SISTEMAS DE CONTROL AUTOMÁTICOS Y ROBÓTICA
3º ESO

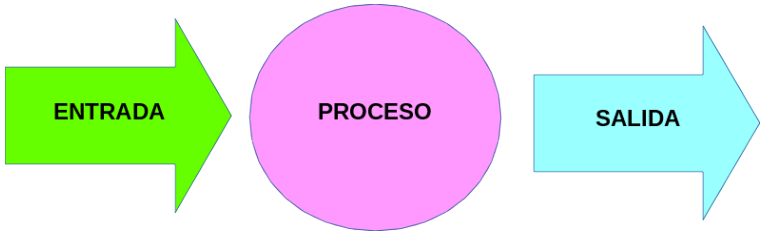
1. Introducción.

Un sistema de control automático o automatismo, es un conjunto de elementos técnicos que unidos son capaces de realizar una series de acciones para resolver un problema sin intervención humana.

Los que vamos a usar en nuestro curso son programados, que significa que el automatismo funcionará dependiendo de un programa que le haremos a medida de la función a realizar.

2. Elementos de un sistema de control.

Nuestro sistema de control va a contener básicamente tres bloques de dispositivos:



- Elementos de proceso (control en sí): son los que se van a programar, recibirán datos de los elementos de entrada para decidir que realizar con los elementos de salida. En nuestro caso es *Arduino*.
- Elementos de entrada: serán sensores que se le pueden conectar a nuestro sistema de control, pueden ser: pulsadores, interruptores, finales de carrera, ntc, ldr, etc.
- Elementos de salida: van a ser los actuadores, los que va a decidir nuestro programa poner en funcionamiento o parar, como: diodos leds, zumbadores, motores, etc.

3. Elementos de control programado.

En nuestro caso vamos a utilizar una placa de control programable que se llama **Arduino**, la cual tiene un uso muy extendido por varias cuestiones:

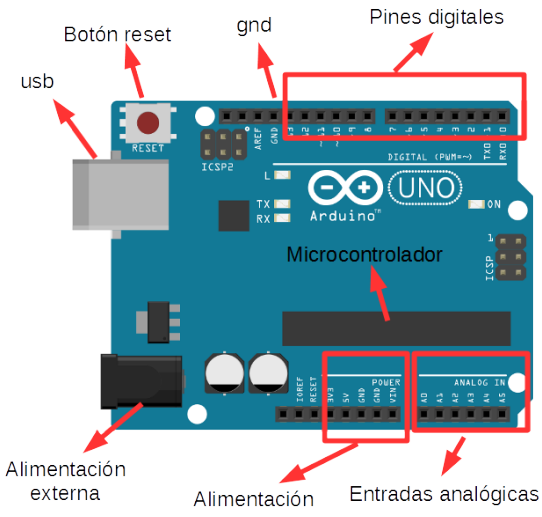
- Hardware libre: los diseñadores han puesto a disposición de todo el mundo los circuitos físicos de la placa para poder fabricarla.
- Software libre: el software que controla la placa también está puesto a disposición de todo el mundo para compartirlo e incluso modificarlo.
- Comunidad: hay una gran cantidad personas compartiendo, proyectos, código y soluciones por internet.
- Precio: muy económico.
- Simplicidad: se pueden realizar soluciones de una manera sencilla.

3.1 Partes de arduino.

Algo es digital cuando sólo puede tomar dos valores (1 o 0, encendido o apagado).

Pines digitales (0 a 13): sirven para conectar tanto salidas digitales (diodos led, motores, zumbadores, etc), como entradas digitales (interruptores, pulsadores, finales de carrera, etc).

Estos pines cuando se usan como salidas podemos decirles que estén conectados o no, que tengan corriente o no. Cuando se usan como entrada detectan cuando los elementos de entrada están cerrados o no.



Pines gnd (ground): sirven para cerrar el circuito, son el polo negativo de nuestra alimentación. Tenemos un gnd en la fila de los pines digitales y dos gnd más en la zona de alimentación (power).

Entradas analógicas: Sirven para introducir en el sistema datos que no toman dos valores, pueden tomar muchos, como temperaturas (con ntc), luz (con ldr), etc. Será objeto de estudio el curso que viene.

Microcontrolador: Es el dispositivo que recibe el programa y hace que se ejecute, así como controla todos los elementos de la placa.

Usb: es un conector usb que sirve para dar alimentación a la placa y además comunicarse con el ordenador.

