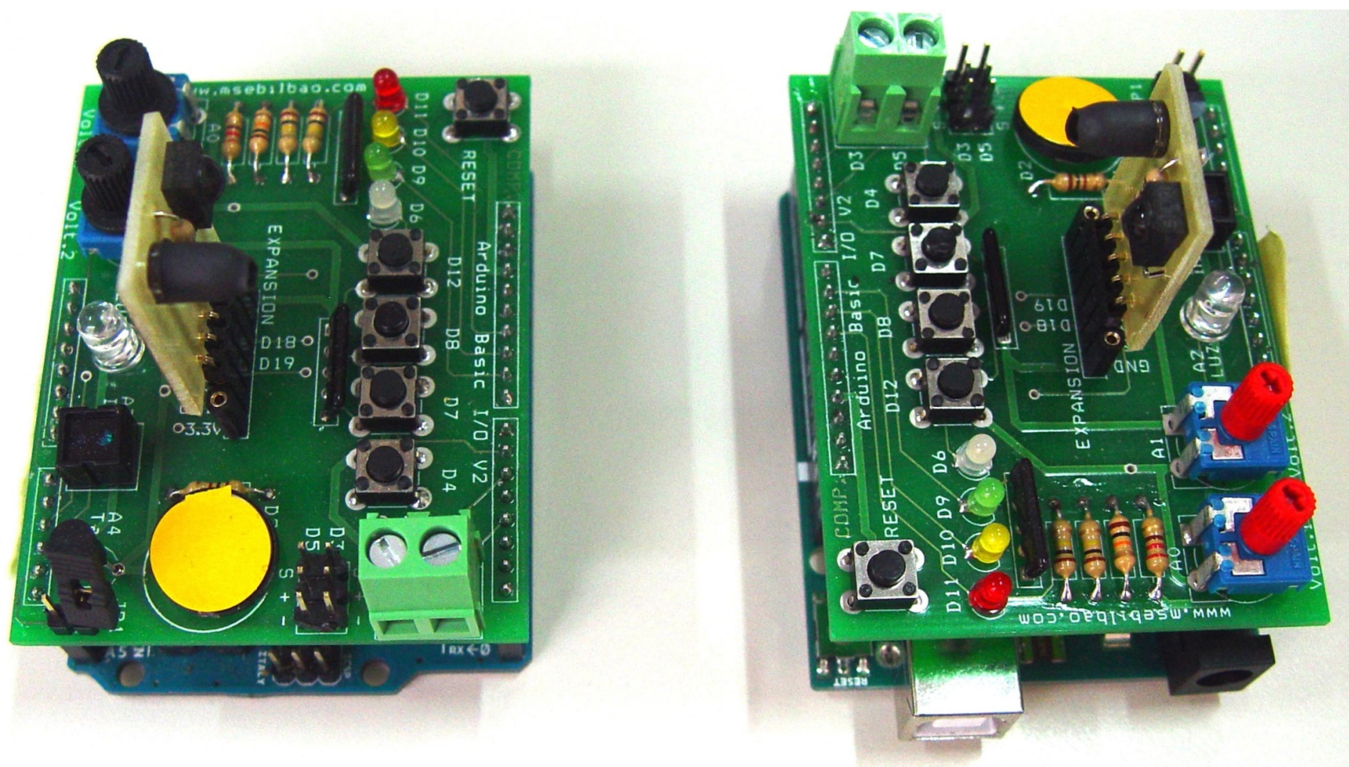


PIN	NOMBRE	DESCRIPCION
1	+5Vcc	Tensión de alimentación de + 5V
2	OUT	Señal de salida del detector con los datos recibidos vía IR (Ir Rx)
3	Ir Tx	Señal de transmisión de datos IR
4	NC	Sin conexión
5	GND	Tierra de alimentación

En la siguiente figura puedes ver sus conexiones con el conector de expansión de la tarjeta BASIC I/O V2, así como la inserción sobre la misma.



Este sencillo módulo permite realizar proyectos y aplicaciones basadas en la transmisión/recepción de luz infrarroja para detección de obstáculos, control remoto, transferencia de datos, etc... Como se muestra en la figura podemos comunicar vía IR dos controladores Arduino UNO dotados de la tarjeta BASIC I/O y de este módulo MSE-IR-TxRx, y enfrentados entre sí.



4 – EL SOFTWARE MyOpenLab

Gracias a los potentes ordenadores actuales, capaces de ejecutar los programas y aplicaciones más increíbles, es posible simular hoy en día todo tipo de sistemas: mecánicos, eléctricos, electrónicos, hidráulicos, etc... Por ejemplo, los arquitectos pueden diseñar y calcular una estructura en el ordenador. Luego pueden simular someterla a diferentes situaciones límite y probarla sin necesidad de construirla físicamente.

