



Physical Etoys et la carte Arduino

Les premiers pas pour programmer la carte Arduino avec
Physical Etoys

GIRA
Version française : Planète Sciences

19 juin 2012

Table des matières

1	Installation	3
1.1	Ce qu'il vous faut	3
1.2	En option...	3
1.3	Installation des pilotes	3
2	Connecter la carte Arduino à Physical Etoys	4
3	Les premiers pas : faire clignoter une LED	7
4	Allumer une LED en appuyant sur un bouton	10
5	Utiliser une photo-résistance	12
6	Faire tourner un servo-moteur avec un rectangle	13
7	Conclusion	14

En raison de la popularité et des différentes fonctionnalités de la carte Arduino, nous allons voir dans ce tutoriel comment accéder à certaines d'entre elles depuis Physical Etoys : au programme, des petits scripts simples et une introduction à l'électronique.

1 Installation

1.1 Ce qu'il vous faut

- une carte Arduino (nous avons testé avec la Duemilanove),
- un câble USB,
- le logiciel Physical Etoys .

1.2 En option...

Une LED, un petit bouton poussoir, une carte de prototypage pour faire des essais de branchements, 3 résistances 10 KOhms et un servo-moteur. Ce matériel nous permettra de vérifier que la carte fonctionne.

1.3 Installation des pilotes

Lorsque vous connectez la carte Arduino, Windows la détecte et lance la procédure d'installation de pilote (si c'est la première fois que vous utilisez une Arduino avec votre ordinateur).

Sous Windows Vista ou Windows 7, le pilote devrait être téléchargé et installé automatiquement.

Sous Windows XP, l'assistant d'ajout de matériel va s'ouvrir :

- Sélectionnez l'option « Non, pas cette fois ci » lorsque Windows essaie d'effectuer une recherche automatique de pilote. Cliquez sur « Suivant ».
- Sélectionnez « Installer depuis une liste ou un emplacement spécifique (utilisateurs avancés) » puis cliquez sur « Suivant »
- Assurez vous que l'option « Rechercher le meilleur pilote à cet emplacement » est sélectionnée ; désélectionnez « Media amovibles » ; sélectionnez « Inclure cet emplacement » et indiquez le chemin d'accès du répertoire /FTDI USB Drivers de l'installation Arduino. si nécessaire, les tout derniers pilotes peuvent être téléchargés sur le site de FTDI (cartes Duemillanove, notamment) ou bien d'Arduino directement (cartes UNO). Cliquez sur "suivant".
- L'assistant va rechercher le pilote, puis indiquer qu'un « Convertisseur série USB » a été détecté (Duemillanove), ou bien une carte Arduino (UNO). Cliquez sur « Terminer ».
- L'assistant d'installation de nouveau matériel est alors relancé (Duemillanove). Suivez les même étapes que précédemment. Cette fois-ci, un « Port série USB » est détecté et installé.

Vous pouvez vérifier que le pilote a été installé en ouvrant le gestionnaire de périphériques de Windows (onglet « Matériel » dans le panneau de contrôle système).

Dans la catégorie « Ports », vous devriez voir apparaître la carte Arduino sous la forme d'un « Port série USB »(Duemillanove), ou bien directement avec le nom « Arduino ».

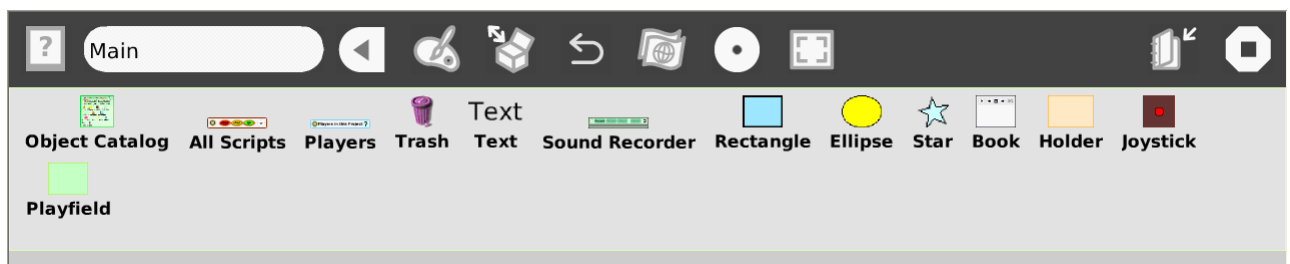
2 Connecter la carte Arduino à Physical Etoys

Pour démarrer, il nous faut un objet graphique représentant la carte Arduino sous Physical Etoys .