Alimentación externa: conector que sirve para alimentar arduino con un elemento externo (cargador, pila de 9v, etc).

4. Software gráfico para programar arduino. Scratch for Arduino (S4A).

Scratch for Arduino (de ahora en adelante S4A) es una variante de Scratch que sirve para programar de forma gráfica Arduino.

El software es libre y se puede descargar de la web oficial del proyecto.

Para que Arduino funcione con S4A debe instalarse primero este programa en la placa Arduino con el software de Arduino.

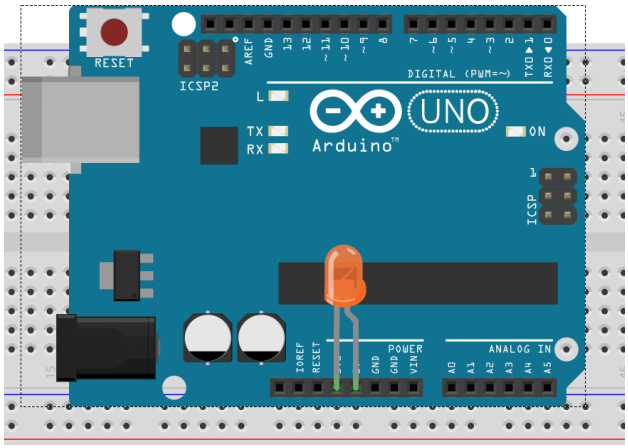
La gran ventaja de S4A es que es fácil de programar, y presenta como inconveniente que tiene que estar conectado nuestro arduino permanentemente al ordenador para que se comunique con el programa que está en el mismo.

5. Controlar salidas digitales (leds) con S4A.

En nuestro paso vamos a empezar a controlar leds, antes de nada debemos aprender como conectar los mismos a nuestro arduino.

Práctica 1. Arduino como pila

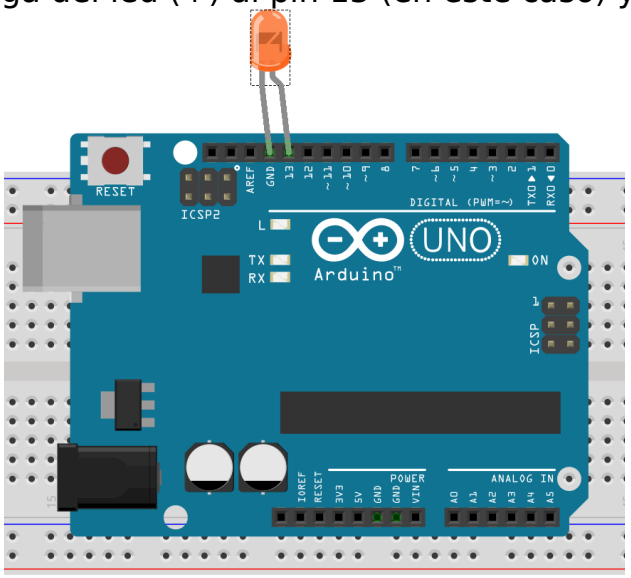
La primera práctica va a ser utilizar nuestro arduino como pila y encender un diodo led con el mismo. Para ello usar la zona de alimentación de nuestro arduino, conectando la patilla larga de nuestro led (+) con el pin de 5V, y la patilla corta (-) con el gnd, como indica la figura.



Práctica 2. Control de led.

Se trata de encender y apagar un led conectado al pin 13. El pin deberá encenderse un tiempo de un segundo y apagarse durante un segundo, esto deberá repetirlo por siempre.

Antes de nada vamos a proceder a conectar el led sin la placa conectada al ordenador (siempre hay que conectar con la placa sin alimentación). Para ello conectamos la pata larga del led (+) al pin 13 (en este caso) y la corta (-) al gnd.



