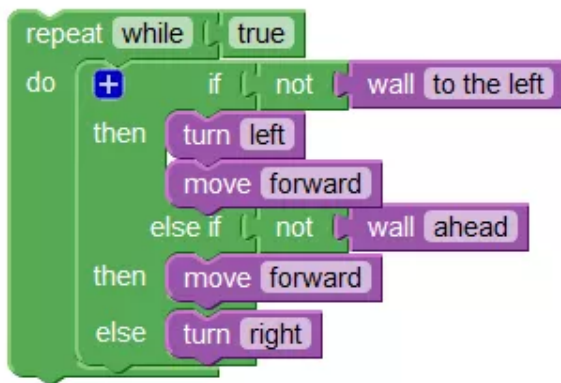


Aprendiendo Arduino

Aprendiendo a manejar Arduino en profundidad

Programación Visual con S4A

Además de la programación tradicional de Arduino, existen varios proyectos para programar Arduino de forma visual añadiendo bloques al estilo de scratch, blockly o similar.

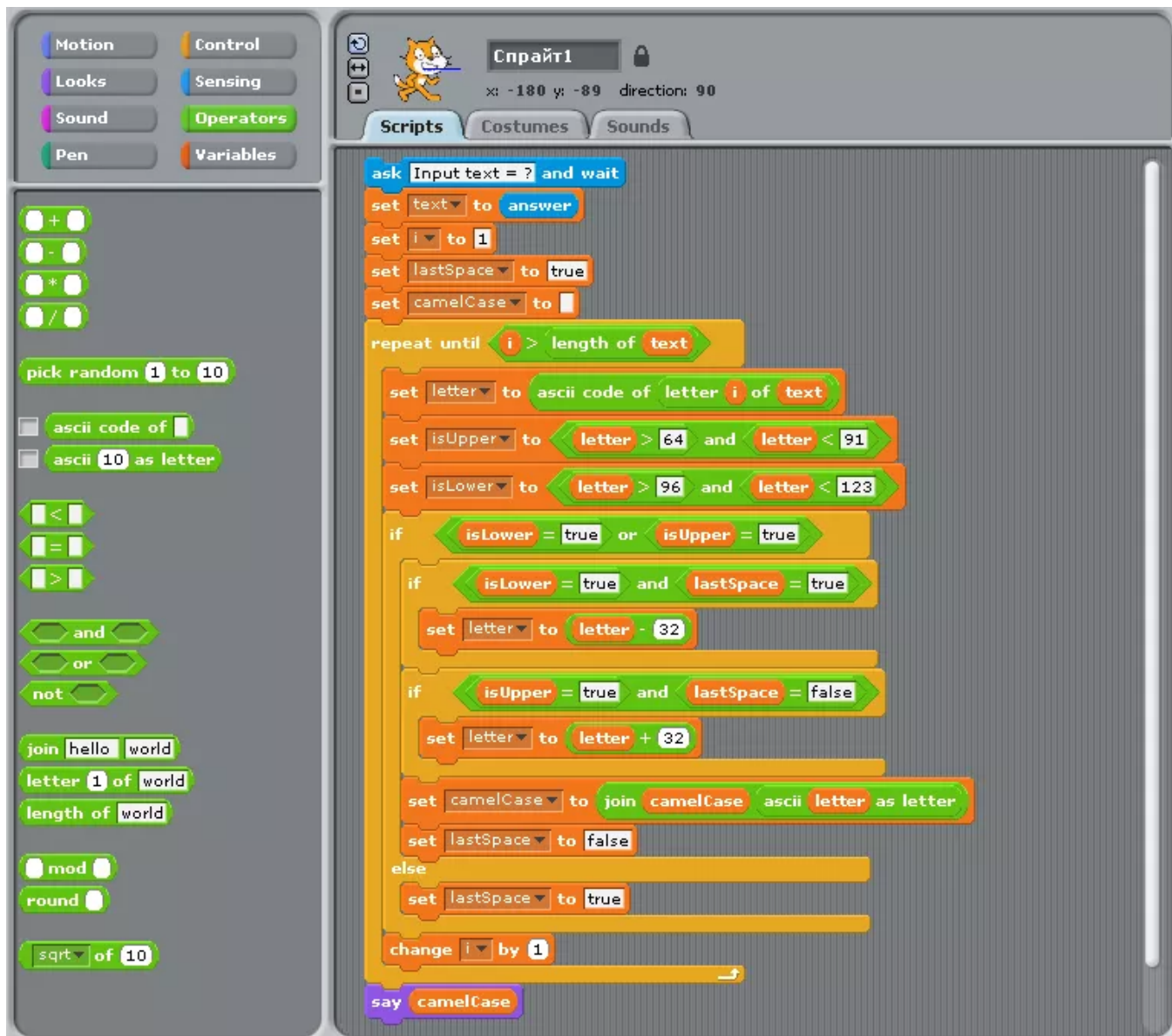


Esta programación es limitada respecto a la programación tradicional, pero puede ser una opción para iniciarse tanto niños como adultos.

Otro problema de este tipo de programación es que no está estandarizada y cada solución aunque parecida en el concepto, son distintas.

Proyectos de programación visual:

- Ardublock: <http://blog.ardublock.com/>
- Sistema de bq: <http://bitbloq.bq.com/>
- S4A es una modificación de Scratch que permite programar la plataforma de hardware libre Arduino de una forma sencilla. Proporciona bloques nuevos para tratar con sensores y actuadores conectados a una placa Arduino.
- Visuino (de pago): <https://www.visuino.com/>
- Visualino: <http://www.visualino.net/>
- mBlock: <http://www.mblock.cc/>
- Y muchos más...



S4A

S4A es Scratch for Arduino. Toda la información está en la Web: http://s4a.cat/index_es.html

S4A es una modificación de [Scratch](#) que permite programar la plataforma de hardware libre [Arduino](#) de una forma sencilla. Proporciona bloques nuevos para tratar con sensores y actuadores conectados a una placa [Arduino](#). También cuenta con un panel de sensores similar al de la PicoBoard.

Fue desarrollado por el equipo de [Smalltalk](#) del [Citilab](#) en el año 2010, y desde entonces se ha usado en todo el mundo para una gran variedad de proyectos.

