

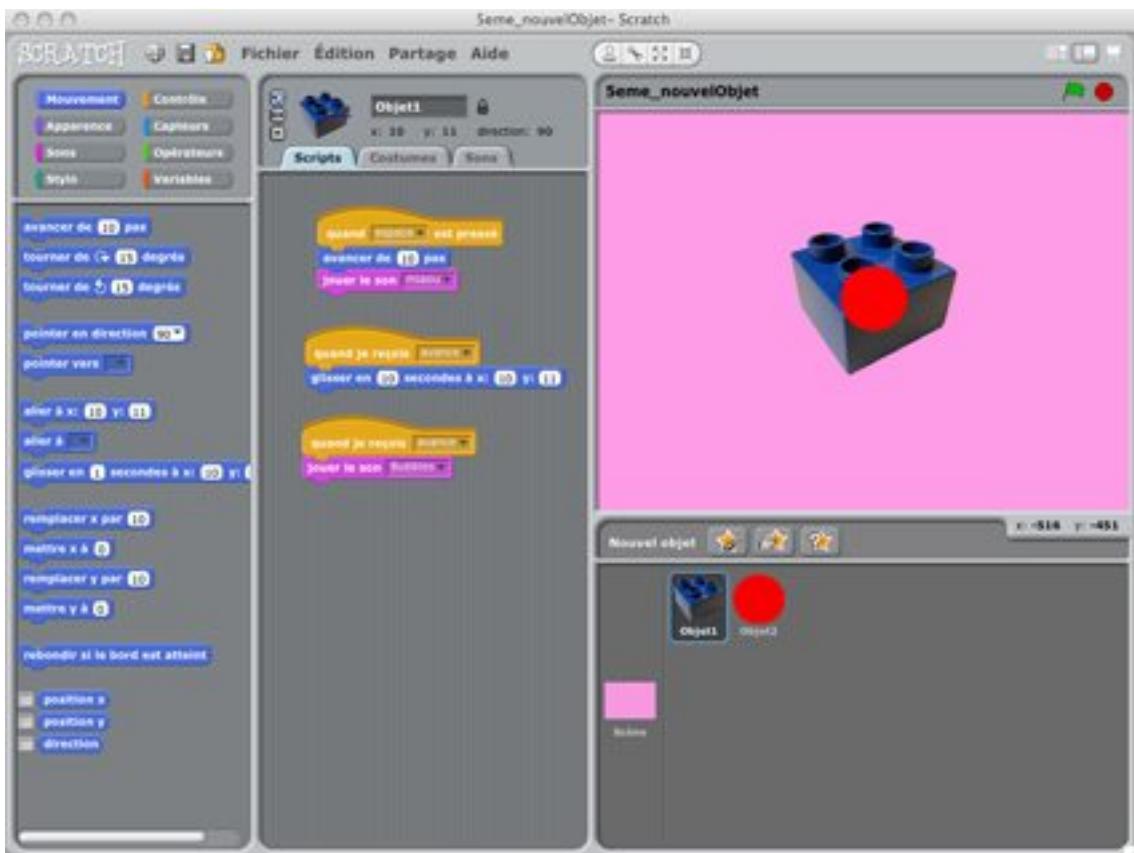
# S4A : un projet pédagogique

Les projets Sctratch et Scratch For Arduino sont développés par des Media Labs à des fins de transmission du savoir « programmer » pour les enfants, adolescents mais aussi les adultes qui découvrent l'univers de la programmation.

## Présentation de S4A et Scratch

S4A (Scratch For Arduino) est une extension (modification) de l'application Scratch qui a été créée en 2007 à des fins pédagogiques : apprendre à programmer des histoires interactives, à concevoir et réaliser des projets, à travailler en équipe. Ces applications permettent donc au plus grand nombre d'appréhender l'univers informatique pour comprendre ces outils numériques qui font dorénavant partie intégrante de notre quotidien.

Un détour vers Scratch va nous aider à découvrir S4A. Scratch est une application développée par la fondation Lifelong Kindergarten au sein du Media Lab du M.I.T. Scratch est un environnement de programmation qui permet d'aborder la programmation de façon ludique et intuitive. Le concept d'écriture du programme repose sur un principe d'assemblage de modules classés par code couleur (comme l'idée du Légo) et chaque module correspond à une instruction. Accessible aux enfants dès 8 ans, Scratch et S4A peuvent être un premier pas vers la programmation pour les adultes et un moyen de prototypage rapide pour les artistes et concepteurs.