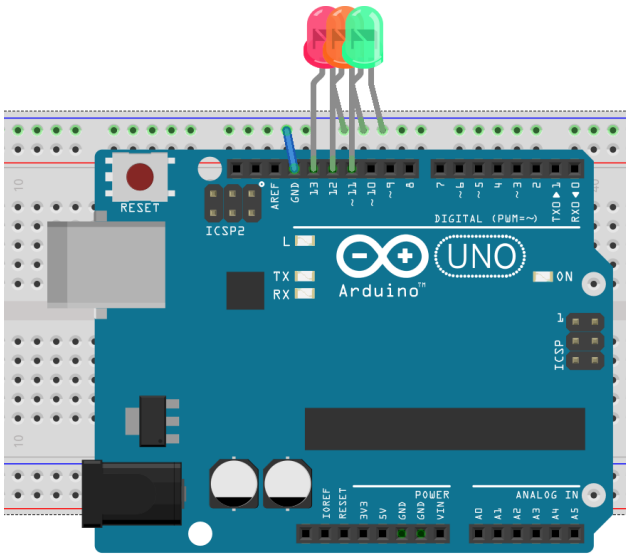
Ahora procedemos a abrir S4A (con usuario-usuario) con la placa conectada al ordenador, y programamos las siguientes instrucciones, en las cuales tendremos que utilizar instrucciones del bloque de control (amarillas) e instrucciones del bloque de movimiento (azules).

Cuando usamos la orden “digital 13 encendido” lo que estamos diciendo es que salga corriente de 5V por el pin 13 (+) que cerrará el circuito con el gnd (-), y cuando usamos la orden “digital 13 apagado” cortamos esa corriente.

Comentar que S4A (en la versión más moderna) puede usar sólo los pines digitales 10, 11, 12 y 13.

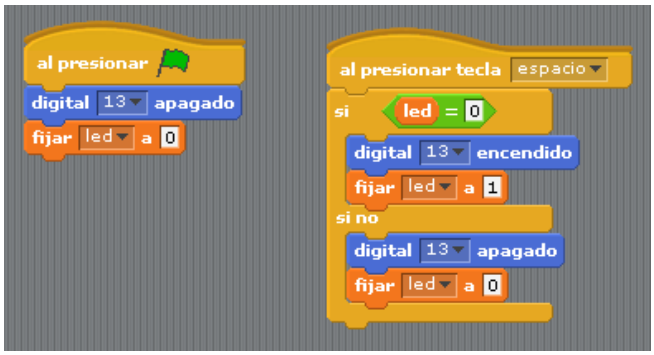
Práctica 3. Semáforo simple

Se trata de realizar el control de un semáforo simple, para ello tendremos que conectar a nuestro arduino tres leds (rojo, ámbar y verde). El verde se debe encender durante 5 segundos, se apagará, se encenderá el ámbar durante 1 segundo, se apagará, y se encenderá el rojo durante 5 segundos, se apagará y se repetirá el proceso por siempre. Para conectarlos lo haremos el rojo al pin 13, el ámbar al pin 12 y el verde al pin 11. A los pines siempre se coloca la pata larga del led (+) y al gnd (-) la pata corta del led (-). Hay un pequeño problema y es que en el pin gnd no cabe más que un una patilla, por tanto tenemos que usar la placa board para solucionarlo de la forma representada en la figura.



<u>Práctica 4. Encendido y apagado de leds con teclas</u>

El objetivo de la práctica es encender un led conectado al pin 13 cuando presionas la tecla flecha izquierda, y apagarlo cuando presionas la tecla derecha.



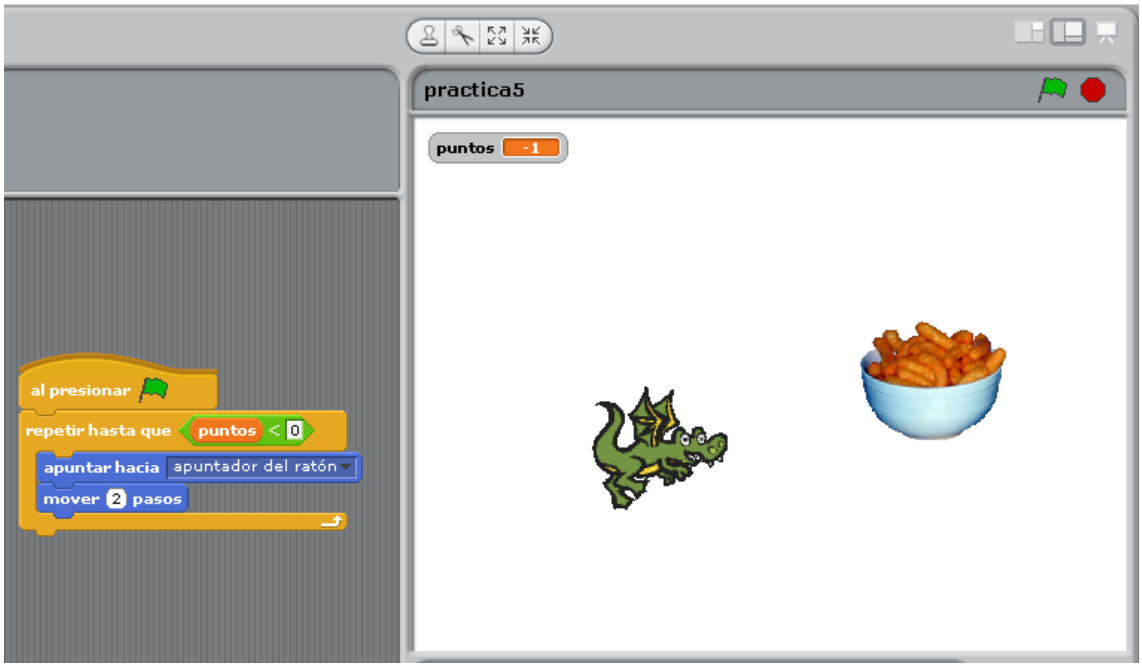
<u>Práctica 5. Apagado y encendido de led con una sola tecla</u>