Otro de sus objetivos es proporcionar una interfaz de alto nivel para programadores de [Arduino](#) con funcionalidades tales como la interacción con un conjunto de placas mediante eventos de usuario.

Tutorial de S4A: <http://recursostic.educacion.es/observatorio/web/fr/software/software-educativo/1018-monograficodesarrollos-de-scratch-para-robotica-enchanted-y-s4a?start=3>

Manual S4A: <http://vps34736.ovh.net/S4A/s4a-manual.pdf>

Primeros pasos con S4A: <https://programarfacil.com/blog/arduino-blog/s4a-primeros-pasos-con-scratch/>

Una de las limitaciones que tiene S4A es la conectividad. Los componentes deben conectarse de una manera determinada para su buen funcionamiento. Esto no impide que podamos aprender a programar con esta plataforma, solo limitará el tipo de proyectos que podemos hacer con ella.

Los diferentes componentes que podemos tratar son:

- 6 entradas analógicas (A0 a A5)
- 2 entradas digitales (2 y 3)
- 3 salidas analógicas PWM (5, 6 y 9)
- 3 salidas digitales (10, 11 y 13)
- 4 salidas digitales para conectar servomotores (4, 7, 8 y 12)

S4A es compatible con los Arduino Diecimila, Duemilanove y Uno. Puede que con otras placas funcione, pero habría que comprobarlo.

Instalar S4A en el ordenador

S4A funciona sobre los tres sistemas operativos de escritorio más populares.

Ir a la sección de descarga de http://s4a.cat/index_es.html y descargar para el Sistema Operativo correspondiente.

Para trabajar con S4A es necesario instalar en el Arduino que usemos el firmware que nos proporciona S4A.

Proceso:



Buen tutorial de instalación: <https://www.prometec.net/instalacion-scratch/>

Instalar el firmware en el Arduino

Este firmware es un programa que debe instalarse en vuestra placa [Arduino](#) para poder comunicarse con la placa desde S4A. Para ello es necesario instalar el IDE de Arduino.