Pour l'obtenir, cliquez sur l'icône du tiroir **Accessoires** dans le menu du haut.



Le tiroir **Accessoires** contient les objets les plus utilisées. Nous en apprendrons plus à leur sujet au fur et à mesure. Celui qui nous intéresse pour le moment est le catalogue d'objets.

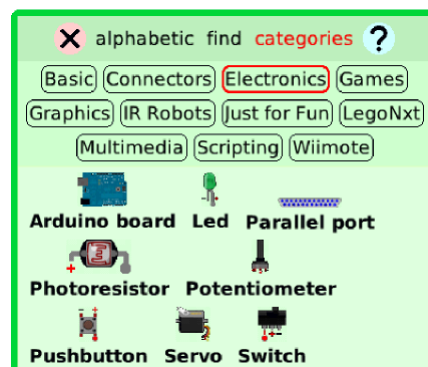


Déposer le catalogue d'objet sur le bureau de Physical Etoys .



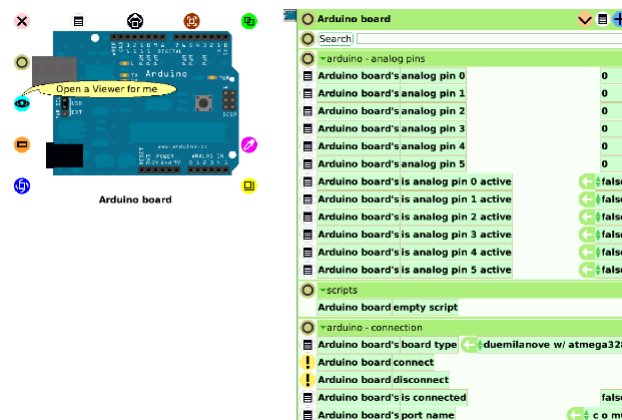
Le catalogue est une sorte de boîte contenant tous les objets que l'on peut utiliser. Ils sont classés par catégories mais vous pouvez les aussi les trier de manière alphabétique.

Maintenant choisissez la catégorie **Électronique**. Puis glissez/déposez le symbole de la carte Arduino sur le bureau.

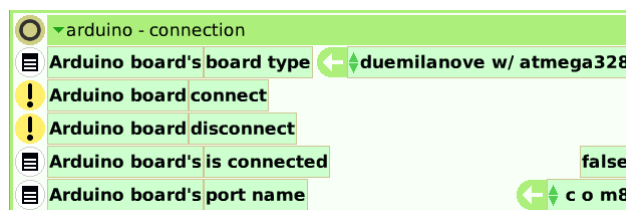


Nous pouvons ensuite afficher le **visualisateur** de la carte en faisant un clic droit sur son image. Ceci affiche une série d'icônes autour de la carte qu'on appelle le **halo**. Chaque icône joue un rôle particulier, comme permettre de faire tourner, agrandir ou rétrécir l'objet, le supprimer...

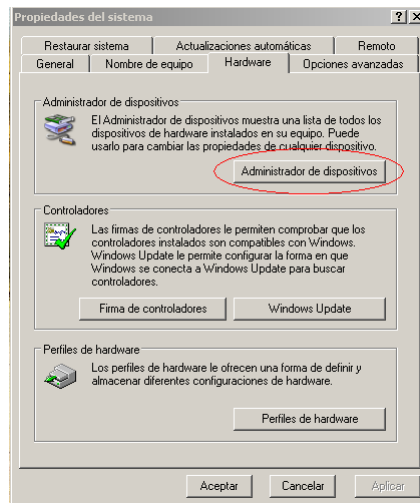
En cliquant sur l'icône  œil icon, le visualisateur apparaît. Le visualisateur est un panneau qui permet non seulement de voir et modifier les propriétés des objets, mais aussi de créer des **scripts**. Les scripts sont les petits programmes qui vont animer nos objets (les faire bouger, tourner ou ... contrôler la carte Arduino). Les propriétés et les actions sont représentés par des **briques**.