Pour en savoir plus :

- [scratch.mit.edu](http://scratch.mit.edu).

- [llk.media.mit.edu](http://llk.media.mit.edu).

L'extension S4A a été développée par Marina conde, Victor Casado, Joan Güell, Jose García et Jordi Delgado du Media Lab Citilab en Espagne aidé par le Smalltalk Programming Group. Cette extension permet de programmer des instructions que la carte d'interfaçage Arduino peut exécuter et ce depuis une GUI Scratch à laquelle les créateurs ont ajouté des fonctions spécifiques à la carte Arduino. En effet, le principe d'écriture d'un programme avec Scratch et S4A consiste à emboîter des blocs classés en code-couleur qui correspondent à des instructions programmatiques : fonctions, variables, opérations, lecture de fichiers importés (images, sons). S4A présente donc des nouveaux blocs destinés uniquement à une programmation d'interactivité entre des capteurs et des actionneurs via une carte Arduino.

Pour en savoir plus :

- [seaside.citilab.eu](http://seaside.citilab.eu).

## Installer S4A

1. Se rendre sur le site <http://seaside.citilab.eu/> à l'onglet « Download ».
2. Veiller à ce que l'IDE Arduino soit installée dans les applications de votre ordinateur.
3. Télécharger le « Firmware » développé par le Citilab.
4. Depuis l'IDE Arduino, configurer le port USB qui communique entre la carte Arduino et votre ordinateur et sélectionner la version de la carte Arduino que vous utilisez.
5. Charger le Firmware dans la carte d'interfaçage Arduino (ce Firmware est un programme Arduino écrit par le Citilab destiné à traduire les programmes S4A afin que l'extension S4A et la carte Arduino puissent communiquer).
6. Télécharger et installer l'extension S4A.

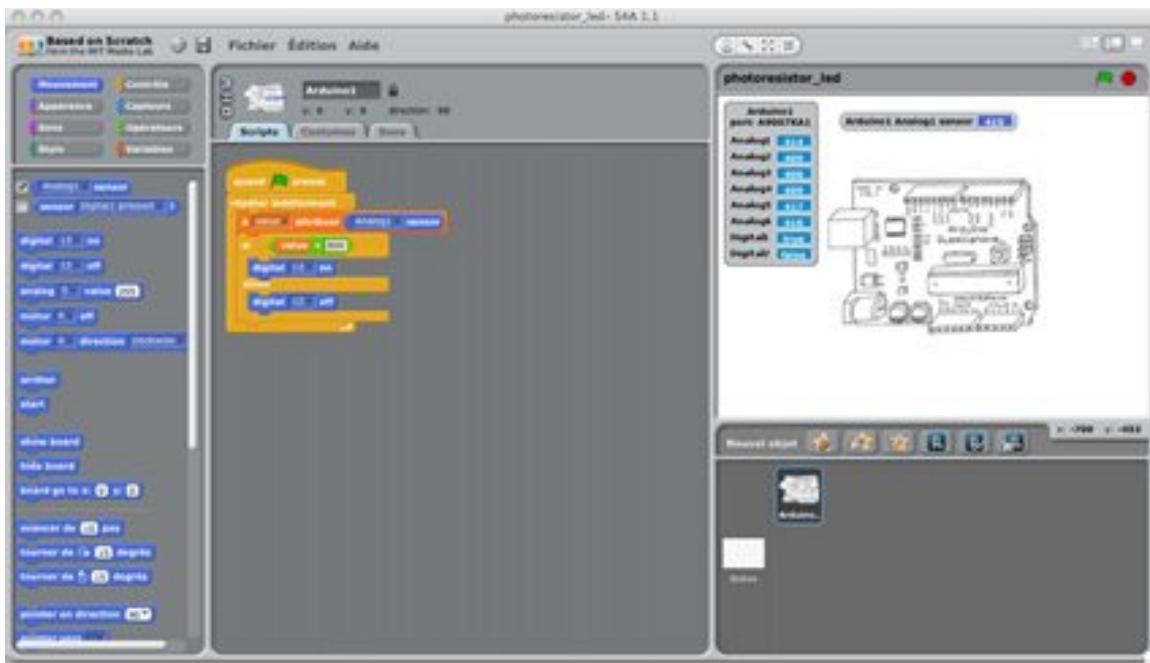
## Débuter avec S4A

Toujours dans l'onglet « Download », un répertoire d'exemples est téléchargeable pour débuter avec S4A (« Basic exemples ») et une vidéo de démonstration présente l'environnement et sa maniabilité à travers un prototype réalisé.