- Descargar e instalar el entorno Arduino siguiendo las instrucciones de <http://arduino.cc/en/Main/Software>.
- Descargar firmware de la web de S4A http://s4a.cat/index_es.html sección descargas.
- Conectar la placa Arduino a un puerto USB de vuestro ordenador
- Abrir el archivo de firmware (S4AFirmwarexx.ino) desde el entorno Arduino
- En el menú Herramientas, seleccionar la versión de la placa y el puerto serie donde está conectada la placa
- Cargar el firmware a la placa mediante la opción de menú Archivo > Subir



Objetos Arduino

En S4A los objetos Arduino proporcionan bloques para las funcionalidades básicas del micro-controlador, escrituras y lecturas digitales y analógicas, y otras funcionalidades de más alto nivel. También encontraréis bloques para tratar con motores estándar y servomotores de rotación continua.

En S4A, una placa [Arduino](#) se representa con un tipo especial de sprite. El sprite Arduino encontrará automáticamente el puerto USB en que se haya conectado la placa.

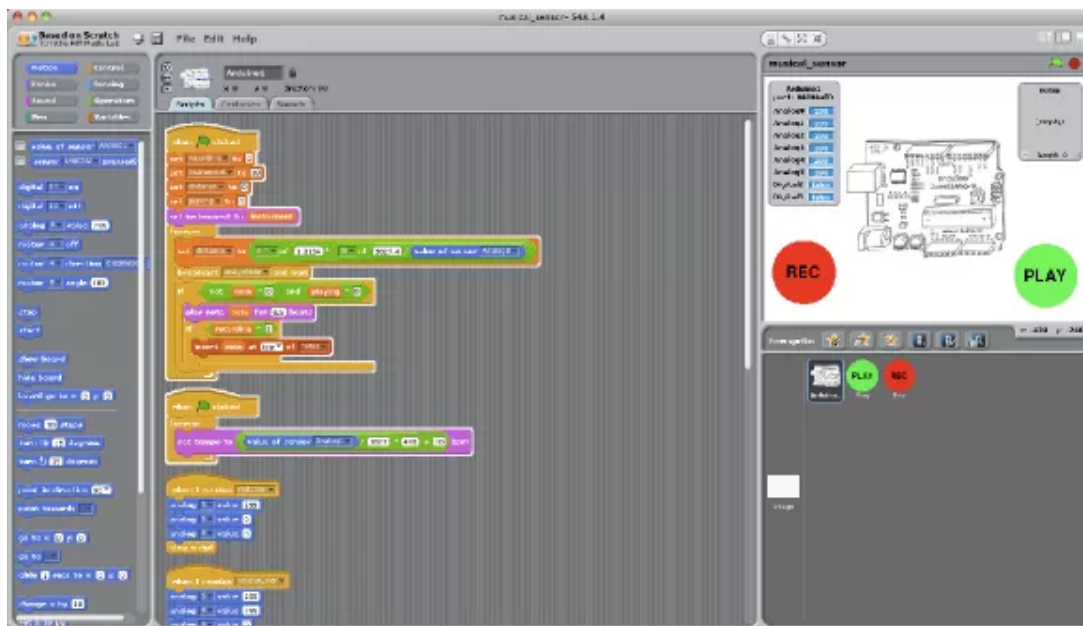
S4A habilita 6 entradas analógicas (pines analógicos), 2 entradas digitales (pines digitales 2 y 3), 3 salidas analógicas (pines digitales 5, 6 y 9), 3 salidas digitales (pines 10, 11 i 13) y 4 salidas especiales para conectar servomotores de rotación continua Parallax (pines digitales 4, 7, 8 y 12).

Es posible comunicarse inalámbricamente con una placa mediante un módulo de RF, como Xbee y con bluetooth.

S4A es compatible con [Scratch](#), de forma que es posible abrir proyectos de [Scratch](#) desde S4A.