Se trata de encender o apagar un led conectado al pin 13, pulsando sólo la tecla espacio. La técnica en este caso es iniciar el programa apagando el led y poniendo una variable a 0. Cada vez que se pulse la tecla espacio comprueba como estaba la variable (a 1 o a 0), enciende o apaga el led según el caso y cambia el valor de la variable.

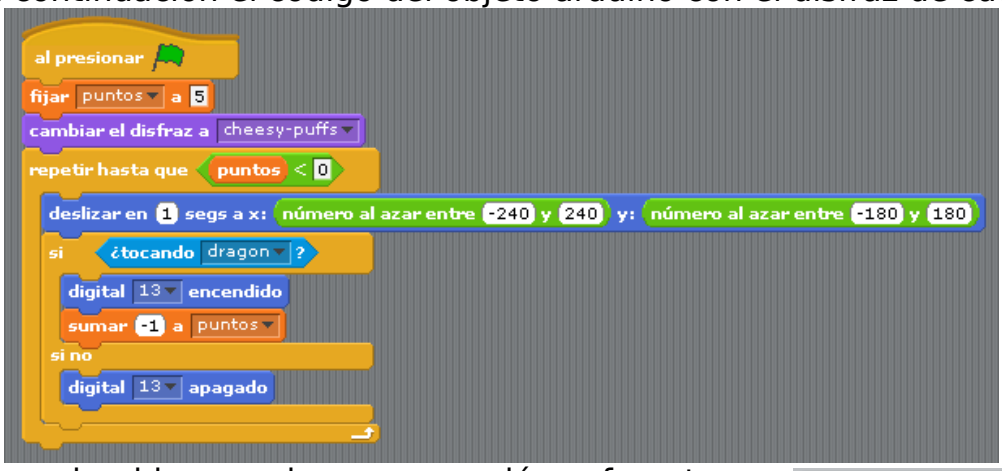
<u>Práctica 6. Objeto que cuando toca el ratón enciende led</u>

Al objeto arduino le vamos a colocar un nuevo disfraz (en nuestro caso un cuenco de gusanitos), y lo vamos a mover aleatoriamente por la pantalla, nosotros controlaremos otro objeto con el ratón (el dragón), partimos con 5 puntos y cada vez que nos toque el objeto arduino (cuenco) nos resta 1 punto y se enciende el led conectado al pin 13. La partida termina cuando tengamos puntos negativos.

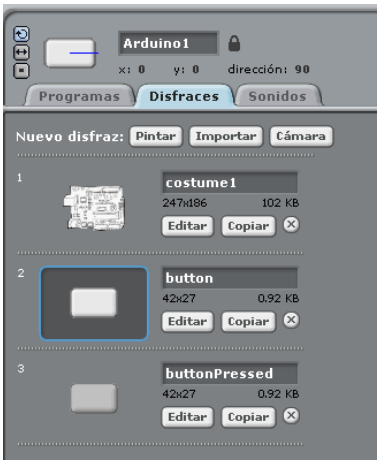
Os dejamos el código del personaje que manejamos nosotros (dragón).



Os pongo a continuación el código del objeto arduino con el disfraz de cuenco.



Comentar que los bloques de programación referentes a arduino, sólo se encuentran en los objetos arduino, en los otros no aparecen.



