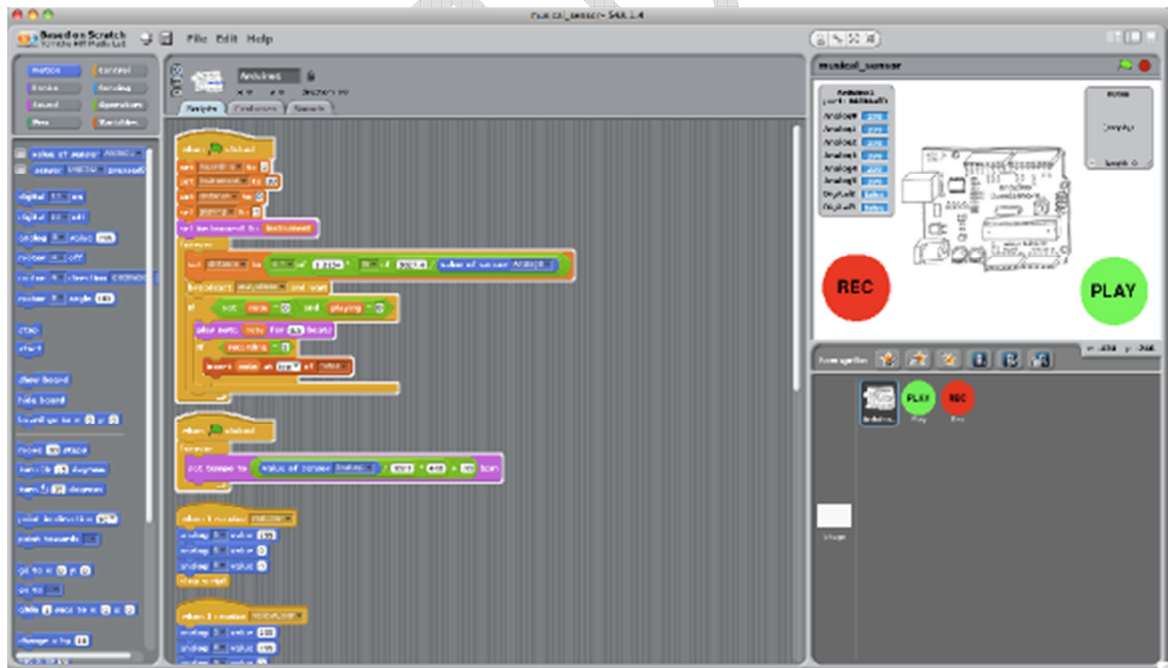


Scratch S4A

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

La finalidad principal del proyecto es atraer a gente al mundo de la programación. Otro de sus objetivos es proporcionar una interfaz de alto nivel para programadores de Martuino con funcionalidades tales como la interacción con un conjunto de placas mediante eventos de usuario.

La interfaz



Un programa de S4A para controlar un Theremin de luz, con capacidad de grabación y reproducción

Los objetos Martuino proporcionan bloques para las funcionalidades básicas del microcontrolador, 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 Martuino se representa con un tipo especial de sprite. El sprite Martuino encontrará automáticamente el puerto USB en qué se haya conectado la placa.

Pueden conectarse múltiples placas a la vez. Sólo es necesario añadir un nuevo sprite Martuino.

Especificaciones técnicas

Placas compatibles

S4A funciona correctamente con Arduino Diecimila, Duemilanove, Uno y Martuino. No se ha probado con otras placas, pero es posible que también funcionen.

Conectividad

Los componentes deben conectarse de una forma determinada. 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 Wifi.

S4A permite controlar tantas placas como puertos USB tenga vuestro PC.

Compatibilidad

S4A es compatible con [Scratch](#), de forma que es posible abrir proyectos de [Scratch](#) desde S4A. En cualquier caso, no es posible compartir proyectos de S4A en la web de la comunidad [Scratch](#), ya que esto violaría los términos y condiciones de uso de [Scratch](#).

Téngase en cuenta que esta compatibilidad no es bidireccional. No es posible abrir proyectos de S4A desde el [Scratch](#) original.

S4A permite utilizar una PicoBoard conjuntamente con un Martuino.

Protocolo

S4A interactúa con Martuino 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, y necesita que se instale un programa específico (firmware) en la placa. Por favor, consultad la sección de Descargas para más instrucciones sobre la instalación.

Hardware? Un juego de niños!