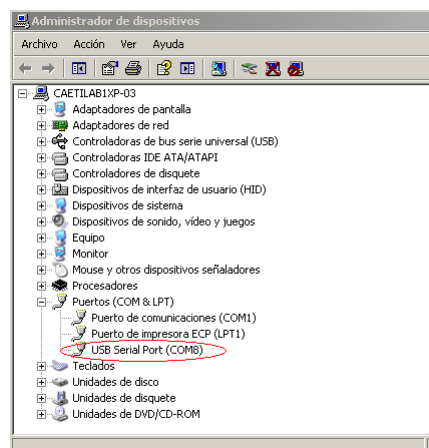
Dans la catégorie appelée **Connection Arduino**, vous pourrez trouver des informations très utiles pour connecter votre carte Arduino. Pour changer de catégorie il faut double cliquer sur le petit triangle en haut à gauche du titre de la catégorie puis choisir la catégorie désirée.



Nous devons d'abord spécifier quelle carte nous allons utiliser. Dans notre exemple nous choisirons la carte Duemilanove avec le microcontrôleur ATmega328. Puis il faut définir le port COM par lequel la carte est connectée. Pour le savoir il suffit de cliquer droit sur « Poste de travail » ou « Ordinateur » (si vous êtes sur PC Windows) puis « Propriétés ». Puis dans le disque dur, cliquez sur « gestionnaire de système ».



Maintenant regardez dans la section « ports » et une fois dedans vous verrez le numéro du port USB près de l'inscription « Port série USB ». Dans notre exemple c'est le port 8.



Donc la valeur correspondant au port sous squeakbot sera COM 8.

Arduino board's port name COM8

Ensuite, la commande importante est **connecter**. En exécutant cette commande (en cliquant sur l'icone **!** **exclamation** icon), nous connectons la carte Arduino avec Physical Etoys, et la brique indiquant que la carte Arduino est connectée changera sa valeur en **vrai**.

! Arduino board connect

Si c'est la première fois que vous essayez de connecter Arduino, une fenêtre apparaîtra vous demandant si vous désirez installer Firmata. Firmata est le programme contenant le protocole de communication permettant à la carte de recevoir des commandes. Il faut donc cliquer sur « oui ».

```
Firmata does not seem to be installed.
Do you want to install it now?
Yes
No
```

Une barre de progression vous indique où vous en êtes dans l'installation.

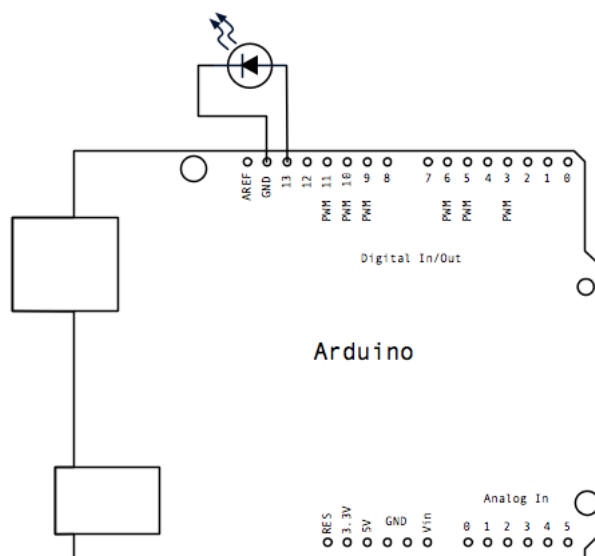
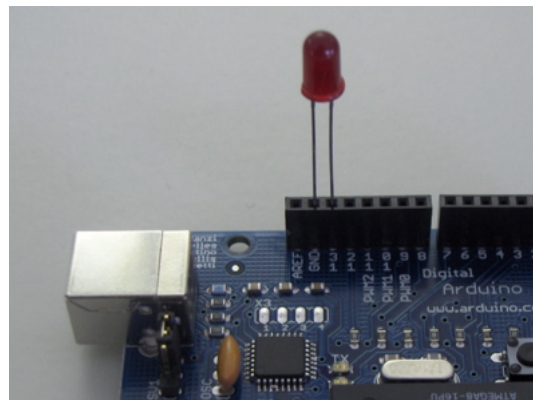
```
Installing Firmata
(this might take a few seconds)
```

Si tout se passe bien, une légende apparaît pour indiquer que l'installation s'est correctement réalisée. Maintenant il faut encore connecter la carte Arduino (en cliquant sur le **!** point d'exclamation de la brique associée). Si la carte Arduino est connectée, la brique **est connecté(e)** passe à **vrai**.

```
Arduino board's is connected true
```