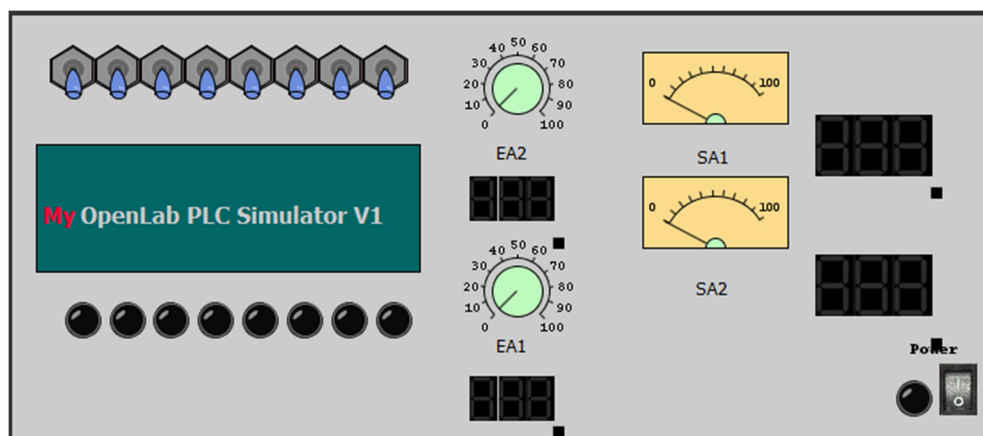
Lo mismo con los sistemas eléctricos o electrónicos. Con el programa informático adecuado pueden dibujar un circuito en el ordenador y luego someterlo a diferentes estímulos y condiciones de trabajo simuladas, para poder evaluar su correcto funcionamiento.

En cualquier caso, a la hora de la verdad, siempre hay que terminar construyendo físicamente esa estructura, ese circuito electrónico o ese sistema mecánico.

En el campo de la electrónica y los sistemas programables (p.e. los microcontroladores) existen multitud de programas informáticos capaces de hacer esas tareas. Por mencionar a uno muy conocido podemos citar a **Proteus**. Es capaz de simular prácticamente cualquier tipo de circuito e incluso simular la ejecución de programas en diferentes microcontroladores como por ejemplo Arduino. Otro muy conocido y empleado en el plano profesional/industrial puede ser el **Lab View**.

4.1 - ¿Qué es MyOpenLab?

- Es un entorno orientado a la simulación y modelado de sistemas físicos.
- Actualmente está disponible en Alemán, Inglés y Castellano.
- Se trata de una aplicación escrita en Java que se puede ejecutar en plataformas Windows y Linux tanto en 32 como en 64 bits.
- Permite el diseño de instrumentos y máquinas virtuales (VM).
- El control de la simulación así como la presentación de los resultados se realiza mediante bloques funcionales capaces de manejar todo tipo de datos, imágenes y sonido.
- Mediante un sencillo interface serie puede interaccionar con el mundo físico, por ejemplo con Arduino.
- En el plano educativo puede competir con otras herramientas de programación visual como es el Lab View.
- Idea original, desarrollo, programación y mantenimiento de **Carmelo Salafia**.
- Se puede descargar gratuitamente desde www.myopenlab.de
- Para aquellos que quieran profundizar en su funcionamiento y empleo hay disponible un curso on line en castellano que se celebra periódicamente en el campus virtual www.campustecnologicovirtual.com



MyOpenLab

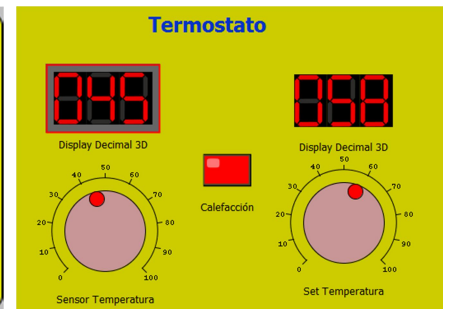
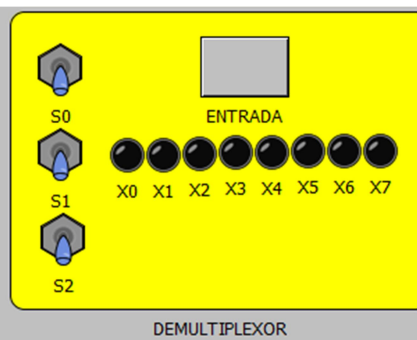
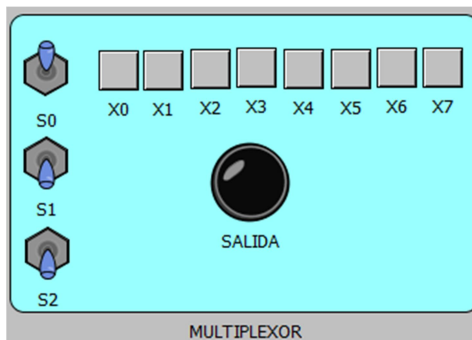
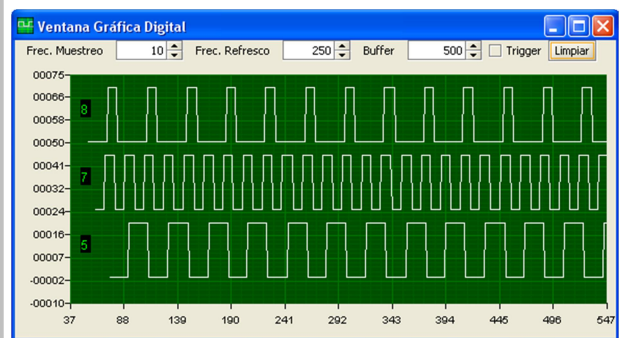
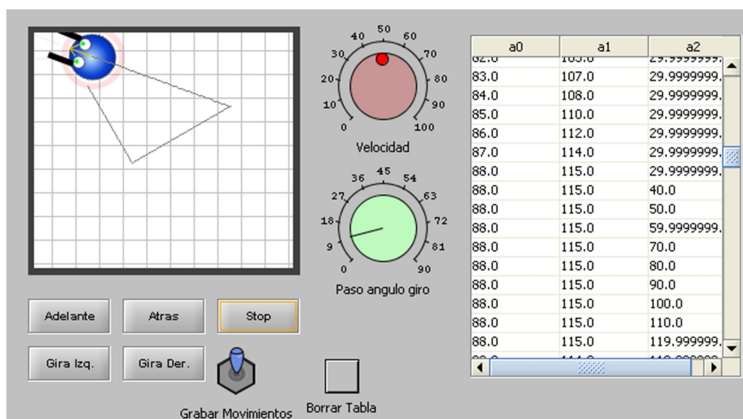
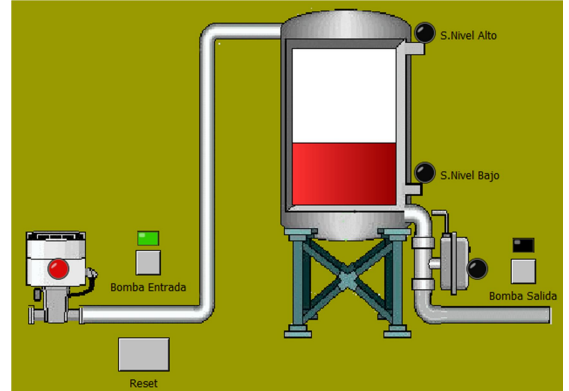