Scratch for Arduino (S4A) es una versión modificada de [Scratch](#), preparada para interactuar con placas Martuino. 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.

Nuestra motivación principal fue proporcionar una forma sencilla para interactuar con el mundo real, aprovechándonos de la facilidad de uso de [Scratch](#).

Documentación y ejemplos

Aquí encontraréis toda la documentación, vídeos, ejemplos y ejercicios que hemos producido hasta la fecha.

Taller

Desde aquí podéis descargaros un zip con materiales y ejercicios completos para realizar un taller:

[Materiales Taller Elisava 2011](#)

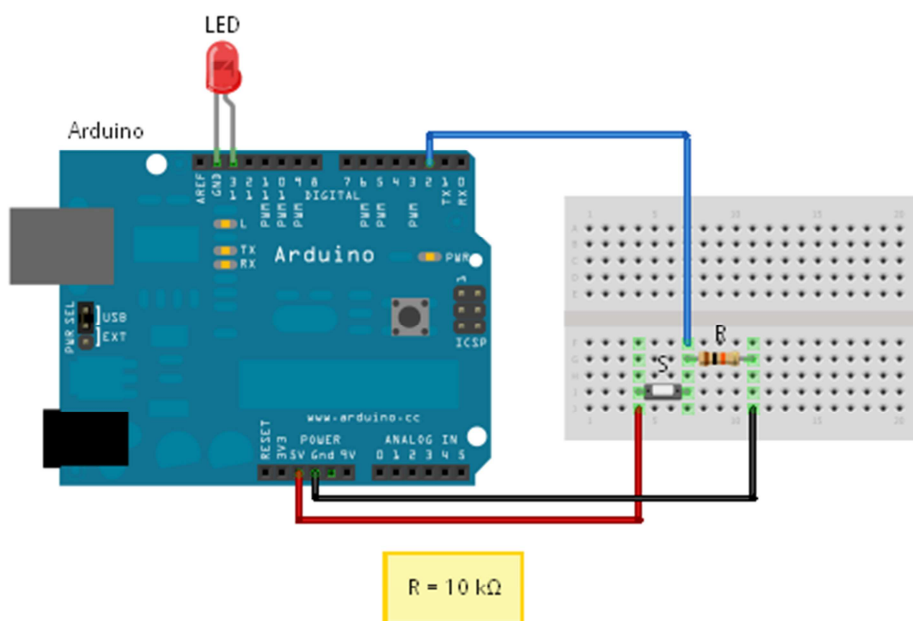
Nuestro equipo creó y utilizó estos materiales en un curso de S4A en la escuela Elisava durante el año 2011.

Ejemplos Básicos

Aquí encontraréis ejemplos básicos de S4A, con los esquemas de Martuino y un archivo .sb con el proyecto [Scratch](#) para descargar.

Pulsador y LED

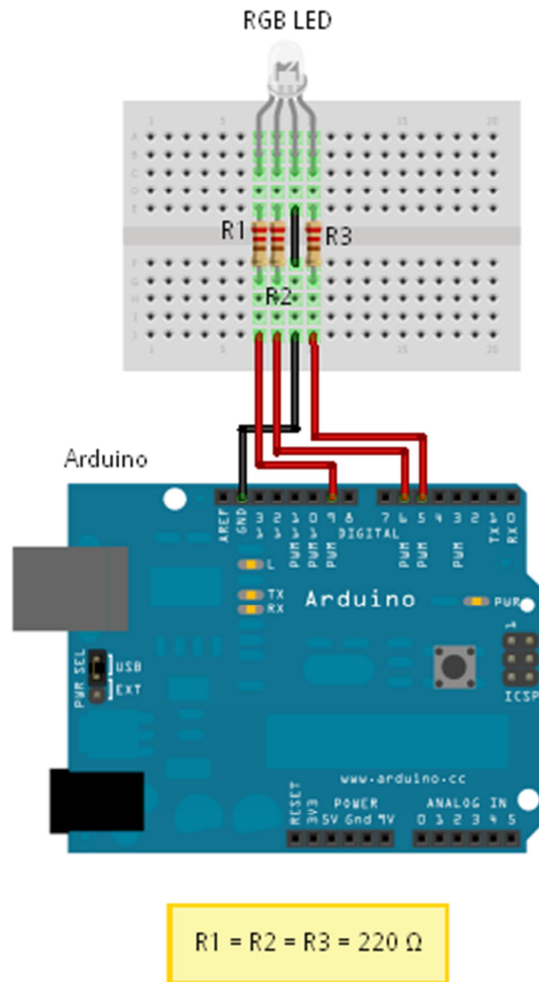
Un ejemplo simple mostrando como activar un LED mediante un pulsador.