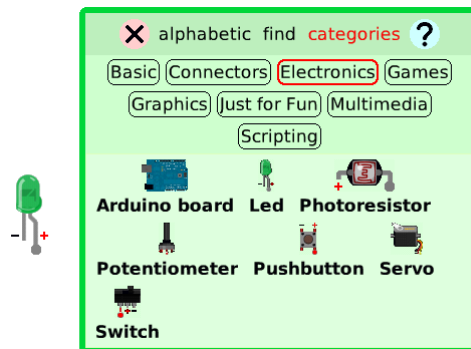
3 Les premiers pas : faire clignoter une LED

Avant de pouvoir utiliser Physical Etoys, il est nécessaire de connecter physiquement la LED à la carte. Dans cet exemple, la cathode (la patte la plus courte) doit être reliée à la masse (GND) et l'anode à la treizième broche (D13).

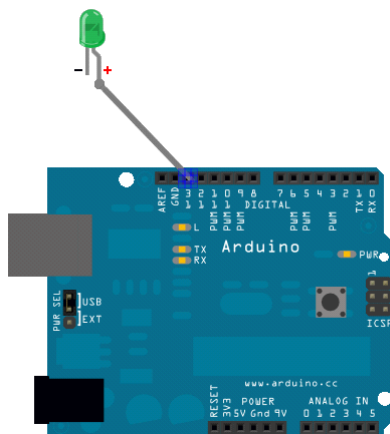


Il est désormais possible de passer à Physical Etoys .

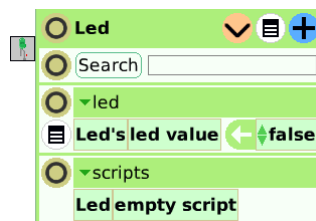
Dans la catégorie **Électronique**, sélectionnons une LED, qu'on peut glisser et déposer dans la fenêtre principale.



Une fois que la LED est placée dans la fenêtre principale, cliquons sur sa broche + pour faire apparaître un fil, qu'on peut ensuite relier à une des broches de l'objet Arduino (dans notre cas, donc, la broche D13).



Ensuite, allons dans le visualisateur de la LED, pour faire apparaître ces briques :



Créons maintenant un nouveau script pour faire clignoter la LED.

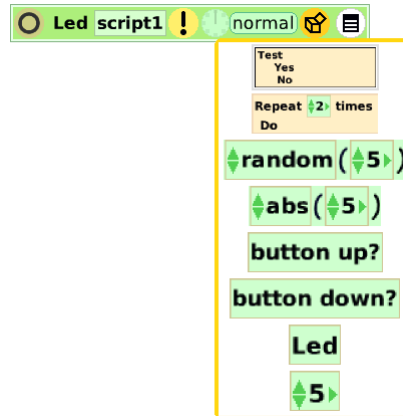
Pour cela, il faut glisser/déposer la brique **script vide** qui se trouve dans la catégorie **Scripts**. Une fois que ceci est fait, on obtient cela :



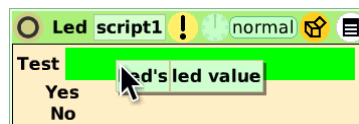
Ensuite, il faut cliquer sur l'icône  **boîte à outil** icon et choisir une brique **Si : Oui/Non**.

Cette brique signifie que le script va faire quelque chose selon une condition donnée (dans ce cas, la valeur de la LED, représentative de son état).

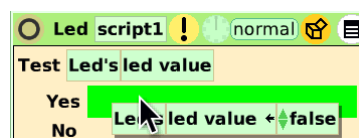
Pour ceux qui ont déjà quelques notions de programmation, cette brique est une **structure conditionnelle** (ou structure IF).



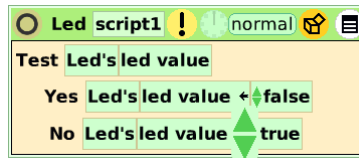
Afin de définir notre condition, il faut attraper la brique **Led.est allumée** et venir la déposer à côté de **Si** (attention, il ne faut pas prendre la brique par la flèche verte puisque nous voulons **utiliser** la valeur de la LED, et non lui **assigner** une valeur).