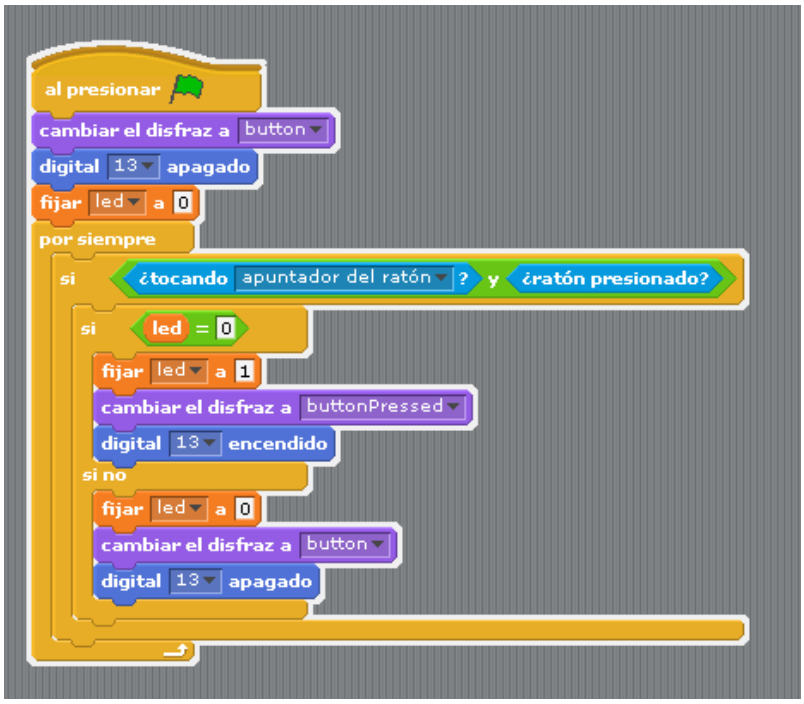
<u>Práctica 7. Control de led con botón en pantalla</u>

Se trata de un objeto arduino con dos disfraces nuevos, uno de botón pulsado y otro sin pulsar. Cuando se mantenga pulsado el botón con el ratón, un led conectado al pin 13 debe encenderse y cambiar disfraz a pulsado, en caso contrario debe apagarse y cambiar disfraz a botón sin pulsar. (Programa 7a)

Otra opción es que cambie el estado del botón y del led cada vez que se haga clic en el mismo. (Programa 7b)

Ver programas a continuación.

7a



7b

5. Controlar salidas analógicas (leds) con S4A.

Práctica 8. Cruce de semáforos

Se trata de controlar un cruce de semáforos, conectando el primer semáforo a los pines 13 (rojo) ,12 (ámbar) y 11 (verde), y el segundo semáforo a los pines 9 (rojo), 6 (ámbar) y 5 (verde).

Los semáforos tienen que estar sincronizados, y cada semáforo pasa por la secuencia de colores rojo-verde-ámbar que se repite.

El encendido de rojo o verde debe ser 6 segundos y el ámbar 1 segundo.

Hay un pequeño problema añadido y es que sólo disponemos de tres salidas digitales en S4A, por tanto tendremos que recurrir a las salidas analógicas, pines 9, 6 y 5.

Una salida es analógica cuando puede tomar muchos valores, en concreto en nuestro caso puede tomar valores entre 0 y 255. Con estas salidas puedo hacer que una led no luzca nada con valor 0, o que luzca a tope con valor 255, o bien que luzca en un valor intermedio. El bloque de instrucciones que lo controla es 

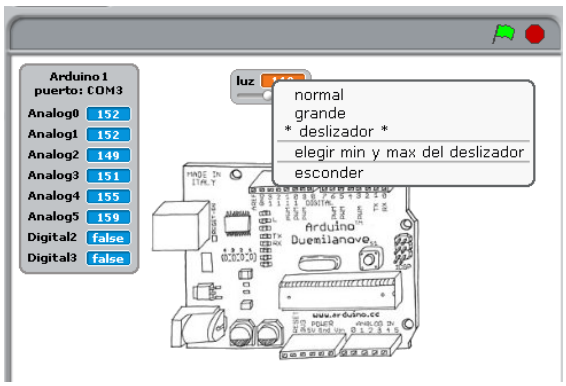
en este caso el led conectado al pin 9 está luciendo al máximo.

Nota: para conectar 6 leds es análogo (igual) a conectar 3, ver la práctica 3.