BASIC I/O

4.2 – Aplicaciones

En el campo de la enseñanza MyOpenLab es una herramienta potente y sencilla de utilizar. Permite una excelente introducción al concepto de lo que es una “máquina virtual” (VM), al manejo de componentes e instrumentos virtuales, la interacción entre ellos, y también a otras herramientas profesionales como Lab View.

- Simulación de circuitos electrónicos.
- Admite programación mediante diagramas de flujo.
- Simulación de instrumentos, automatismos y periféricos.
- Modelado de fenómenos físicos
-

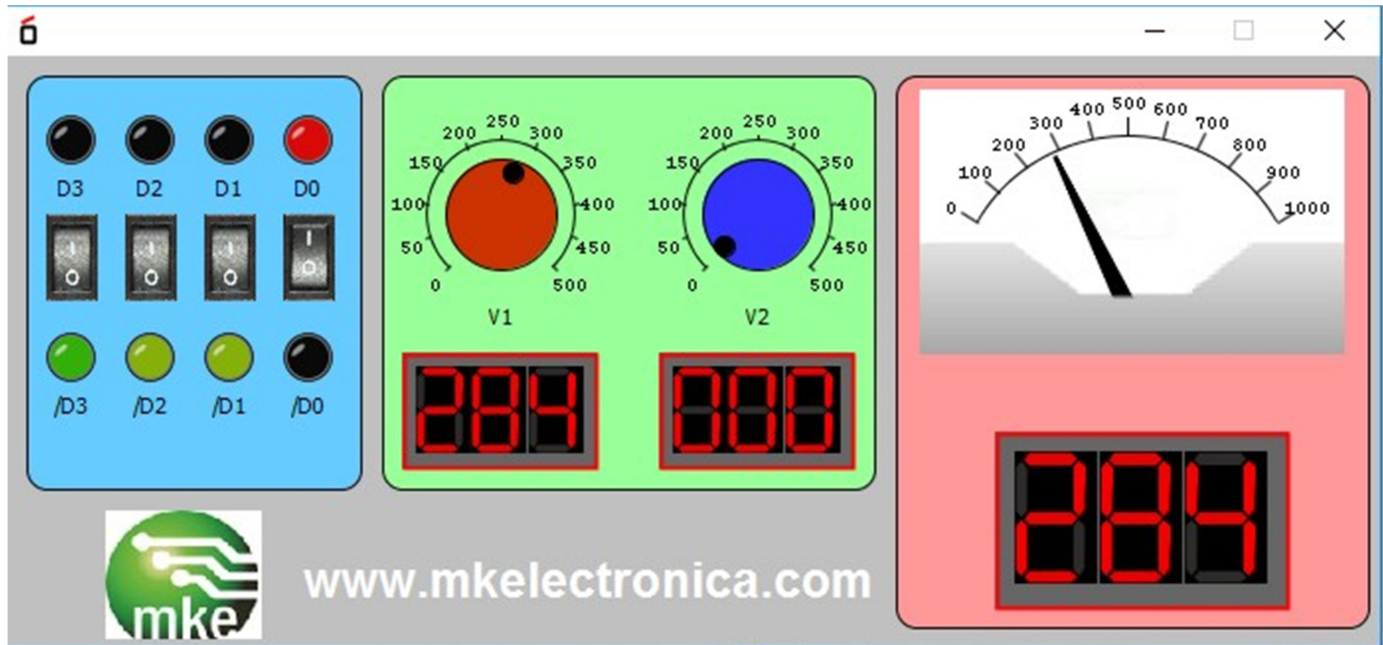


4.3 – El panel frontal

El trabajo con MyOpenLab gira básicamente en torno a dos paneles o áreas de trabajo. Uno de ellos es el panel frontal.