Maintenant, nous devons décider de ce que le script va faire lorsque la LED est allumée. Très logiquement, nous voulons... l'éteindre. Donc cette fois-ci, il faut **assigner** la valeur **Faux** à la LED. Il faut donc prendre la brique **Led.est allumée** par la flèche verte.



Maintenant, nous allons faire la même chose avec une petite différence : quand la LED est éteinte, Physical Etoys la fera s'allumer. Nous devons donc faire glisser la brique **Led.est allumée** avec sa flèche en face du **Non** de la brique **Si : Oui/Non**, en s'assurant que sa valeur est **vrai**.



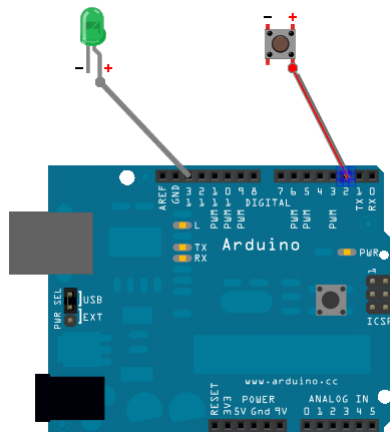
Pour conclure, nous pouvons démarrer le script en cliquant sur l'icone **horloge** icon .



Si tout fonctionne correctement, la LED va clignoter.

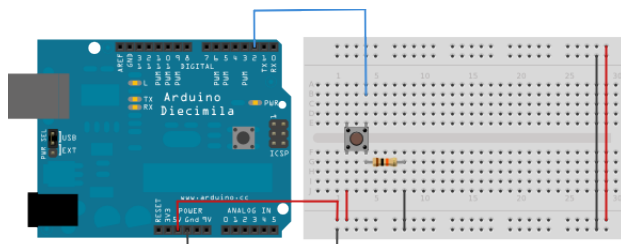
4 Allumer une LED en appuyant sur un bouton

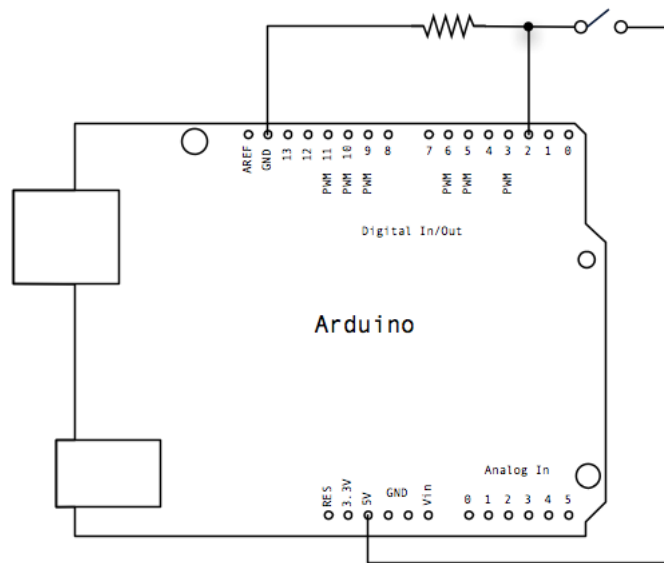
Dans la catégorie **Électronique** du catalogue d'objet, attrapons le **Bouton poussoir**, pour le connecter ensuite à la deuxième broche de la carte Arduino :



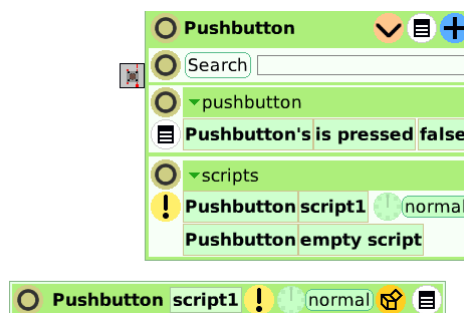
Il vaut mieux arrêter le script précédent (en cliquant sur l'icone **Stop** icon) pour cette partie. Par contre, gardons la LED branchée sur la broche 13.

Branchons le bouton (avec la résistance sur la carte de prototypage) comme ceci :

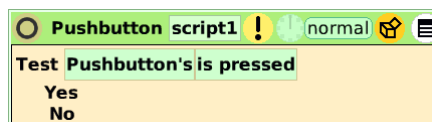




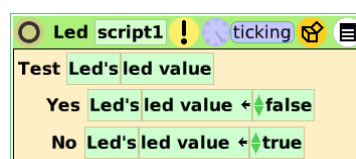
Maintenant, ouvrons le visualiseur du bouton poussoir dans Physical Etoys et créons un nouveau script.