[Código](#)

LED RGB

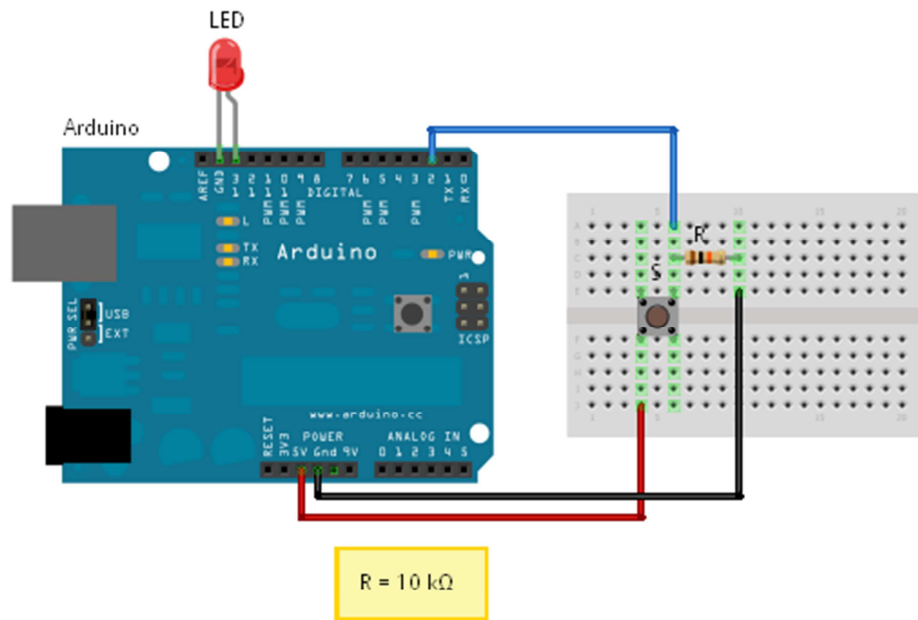
Como usar un LED RGB desde Scratch.



[Código](#)

Pulsador, LED y conservación de estado

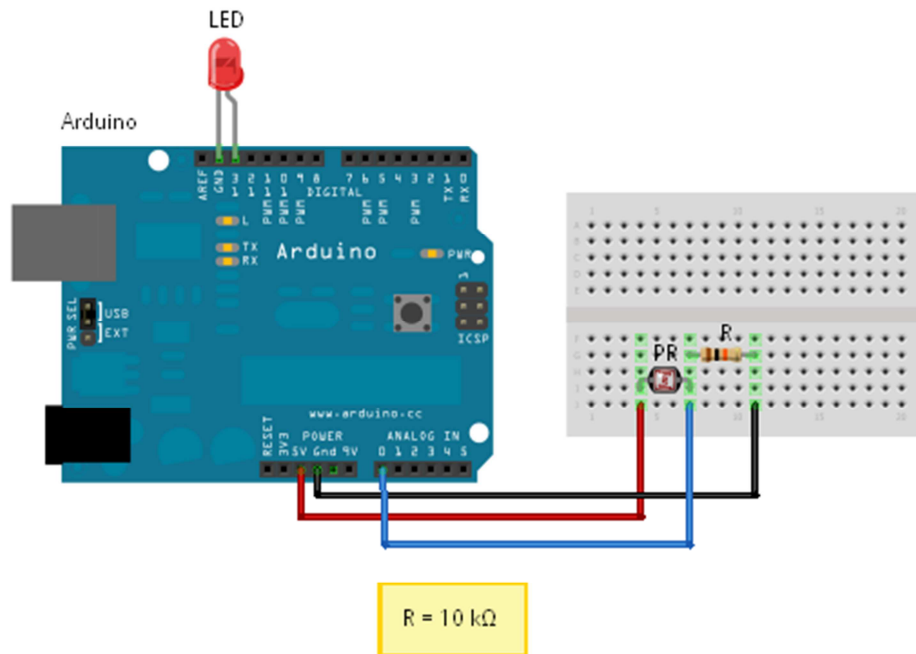
Un LED que cambia de estado al pulsar un botón.