Práctica 9. Led que varia su intensidad luminosa con deslizador

Creamos una variable en nuestro caso “luz”, a la representación gráfica 

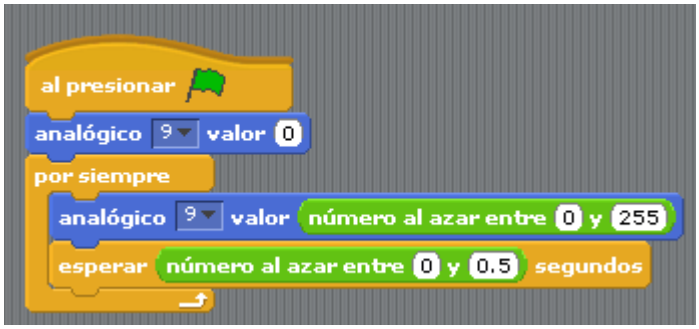
de la variable en el escenario le hacemos clic con el botón derecho del ratón y elegimos en el menú que sale la opción “deslizador” y más tarde hacemos clic de nuevo y elegimos los valores mínimo y máximo del mismo que serán 0-255.



La idea es que al mover el deslizador cambie la luminosidad del led conectado al pin 9 con el siguiente código.

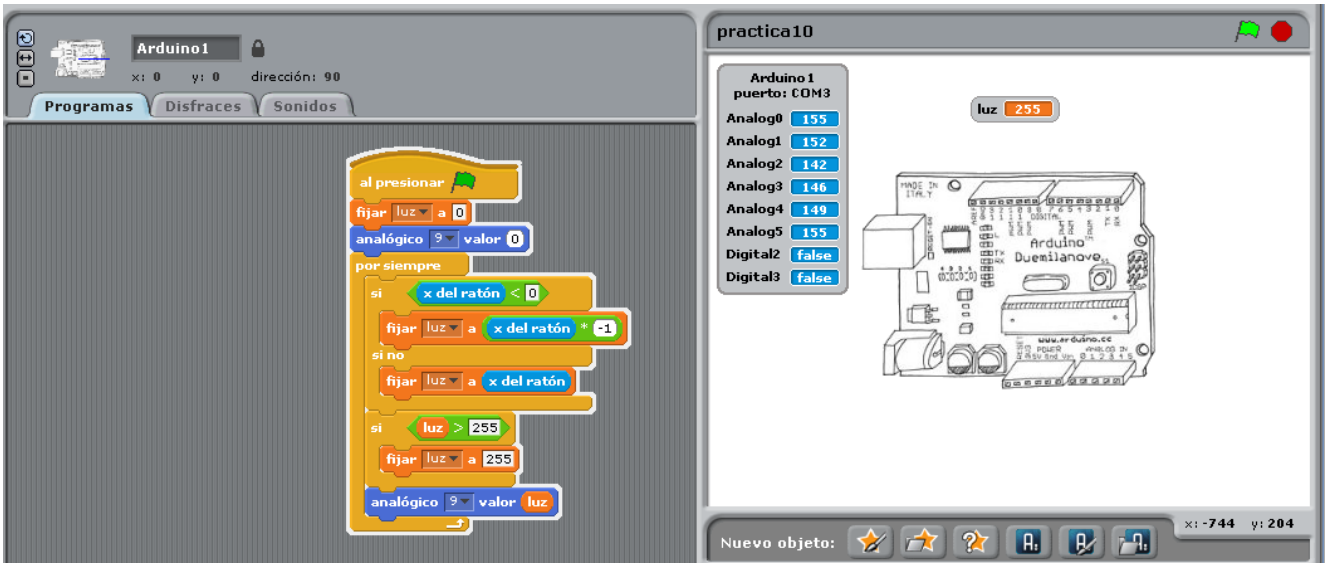
Práctica 10. Vela electrónica

Se trata de hacer que la luz de un led ámbar conectado al pin analógico 9 cambie de intensidad de forma aleatoria, además también cambiamos el tiempo que permanece encendida a esa luminosidad de forma aleatoria, con esto conseguimos una vela electrónica.



Práctica 11. Controlar luminosidad de led con ratón

Se trata que al mover el ratón a la izquierda o a la derecha del escenario la luminosidad suba, y en el centro del mismo sea 0. Ojo los valores nunca pueden ser negativos ni superar los 255, para ello presentamos la siguiente solución.



Por: *Pedro Ruiz Fernández* (formaentecnologia@gmail.com)