Ajoutons ensuite une brique **Si : Oui/Non** afin de savoir quand le bouton est pressé. Nous devons faire glisser la brique par son nom car ce n'est pas une instruction. Nous voulons juste connaître sa valeur.



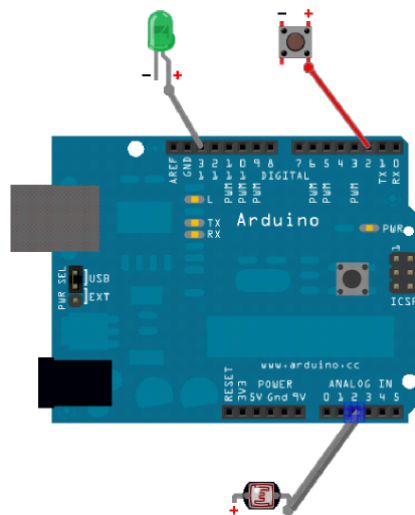
Quand le bouton est appuyé nous voulons assigner à la LED la valeur V_{vrai} , dans l'autre cas, nous lui assignons la valeur F_{faux} . N'oubliez pas cette fois de faire glisser la brique avec sa flèche. Pour conclure, démarrons le script en cliquant sur l'icône **horloge** icon.



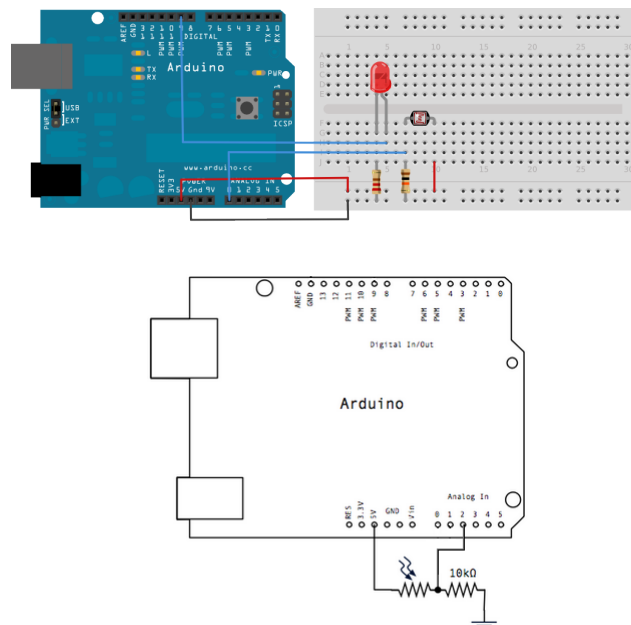
Si tout fonctionne correctement, quand on appuie sur le bouton, la LED clignote.

5 Utiliser une photo-résistance

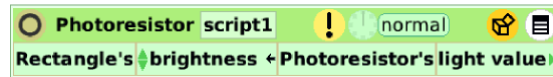
Tant que la carte Arduino est connectée, nous allons ajouter une **photo-résistance** (qui est elle aussi dans la catégorie **Électronique**) à la seconde broche des entrées analogiques.



Une fois branché, le montage ressemble à ceci :



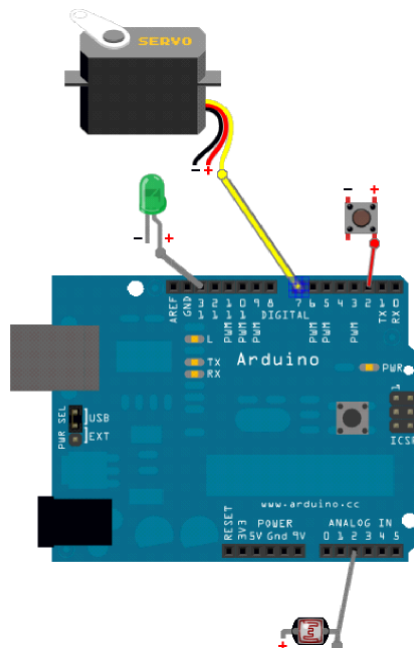
Maintenant, ouvrons le visualiseur de la photo-résistance et créons un nouveau script. Ouvrons le catalogue d'objet et faisons glisser un rectangle à partir de la catégorie **Objets de base**. Ensuite ouvrons le visualiseur du rectangle à la catégorie **Couleur** afin de faire glisser (avec sa ← flèche) la brique **luminosité** dans le script de la photo-résistance. Maintenant, ouvrons le visualiseur de la photo-résistance pour remplacer la valeur de la luminosité du rectangle (**100**) par la brique **luminosité** de la photo-résistance (sans la flèche, car nous voulons juste lire la valeur).