En él disponemos de una gran colección de objetos en forma de periféricos, instrumentos, visualizadores, herramientas gráficas y accesorios con los que construir el aspecto “visible” de una “máquina virtual” (VM) como la de la figura de ejemplo.



Cada objeto que colocamos en ese panel frontal tiene sus propios atributos que podemos modificar según nuestras necesidades funcionales y estéticas:

- Tamaño del objeto
- Colores tanto del fondo como del perfil del objeto.
- Posición del objeto en nuestra máquina
- Rangos, escalas, valores máximos, mínimos, etc..

4.4 – El panel de circuito

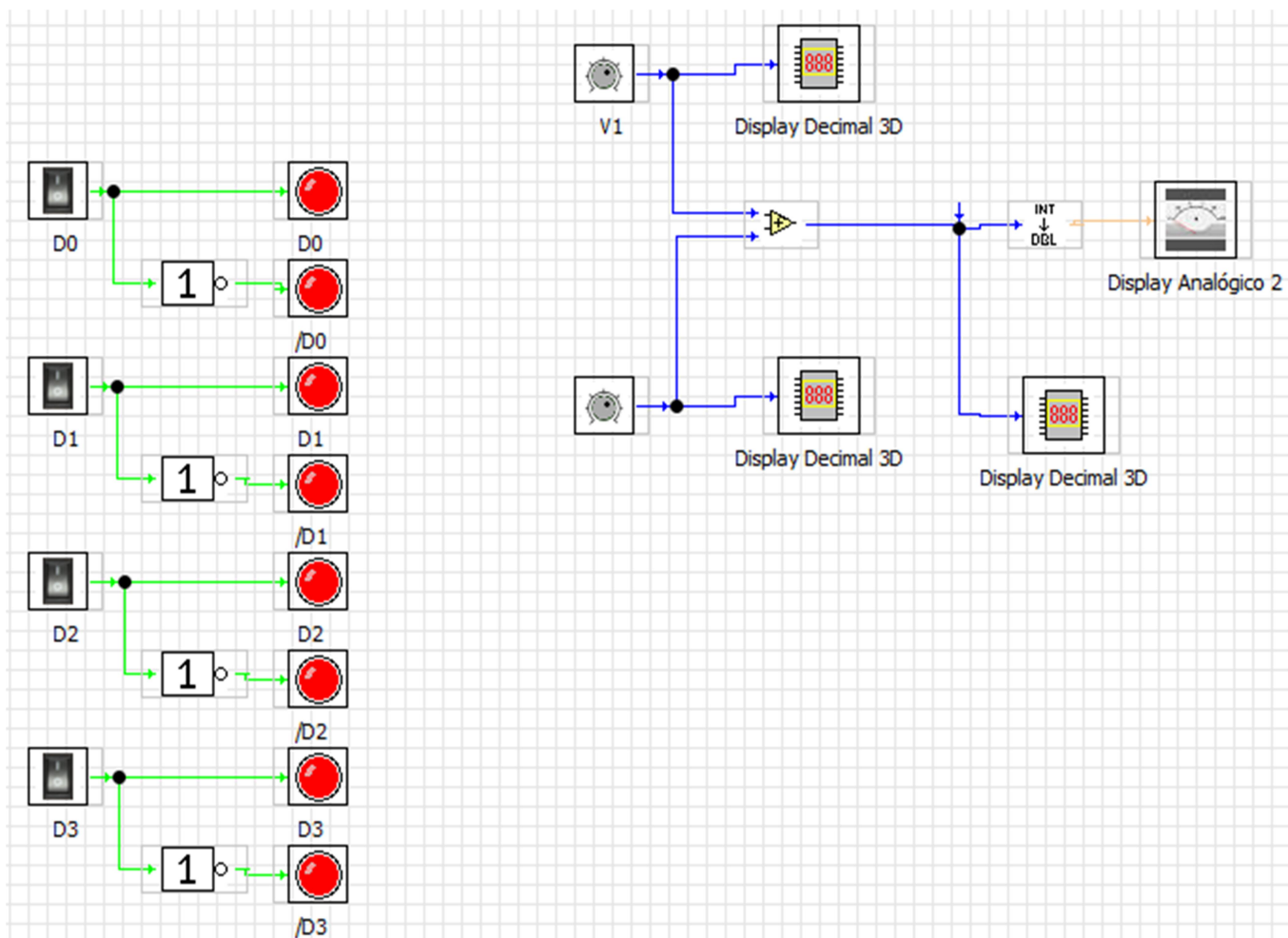
Es el segundo de los paneles de trabajo. Sobre él se desarrolla la parte “oculta” de la “máquina virtual”, es decir, la parte funcional. En este panel se realizan las conexiones de los objetos empleados en el panel frontal anterior y se establecen los algoritmos de funcionamiento de la máquina.