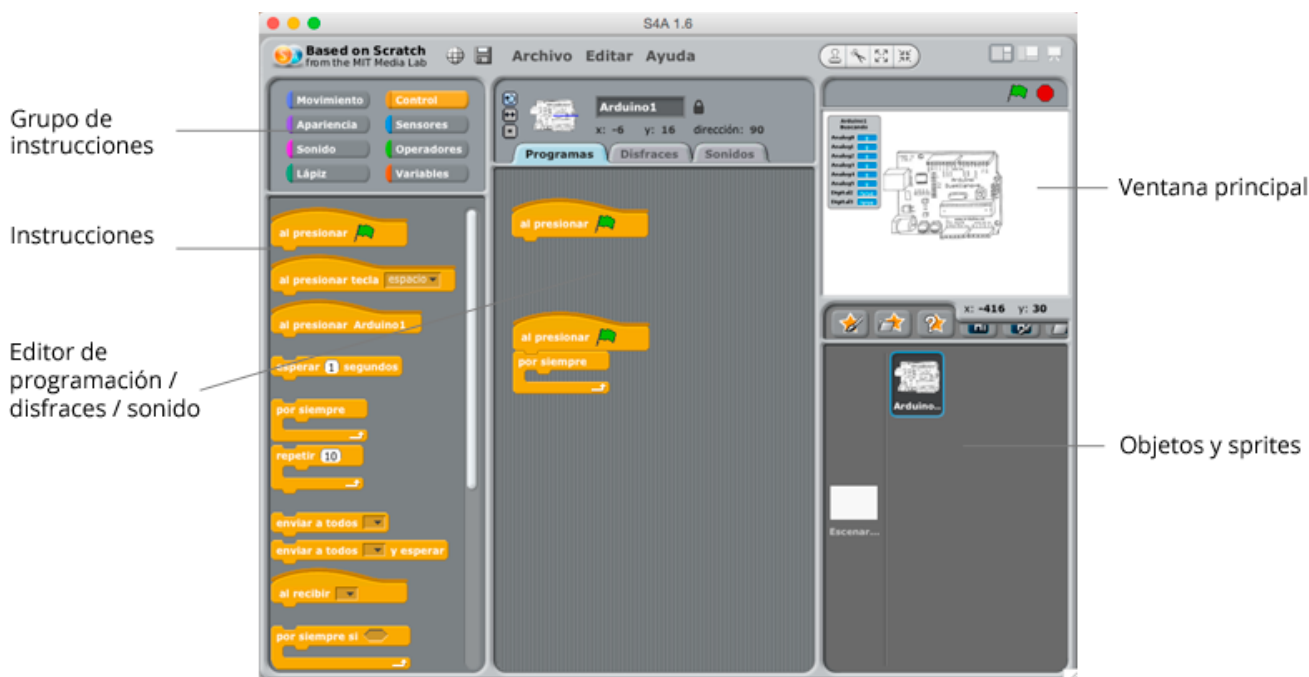
Entorno S4A

La interfaz de S4A es muy parecida a la de [Scratch 1.4](#) al ser una versión del mismo. Podemos importar los proyectos que ya tengamos de dicha versión de Scratch y adaptarlos para trabajar con nuestro Arduino. [S4A](#), al igual que Scratch 1.4 trabajan con proyectos con la extensión .sb. En el supuesto de querer basar nuestro trabajo en un proyecto de Scratch 2.0 (extensión .sb2) podríamos utilizar el [Retro Convertidor](#) de blob8108 o [seguir las instrucciones dadas en esta wiki](#) para convertir el archivo.

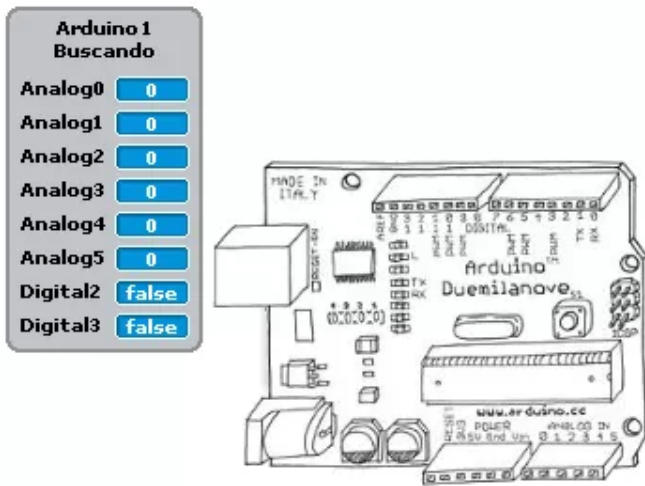


S4A se compone de 5 partes principalmente:

- **Grupo de instrucciones** clasificadas por colores en las siguientes categorías:
 - **Movimiento:** Conjunto de instrucciones relacionadas con el control de los pines de la tarjeta de Arduino, así como el control del movimiento de cualquier personaje del escenario.
 - **Apariencia:** Instrucciones orientadas a modificar el aspecto de los personajes de nuestra aplicación. Para el caso de Arduino, es un conjunto de instrucciones que apenas se utiliza.
 - **Sonido:** Conjunto de instrucciones relacionadas con la elaboración de aplicaciones musicales, emitiendo sonidos y notas musicales.
 - **Lápiz:** Scratch nos ofrece la posibilidad de que los personajes dejen un rastro durante sus movimientos por el escenario como si arrastrara un lápiz durante su trayectoria.
 - **Control:** Las instrucciones incluídas en esta sección son imprescindibles para crear la lógica de nuestros programas. Incluyen condicionales, bucles y llamadas de procedimientos.
 - **Sensores:** Instrucciones de iteración con el ratón, el teclado, sonidos y los personajes.
 - **Operadores:** operaciones matemáticas, lógicas y con cadenas de texto.
 - **Variables:** Instrucciones para el almacenamiento y gestión de datos.
- **Instrucciones de programación:** Las instrucciones de cada grupo corresponden a instrucciones de programación.
- **Editor:** Es la parte principal donde estructuramos y programamos nuestro programa.
 - Programas: Se compone de todas las instrucciones que hace funcionar el código que programemos.
 - Disfraces: Cada objeto puede tener diferentes apariencias o disfraces para utilizar a lo largo de nuestro programa.
 - Sonido: También es posible añadir o grabar sonidos y guardarlos para futuros usos.
- **Escenario o ventana principal:** Es el resultado de nuestro programa.
- **Objetos y sprites:** Distinguimos principalmente los objetos de tipo Arduino y Sprites.
 - Los objetos de tipo arduino son aquellos que interactúan con Arduino.
 - Los sprites son similares al entorno de scratch y no interactúan con Arduino.



En la parte del escenario veremos una tabla con pines de entradas y salidas de la placa Arduino, y un dibujo de nuestra placa. Si esta no está conectada al equipo mediante el USB aparecerá una ventana emergente con el mensaje "Buscando placa...", que desaparecerá cuando el S4A la reconozca.



Scratch for Arduino tiene varios bloques específicos para interactuar con la placa de Arduino. Se encuentran en la categoría "Movimiento" y permite enviar instrucciones al micro-controlador de la placa de Arduino, como escrituras y lecturas digitales. También hay bloques para tratar con motores estándar y servomotores de rotación continua.



Por otro lado, la parte de edición de objetos incorpora botones para crear, dibujar e importar objetos Arduino. Se podría conectar otra placa de Arduino y asociarla a otro objeto en un momento dado.



Programación con S4A

Normalmente los programas los empezaremos con el primero de los bloques (El de la bandera verde) y todos los demás están por debajo de éste. Pulsamos con el botón izquierdo del ratón encima de él y lo arrastramos hacia la parte central de S4A. Con este bloque lo que hacemos es que el programa se ejecute cuando pulsemos la bandera verde de la parte de arriba a la derecha y hasta que pulsemos en el hexágono rojo.

Como queremos que el programa se ejecute un número indefinido de veces, vamos a coger ahora el bloque de control “por siempre”. Si nos fijamos en la forma de este bloque veremos que la parte de arriba encaja con la parte de abajo del bloque con el que hemos empezado.

Además el bloque “por siempre” tiene un hueco en medio en el que podremos encajar otras instrucciones, de manera que todo lo que esté dentro de él se ejecute en secuencia hasta el final. Una vez que termina, vuelve a empezar desde el principio del bloque, haciendo un ciclo sin fin.



Por ejemplo el blink se haría así:

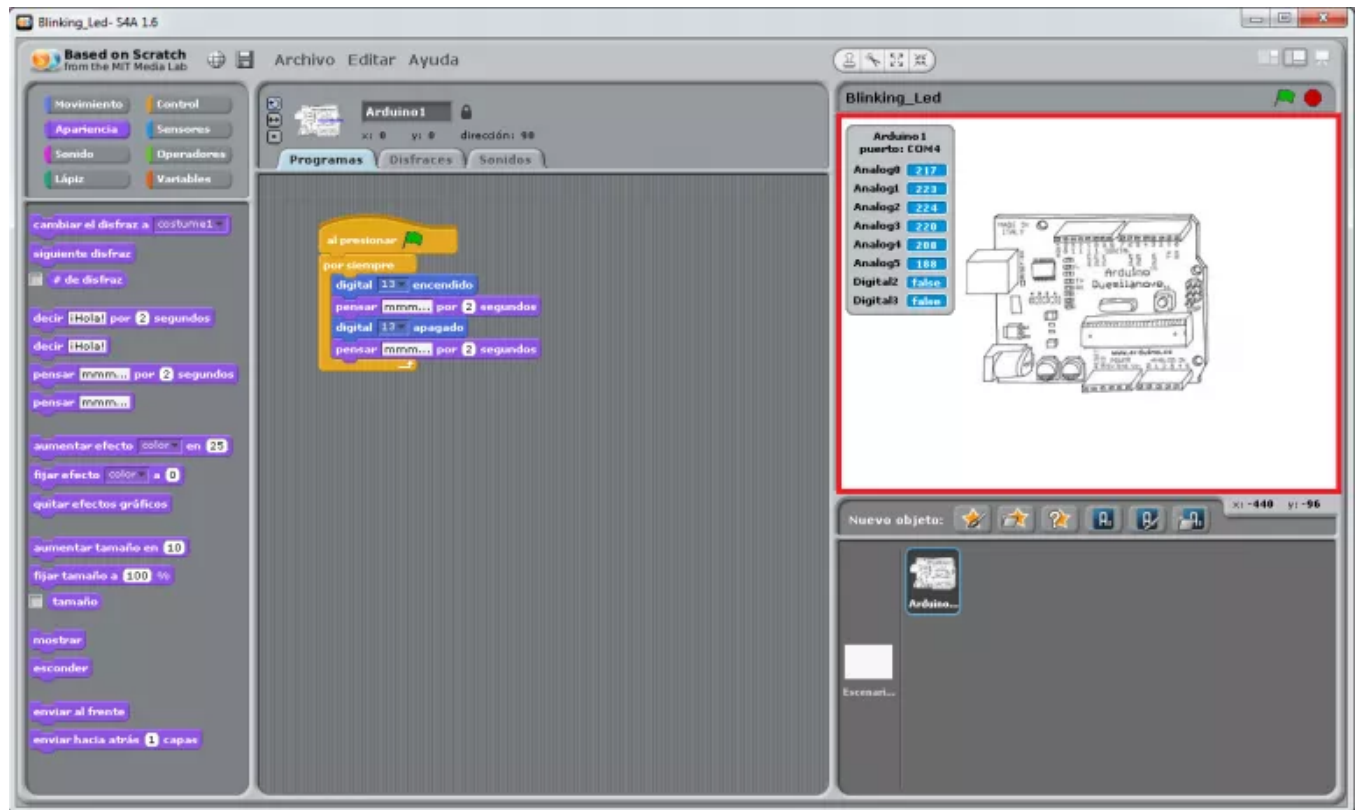


Para ejecutar el programa hay que pulsar la bandera verde, o también podemos hacer click con el botón izquierdo encima de cualquiera de los bloques de instrucciones. Veremos que todos los bloques de nuestro programa y la bandera verde del escenario se iluminan y el LED de nuestra placa Arduino comienza a parpadear cada dos segundos.

Por último, y si queremos guardar nuestro programa iremos a la pestaña “Archivo”, y le daremos a “Guardar Como...”. En ese menú tenemos también las opciones de “Abrir...” para abrir un proyecto que hayamos guardado anteriormente y “Guardar” para guardar el archivo encima del que tenemos abierto.

Escenario y Disfraces

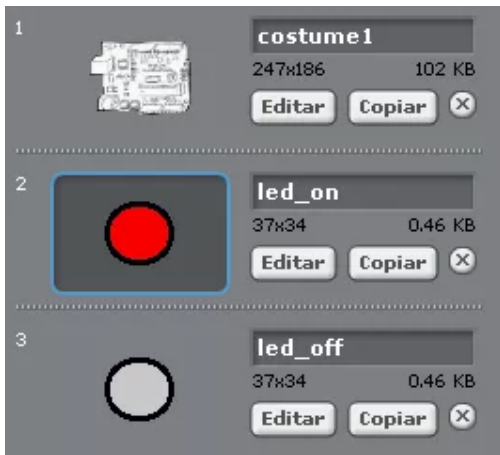
En el escenario podemos ver en el monitor de nuestro PC cómo se desarrolla nuestro programa. Es la pantalla en blanco que hay arriba a la derecha, donde aparece por defecto un dibujo de la placa Arduino. Ese dibujo es un ejemplo de lo que llamamos "Disfraces" en S4A.