[Código](#)

Fotoresistor y LED

Como usar la entrada de una LDR (foto-resistencia) para activar un LED.



[Código](#)

Vídeos

Aquí encontraréis una colección de demos en castellano. Algunos de estos vídeos también han sido subtitulados al inglés.

Robot completo

Un robot con una cámara rotativa, sensores de distancia y control remoto. Consta de dos placas Martuino (para el control remoto y el robot), tres sensores de infrarrojos, un módulo de cámara y tres servomotores.

Introducción al taller de S4A

Este vídeo sirve como introducción para los talleres de S4A que se imparten en el [Citilab](#). Encontraréis unos cuantos ejemplos de casos de uso de nuestro software.

Ejercicios

[Música: Un instrumento similar a un theremin](#)

[Domótica: Un controlador de persiana](#)

[Robótica: Un coche controlado por teclado](#)

Para programadores

[Especificación del protocolo S4A](#)

Enlaces externos de documentación

[Scratch IO Wiki - por José Javier Vitón](#)

[Blog Arduteka: S4A](#)

Android

Hi Scratch for Arduino!

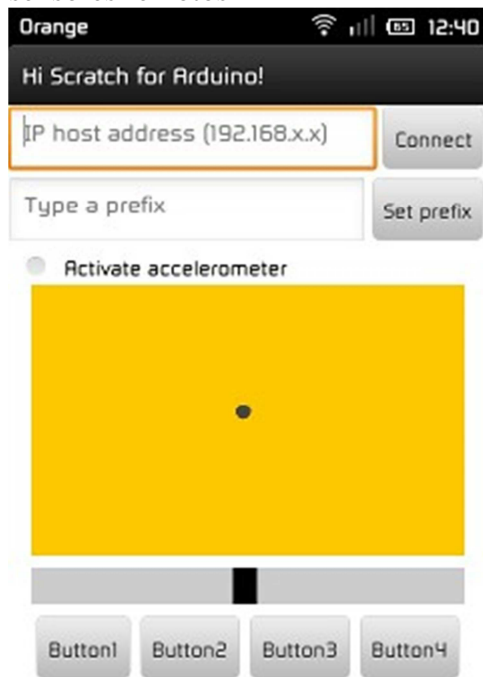
Hemos desarrollado una aplicación simple de Android que puede comunicarse con S4A via HTTP mediante el protocolo de sensores remotos de [Scratch](#). Os lo podéis descargar desde [aquí](#), o escaneando este código QR:



Para utilizarlo, seguid estas instrucciones:

- En S4A, activad los sensores remotos (clic derecho en cualquier bloque de "Sensores")
- En HiS4A (Android), introducid la IP local del PC en qué S4A esté funcionando (la podéis encontrar desde el menú Editar > Host de red), y continuación el prefijo (sólo necesario si os estáis conectando desde más de un cliente de Android)

- Los botones se corresponden con mensajes, y el touchpad y deslizador a sensores remotos



¡IMPORTANTE! *¡Tanto S4A como HiS4A tienen que estar conectados a la misma red local!*

API HTTP

La aplicación Android utiliza un protocolo muy sencillo que hemos desarrollado 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.

En primer lugar, debe activarse la opción de sensores remotos en S4A (clic derecho en cualquier bloque de "Sensores").

Especificación de los mensajes

Enviar un mensaje:

http://[IP]:42001/broadcast=[nombre-del-mensaje]

Enviar el valor de un sensor remoto (por ejemplo, el acelerómetro del cliente, la posición del puntero del ratón, etc):

http://[IP]:42001/sensor-update=[nombre-del-sensor]=[valor]

Actualizar una variable de S4A:

http://[IP]:42001/vars-update=[nombre-de-la-variable]=[valor]

Aplicación web de prueba

Para ver la API HTTP en funcionamiento, podéis utilizar la aplicación web de prueba:

[Hi S4A! desde el Web](#)

Uso de la aplicación web de ejemplo:

- Las variables de S4A con nombres que empiecen con un signo más (ej. +nombreJugador) son visibles y modificables desde el cliente web
- Las variables de S4A con nombres que empiecen con un signo menos (ej. - puntuacionJugador) sólo son visibles y modificables desde el cliente web
- Las variables de S4A con nombres que no empiecen con los signos más o menos no son visibles desde el cliente web
- Los mensajes de S4A con nombres que empiecen con un signo más (ej. +iniciarJuego) son visibles desde cliente web
- Los mensajes de S4A con nombres que no empiecen con un signo más no son visibles desde el cliente web