Dispone de herramientas con las que puedes hacer:

- Operaciones lógicas y con bits
- Operaciones aritméticas y tratamiento de caracteres
- Elementos analógicos y gestión de ficheros
- Operadores de comparación y toma de decisiones
- Gestión de color, imágenes y sonido.
- Operaciones con vectores y matrices
- Programación mediante diagramas de flujo
- Interfaces con el exterior

- etc...



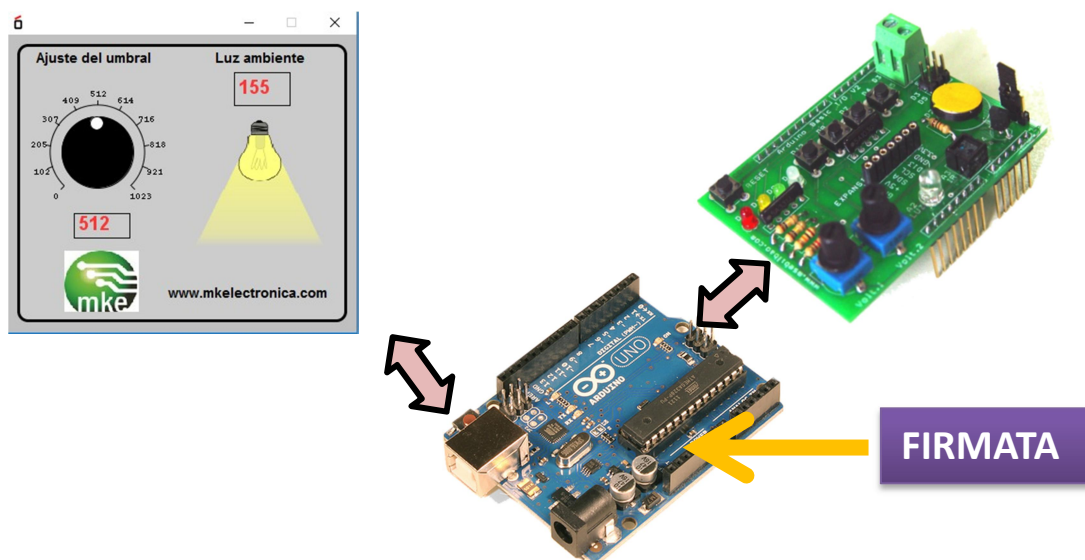
4.5 - Conexión con el mundo físico

Aquí queríamos llegar. El mundo “virtual” no sirve de gran cosa si no se puede enlazar con el mundo “real”. Con MyOpenLab es posible conectar y gestionar dispositivos y tarjetas electrónicas externas. En nuestro caso podemos utilizar MyOpenLab para controlar los periféricos y dispositivos disponibles en nuestra tarjeta BASIC I/O cuando se conecta a una controladora Arduino UNO o compatible.

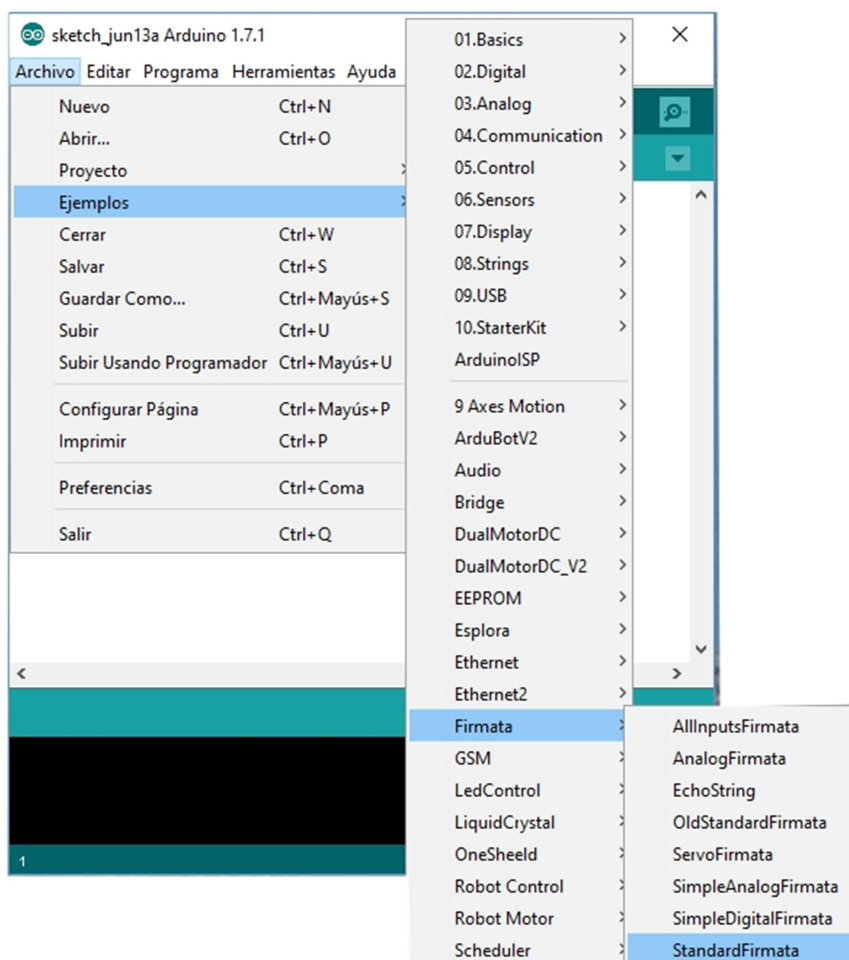
Según el esquema de la siguiente figura, la tarjeta BASIC I/O objeto del presente documento se conecta con la tarjeta controladora Arduino UNO o compatible tal y como se ha explicado en los apartados previos. A esta controladora se le graba el programa o sketch “**Firmata**”.