Suivons l'exemple donné par le Citilab d'un programme S4A : « photoresistor\_led.sb » (« .sb » est l'extension usuelle des fichiers des programmes Scratch) avec le schéma électrique du montage fait sous Fritzing (« photoresistor\_led.png »).

Connectez d'abord votre carte Arduino à l'ordinateur en veillant à ce que sous l'application Arduino vous ayez bien sélectionné la carte Arduino que vous possédez et le port USB de votre ordinateur que vous utilisez et en ayant préalablement effectué le montage de la LED et de la LDR (Light Diode Resistance pour capteur de lumière) tel le schéma électrique proposé.

Lancez l'application S4A et depuis son menu choisir « ouvrir », une fenêtre S4A apparaît avec les différents exemples que vous avez téléchargés : ouvrir le fichier « photoresistor\_led.sb ». Le programme S4A en visuel ci-dessous apparaît.



Il se peut que S4A affiche une fenêtre « Arduino 1 » comme le visuel ci-dessous. Il s'agit d'accepter la connexion entre l'application S4A et la carte Arduino. Sélectionnez « Use connection of sprite - Arduino1 » et cliquez sur « Accept ».



Expliquons le programme « photoresistor\_led » ci-dessous :



« Quand le drapeau vert est pressé » signifie « quand nous avons compilé le programme et que nous l'exécutons ».

« Répéter indéfiniment » correspond à la fonction pré-enregistrée dans Processing « draw » soit le programme est exécuté en boucle jusqu'à ce que l'on appuie sur « stop » pour arrêter la lecture en boucle.

« A value attribuer Analog1 sensor » : ici nous avons d'abord déclaré une variable nommée « value » (ceci s'est fait en cliquant sur l'onglet rouge « Variables » en haut à gauche de l'interface S4A puis en cliquant sur « Nouvelle Variable ») puis nous attribuons à cette variable « value » la valeur numérique traduite du capteur LDR connecté à la carte Arduino et que nous avons nommé « Analog1 » car ce capteur est connecté sur le port (pin) A0.

N.B. : sur notre carte Arduino, les ports d'entrées/sorties analogiques sont nommées de A0 à A5 (par exemple pour une carte de type Duemilanove). Sur l'interface de l'application S4A, ces ports sont traduits en commençant à 1, c'est-à-dire que le port A0 de la carte correspond à la nomination « Analog1 », le port A1 à « Analog2 », etc.

Nous pouvons lire les valeurs numériques traduites du capteur dans le tableau de lecture qui apparaît par défaut dans l'interface de S4A (ci-dessous) :



Ensuite est intégrée dans l'exécution en boucle du programme la structure conditionnelle « si » et « sinon » : si la valeur attribuée à notre variable « value » est inférieure à (800), nous envoyons l'instruction « on » au port digital 13 de la carte Arduino, celui-la même sur lequel nous avons connecté la LED : donc la LED s'allumera. Sinon, nous envoyons l'instruction « off » et la LED restera éteinte.

Nous pouvons dès lors modifier notre programme en cliquant sur la valeur (800) et en entrer une autre pour paramétriser notre programme en fonction de la luminosité ambiante. Nous pouvons aussi connecter d'autres capteurs de lumière et LED à la carte Arduino et rajouter des instructions de lecture : dans l'onglet « Mouvement », choisir par exemple « digital 13 on » et en cliquant sur la flèche de déroulement d'items, sélectionner (10) ou (11) pour rajouter sur les ports de la carte des LED sur les ports 10 et 11.

Et enfin, nous pouvons aussi importer un nouvel objet dans S4A comme un visuel d'une créature fantastique et programmer ses actions sur la scène (background) en fonction de capteurs (bouton, potentiomètre, flexible, etc.) pour le faire jouer nous même.

Amusez-vous bien !