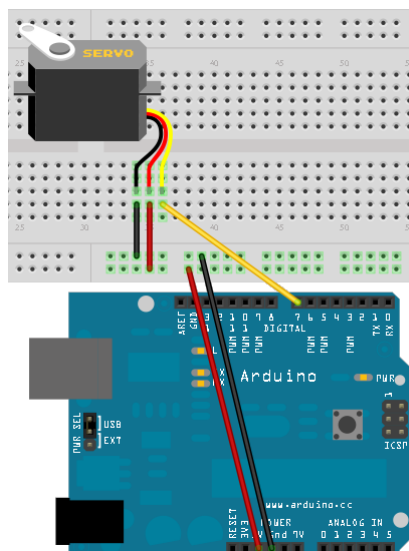
Pour finir, nous pouvons exécuter le script pour voir comment le rectangle change d'intensité lumineuse en fonction de la lumière ambiante.

6 Faire tourner un servo-moteur avec un rectangle

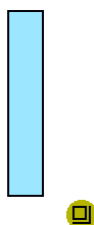
Avec la carte Arduino toujours connectée, branchons un servo-moteur (lui-aussi dans la catégorie **Électronique** du catalogue) sur la septième broche.



Sur la carte réelle, ça doit ressembler à ça :



Afin de gagner du temps, nous pouvons ré-utiliser le rectangle créé dans le dernier exercice. Ouvrons son halo, et utilisons l'icône  **Mise à l'échelle** icon pour changer sa taille.

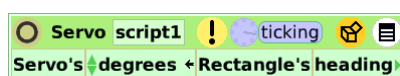


Maintenant, il faut ouvrir le visualisateur du servo-moteur et créer un nouveau script avec la brique `servo.degrés`. Nous voulons modifier cette valeur, donc c'est un assignement et nous devons attraper la brique par sa  flèche.

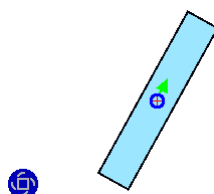


Ensuite, il faut ouvrir le visualisateur du rectangle, et dans la catégorie **Briques de base**, prendre la brique `cap`.

Nous pouvons démarrer le script.



Pendant que le script tourne, ouvrons le halo du rectangle. Et utilisons l'icône  **Faire tourner** icon pour faire tourner le rectangle.



Si tout s'est bien passé, le servo-moteur devrait tourner en même temps que le rectangle.

7 Conclusion

Voilà, c'est tout ce dont vous avez besoin pour commencer à utiliser la carte Arduino. Les possibilités d'interaction entre l'ordinateur, Arduino et les autres composants électroniques que fournit Physical Etoys sont très nombreuses : nous vous encourageons à découvrir les autres fonctions et composants en explorant le reste de l'environnement Squeak (et pour celà, rien de tel que tester, jouer, toucher et même en casser si besoin !).

Amusez-vous bien !

Glossaire

halo C'est l'ensemble des icônes qui entourent un objet quand on fait un clic droit dessus. 5

photo-résistance Une photo-résistance est un petit capteur de lumière dont la résistance varie avec l'intensité lumineuse qu'il reçoit. 12

script Les scripts sont les programmes associés à chaque objet. On peut afficher la liste des scripts d'un objets dans son **visualisateur**, dans la catégorie **Scripts** plural. 5

visualisateur Le visualisateur est la zone, spécifique à chaque objet, qui contient toutes les briques permettant la programmation. On l'affiche grâce à l'icône  **œil** icon du halo de l'objet. Le visualisateur est divisé en *Catégories* qui servent à organiser les briques. 5

GIRA 2011

Planète Sciences 2011 pour la version française. Cette documentation est diffusée sous les termes de la licence Creative Commons Paternité-Partage à l'identique.



Les sources LaTeX originales de ce guide peuvent être téléchargées sur GitHub.
<http://github.com/GIRA/Physical-Etoys>