



Il existe une version de SCRATCH pour l'Arduino. Elle permet de piloter un Arduino à partir du code SCRATCH et de ce fait rend accessible à tout public la programmation d'un robot à partir d'un environnement aussi ludique, visuel et intuitif que celui de SCRATCH.

Cette fiche permet de s'initier à la programmation avec SCRATCH pour la robotique, de façon ludique et à l'aide de 4 montages :

1. la réalisation d'un minuteur, permettant d'alerter sur le temps restant selon 3 niveaux : vert, orange rouge ;
2. l'utilisation d'un capteur, une photorésistance et le traitement de son signal selon différents cas de figure ;
3. la commande d'un servomoteur pour vider un bac de ramassage (par exemple) et revenir à sa position de départ, lors qu'on appuie sur la barre espace ;
4. la commande de deux servomoteurs d'un bras robotisé, un pour le poignet et l'autre pour la pince ; à l'aide des touches du clavier.

Pour chaque application, il est recommandé de suivre le pas à pas suivant :

1. mettre l'arduino sous tension et le charger avec le firmware pour Scratch
2. lancer S4A
3. faire le montage électronique avec la breadboard ou en soudant ou liant les composants
4. créer le programme dans S4A
5. Expérimenter et changer quelques valeurs ou instructions pour voir ce que cela fait.

Le CRVG remercie le Centre d'Arts Sensitifs (craslab.org) pour son tutoriel et l'équipe espagnole du Citilab (<http://seaside.citilab.eu/scratch>) pour la fourniture de la version de SCRATCH For Arduino (S4A).

L'installation de cet environnement spécifique est décrit dans une fiche du CRVG.

Application 1 – Un minuteur lumineux

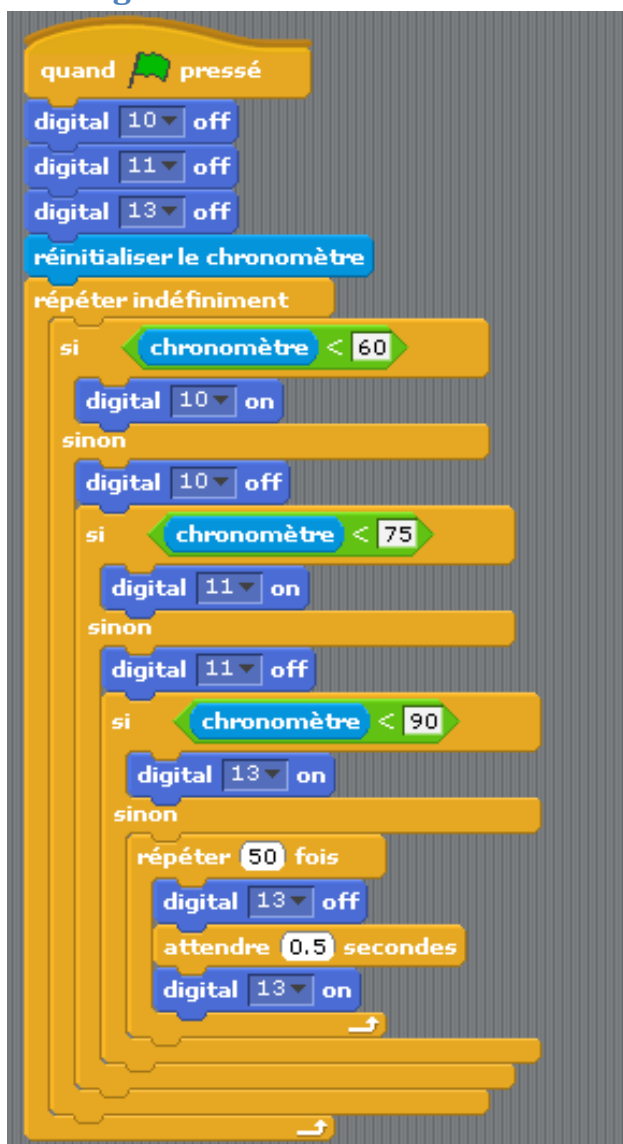
On imagine une aide précieuse et visuelle apportée le jour d'une compétition par un petit dispositif assez simple basé sur 3 LED placées en haut du robot :

- verte : allumée dans la première minute, éteinte ensuite ;
- jaune ou orange ; allumée de 1 minute à 1 minute 15, éteinte ensuite ;
- rouge : clignotante de 1 minute 15 à 1 minute 30 clignotante ensuite.

1 – Montage

Il est très simple : on monte les LED avec la cathode sur GND et l'anode reliée à une résistance qui est connectée au pin 11, 12 et 13.