Básicamente su ejecución por parte de Arduino consiste en leer el estado actual de todas sus entradas y transmitirlo al exterior vía comunicación serie. De igual forma, desde el exterior se puede determinar el nuevo estado de las salidas del controlador.

Pues bien, MyOpenLab puede recoger esa información procedente de las entradas del Arduino, procesarlas de acuerdo a un algoritmo y generar unas determinadas señales de salida. En otras palabras, hay una interacción entre el mundo virtual (MyOpenLab) y el mundo real (Arduino + BASIC I/O).



El programa “**Firmata**” se incluye en los ejemplos que se instalan junto con el IDE de Arduino y lo puedes encontrar en: **Archivo → Ejemplos → Firmata → StandarFirmata**. Grábalo como si fuera cualquier otro programa típico de Arduino.



4.6 - Ejemplos

Par ofrecer una visión más precisa de la relación entre el mundo virtual (MyOpenLab) y el mundo real (Arduino UNO + BASIC I/O) vamos a ver dos sencillos ejemplos.

4.6.1 E3 Control de BASIC I/O

Desde MyOpenLab se controlan todos los periféricos contenidos en la tarjeta BASIC I/O V2. En el panel frontal se representan los diferentes dispositivos.

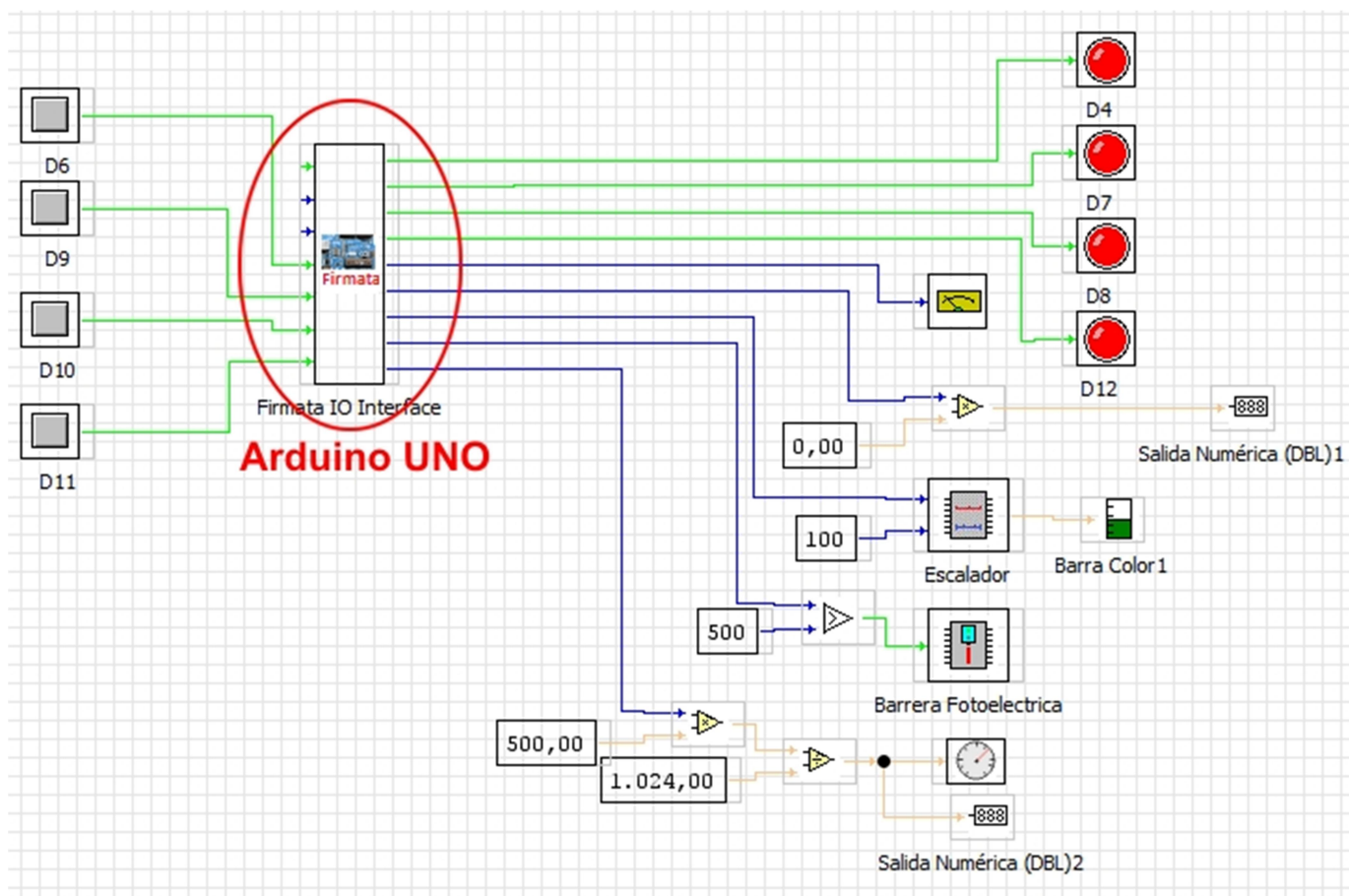


Con los pulsadores D6, D9, D10 y D11 vamos a controlar ON/OFF los leds de salida de la tarjeta BASIC I/O. De igual manera el estado de los pulsadores de entrada D4, D7, D8 y D12 de la tarjeta se visualiza sobre los leds virtuales de MyOpenLab.

Por su parte el valor de los dispositivos y sensores analógicos de la tarjeta BASIC I/O se visualizan mediante diferentes instrumentos virtuales de MyOpenLab:

- El valor del potenciómetro del canal analógico A0 se visualiza mediante un medidor analógico de aguja.
- El valor del potenciómetro conectado en A1 se visualiza de forma numérica convertida a tensión (0 a 5V).
- El sensor de luz en A2 emplea una barra gráfica para visualizar su valr.
- El sensor infra rojo de A3 se representa mediante una barrera foto eléctrica que se abre o cierra según haya o no reflexión de luz.
- El sensor de temperatura en A4 se visualiza sobre un medidor de aguja y también de forma numérica.

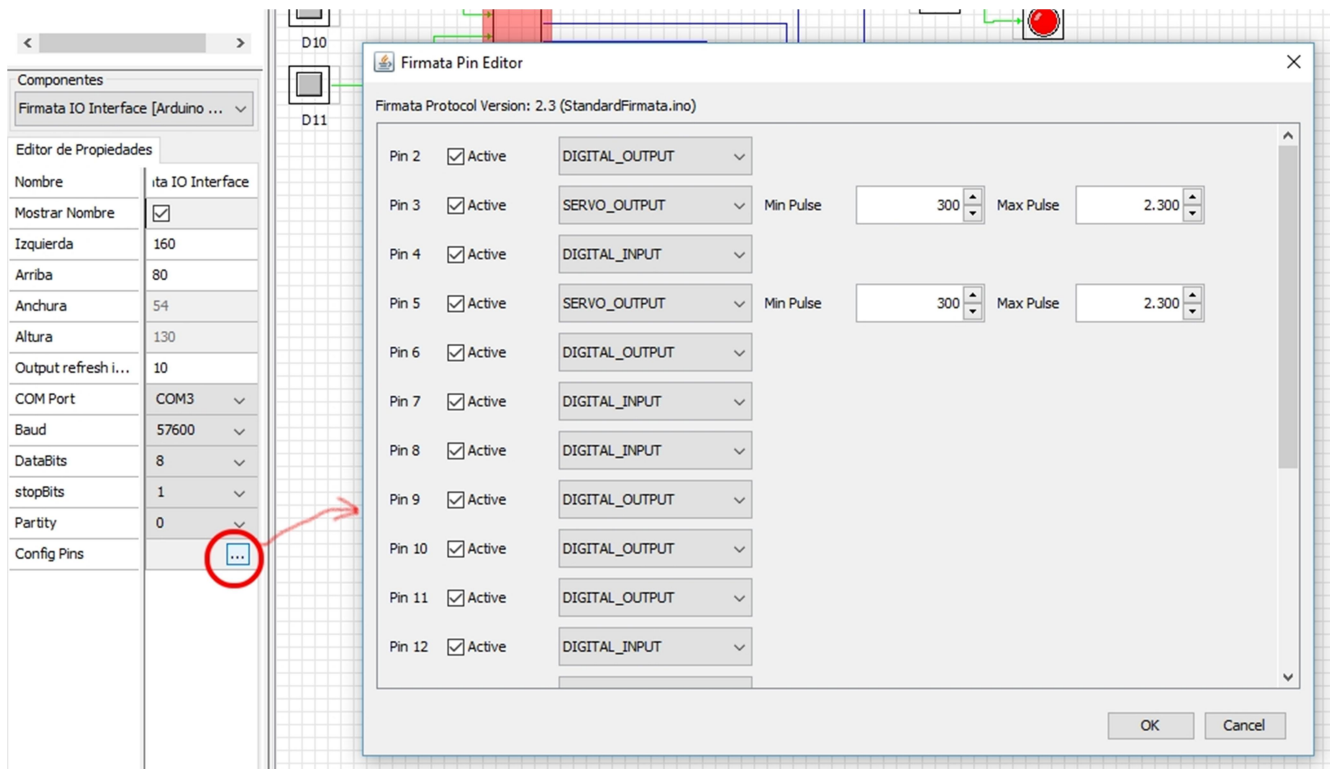
En el panel de circuito se establecen las conexiones y algoritmos de funcionamiento.