Lo primero que haremos será ir a la pestaña "Disfraces". Ahí podemos ver que el único que hay es un dibujo de una placa Arduino que viene por defecto. Para dibujar un disfraz nuevo haremos click en el botón "Pintar".



Para que en el escenario aparezca un led encendiéndose y apagándose hay que pintar dos disfraces uno con un círculo rojo y otro blanco.



Los bloques relacionados con los disfraces están en el grupo “Apariencia”. En este caso usaremos el bloque “cambiar el disfraz a”.

Y el programa queda:



Más información:

- <http://www.prometec.net/scratch-primer/>
- <http://www.prometec.net/s4a-disfraces/>

Cómo Funciona S4A

S4A se comunica por puerto serie con Arduino que está corriendo un firmware que ejecuta las acciones que se indica desde S4A.

Firmware: <https://github.com/jecrespo/S4A/tree/master/Firmware%20Oficial/S4AFirmware16>

El firmware no es más que un intérprete de las instrucciones que le manda S4A por el puerto serie. S4A permite controlar tantas placas como puertos USB tenga vuestro PC.

Especificación del protocolo de comunicación entre S4A y Arduino: <http://s4a.cat/downloads/s4a-protocol.pdf>

S4A interactúa con [Arduino](#) enviando el estado de los actuadores y recibiendo el estado de los sensores cada 75 ms, por lo tanto el ancho del pulso tiene que ser mayor que éste período. El intercambio de datos sigue el protocolo PicoBoard.

Picoboard: <https://www.picocricket.com/picoboard.html>

Analizar el firmware que ejecuta Arduino y entender el código:

<https://github.com/jecresco/S4A/blob/master/Firmware%20Oficial/S4AFirmware16/S4AFirmware16.ino>

S4A en Android

Aplicación simple de Android que puede comunicarse con S4A via HTTP mediante el protocolo de sensores remotos de [Scratch](#).

La aplicación Android utiliza un protocolo muy sencillo sobre el protocolo de mensajes HTTP de [Scratch](#), la cual cosa permite interactuar con S4A desde un navegador o cualquier otra aplicación con acceso a la red.

Para ello debe activarse la opción de sensores remotos en S4A (clic derecho en cualquier bloque de "Sensores").

Especificación de los mensajes

Enviar un mensaje: <http://%5BIP%5D:42001/broadcast=%5Bnombre-del-mensaje%5D>

Enviar el valor de un sensor remoto (por ejemplo, el acelerómetro del cliente, la posición del puntero del ratón, etc): <http://%5BIP%5D:42001/sensor-update=%5Bnombre-del-sensor%5D=%5Bvalor%5D>

Actualizar una variable de S4A: <http://%5BIP%5D:42001/vars-update=%5Bnombre-de-la-variable%5D=%5Bvalor%5D>

Se pueden probar los mensajes desde la web: <http://s4a.cat/android/index.html> que manda las instrucciones de la API.

La forma de trabajar es que desde la app de S4A se conecta al ordenador que ejecuta S4A y este manda las instrucciones a Arduino. Para ello es imprescindible que tanto S4A como HiS4A tienen que estar conectados a la misma red local.

Toda la información:

- En la web http://s4a.cat/index_es.html
- Pdf: <http://martuino.com/img/S4ainfo.pdf>

Más información

- Prometec ¡¡¡MUY BUENO!!!: <http://www.prometec.net/scratch/>
- Programar fácil: <https://programarfácil.com/blog/arduino-blog/s4a-primeros-pasos-con-scratch/>
- Rufian: <http://rufianenlared.com/s4a/>
- Pdf: <http://codigo21.educacion.navarra.es/wp-content/uploads/2016/02/s4a-manual.pdf>

Ejercicios

Listado de materiales y prácticas de El Cable Amarillo:

- <https://github.com/ElCableAmarillo/Listado-de-materiales>
- <https://github.com/ElCableAmarillo/Listado-de-practicas>

Anuncios

Esta entrada se publicó en Arduino, Curso Avanzado 2017, Práctica, Programación Visual, S4A, scratch y está etiquetada con Arduino, Curso Avanzado 2017, Práctica, Programación por Bloques, Programación Visual, S4A, scratch en 7 septiembre, 2017 [<https://aprendiendoarduino.wordpress.com/2017/09/07/programacion-visual-con-s4a/>] .