Cambios

Versión 1.5

- Arreglado bug de colisión entre objetos (Arduino y no Arduino)
- Arreglado bug relacionado con un error persistente al parar y reanudar la conexión con la placa

- Primitivas indexadas cambiadas a primitivas con nombre para una mejor compatibilidad con las nuevas máquinas virtuales de Squeak (gracias a Miriam Ruiz por el parche)
- Cambiado el pin 12 de servo estándar a salida digital normal
- Cambiado el pin por defecto de los bloques digital on/off de 13 a 10
- Añadido bloque de reiniciar actuadores
- Traducciones al catalán y castellano mejoradas

Versión 1.4

- Añadida compatibilidad completa con el protocolo HTTP (POST y GET). Ahora es posible mandar mensajes a S4A y observar/modificar valores de variables desde un navegador web (entre otras cosas). Ver sección de Android.
- Añadida opción para comprobar si hay actualizaciones disponibles desde el menú de ayuda. Ya no hace falta visitar la web para comprobarlo
- El idioma se detecta automáticamente al arrancar por primera vez (cuando el archivo Scratch.ini no existe). También se guarda en este mismo archivo al cambiarse desde el menú de idioma
- El puerto serie en qué se conecta una placa se abre correctamente al arrancar S4A
- Arreglado bug de la versión para Windows, en qué la barra de título no mostraba el nombre y la versión de S4A

Versión 1.3

- Arreglada la compatibilidad con Arduino UNO en Linux (gracias a Jorge Gómez). Es imprescindible cargar el nuevo firmware, así como actualizar el bootloader en determinadas placas
- Cambiados los nombres del panel de sensores y sus correspondientes bloques de Arduino para mostrar números de pin reales
- Añadidos comandos de servidor en el menú "Editar" para trabajar con comunicación en red sobre sockets
- Añadida compatibilidad para aplicaciones Android (HiS4A)
- Traducciones al catalán y castellano mejoradas
- Muchos bug arreglados

Descarga e instalación



Para instalar S4A, hay que instalar software tanto en el PC como en la placa Martuino. Aquí encontraréis los pasos detallados para poner S4A en marcha.

Instalar S4A en el ordenador

S4A funciona sobre los tres sistemas operativos de escritorio más populares. Descargad e instalad el que os convenga.

- [Windows](#)
- [Mac](#)
- [Linux \(Debian\)](#)
- [Linux \(Fedora\)](#) (*version 1.5*)
- [Raspbian \(Debian para RaspberryPi\)](#)

Instalar el firmware en el Martuino

Este firmware es un programa que debe instalarse en vuestra placa Martuino para poderos comunicar con ella desde S4A.

- Descargad e instalad el entorno Martuino siguiendo las instrucciones de <http://www.martuino.com/wpimages/guiaenergia.pdf>. Hay que tener en cuenta que, para trabajar con Martuino, se necesita la versión 0022 o superior.
- Descargad nuestro firmware [aquí](#)
- Conectad la placa Martuino a un puerto USB de vuestro ordenador
- Abrid el archivo de firmware (S4AFirmware15.ino) desde el entorno Energia o Visual Studio 2015.
- En el menú Herramientas, seleccionad la versión de la placa y el puerto serie en qué esté conectada la placa
- Cargad el firmware a la placa mediante la opción de menú Archivo > Subir

Para programadores

S4A es software libre sujeto a una licencia [MIT](#).

Como [Scratch](#) es, en realidad, una imagen de [Squeak](#) Smalltalk, todo su código viene incluido en la versión final. Para acceder a él, pulsad mayúsculas + clic izquierdo sobre el menú "Archivo" de S4A, y seleccionad "Salir del modo de usuario". Os encontraréis delante del entorno clásico de una imagen de [Squeak](#) 2.8, y tendréis acceso a un Browser, Workspace y todas las herramientas a qué estáis acostumbrados.

De todas formas, a la versión final de S4A hay código que ha sido compilado, y se habrán perdido varios nombres de variables y argumentos de mensajes. Por este motivo, también distribuimos una imagen con el código fuente limpio para desarrolladores, descargable desde este enlace:

[S4A para desarrolladores](#)