Remarcado en un círculo rojo se representa al componente Arduino UNO (o compatible) al que se le supone cargado con el programa **StandarFirmata**.

- Los pulsadores virtuales D6, D9, D10 y D11 se conectan con esas mismas líneas del controlador donde a su vez están conectados los leds reales blanco, verde, ámbar y rojo de la tarjeta BASIC I/O. Estos se activan o no según el estado de esos pulsadores.
- Los leds virtuales D4, D7, D8 y D12 se conectan con esas mismas líneas del controlador que a su vez están conectadas con los pulsadores reales de la tarjeta BASIC I/O. Cuando se activa cualquier pulsador se iluminará el correspondiente led virtual.
- El valor analógico A0 (potenciómetro de BASIC I/O) se conecta directamente con el medidor analógico de aguja. Representará el resultado de la conversión entre 0 y 1023.
- El valor analógico A1 (potenciómetro de BASIC I/O) se multiplica por la constante 0.00488 para obtener una tensión equivalente entre 0 y 5V. Dichas tensión se visualiza de forma numérica.
- El valor analógico del canal A2 (sensor de luz en BASIC I/O) se escala y redondea a un valor comprendido entre 0 y 100 y luego se visualiza sobre la barra gráfica virtual de MyOpenLab.
- El valor analógico del canal A3 (sensor infra rojo en BASIC I/O) se compara con 500. Si es mayor se activa la barrera fotoeléctrica virtual. En caso contrario la barrera permanece desactivada.
- El canal analógico A4 (sensor de temperatura de BASIC I/O) se multiplica por 500 y al resultado se divide entre 1024. Se obtiene así la temperatura que se visualiza tanto en un indicador de aguja como en uno numérico.

Cuando se selecciona el componente Arduino UNO en el Panel de Circuito, a la izquierda de la pantalla se deben configurar una serie de propiedades como el puerto COM de comunicaciones con Arduino UNO y la velocidad de la misma en Baudios. También se deben configurar las patillas o pines, indicando cuáles se emplean y si son entradas o salidas.



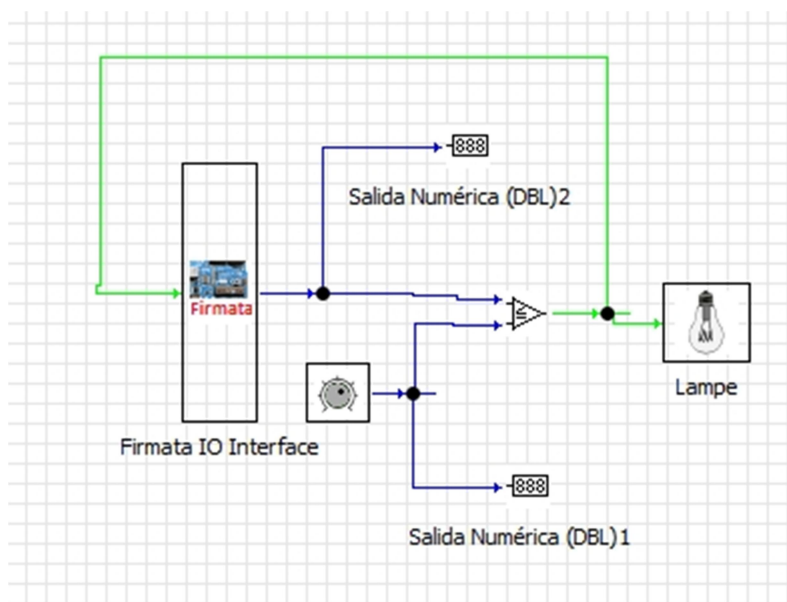
4.6.2 – E4 Control Alumbrado

En el panel frontal se dispone de un indicador numérico que representa la cantidad de luz visible que incide sobre el sensor de luz de BASIC I/O conectado en la entrada analógica A2. También hay un potenciómetro con su correspondiente indicador numérico que permite ajustar el umbral de luz deseado. Por defecto está ajustado a 512.

Un objeto tipo bombilla se activará cuando la luz que incide sobre el sensor esté por debajo del valor de umbral seleccionado con el potenciómetro. La bombilla real se conecta con la patilla D6 donde se encuentra el led blanco en la tarjeta BASIC I/O.



En el Panel de Circuito se establecen las conexiones y condiciones de funcionamiento. El valor analógico presente en A2 (sensor de luz de BASIC I/O) se conecta con un visualizador numérico y con una de las entradas del comparador. La otra entrada de éste se conecta con el potenciómetro de ajuste del umbral y con otro visualizador numérico.



Si el valor del sensor de luz es igual o menor que el valor que introduce el potenciómetro, se activa la lámpara virtual y también la patilla D6 de Arduino UNO, donde físicamente se encuentra el led blanco de la tarjeta BASIC I/O.