2 – Programme S4A

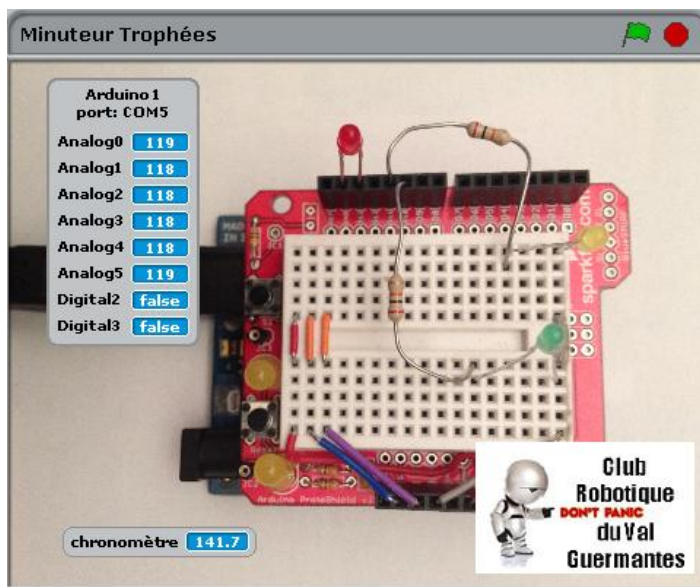


Après avoir initialisé les pin de sortie pour les LED et remis le chronomètre à zéro, on entre dans une boucle infinie.

Ensuite, 3 conditions en cascade, c'est-à-dire que si la première n'est pas ou plus satisfaite, alors on passe à sa clause « sinon » et ainsi de suite. Chaque fois qu'une condition n'est plus satisfaite, on passe donc à la séquence suivante et on éteint la LED précédente.

Au final, on fait clignoter la LED rouge cinquante fois toutes les demi-secondes .

Une instruction « switch » aurait fait la même chose en étant plus compacte.



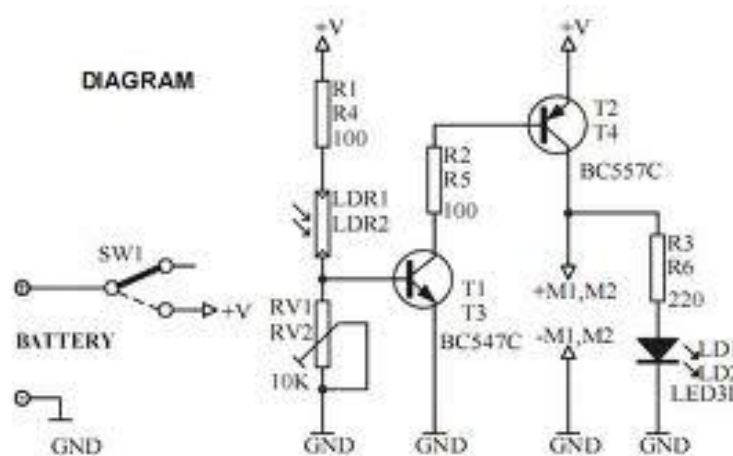
3 – Expérimentation

L'effet n'est pas très spectaculaire mais en imaginant les conditions d'une compétition, cette aide visuelle trouve son utilité. Essayer de modifier les paramètres de durée et de clignotement.

Application 2 – Un capteur et son comportement : une photorésistance

Le montage est très simple et le programme est plutôt facile. Il permet d'obtenir le même comportement que le montage à transistor du kit Velleman « Microbug ». Les LED simulant les moteurs.

L'avantage du programme est dans le changement de comportement par une modification du code alors que dans le cas du montage électronique, il faut modifier le circuit.



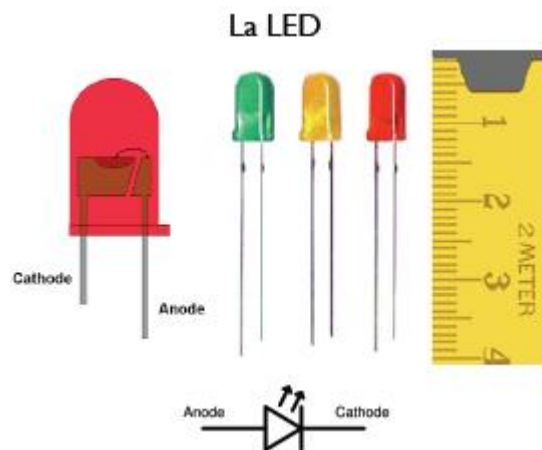
1 – Montage

Remarque : on ne câble JAMAIS directement une LED sur l'alimentation +5V : on met toujours une résistance en série avec elle. Exception : le pin 13 est monté avec une résistance en série sur le circuit de l'Arduino (pas dans le microcontrôleur Atmel).

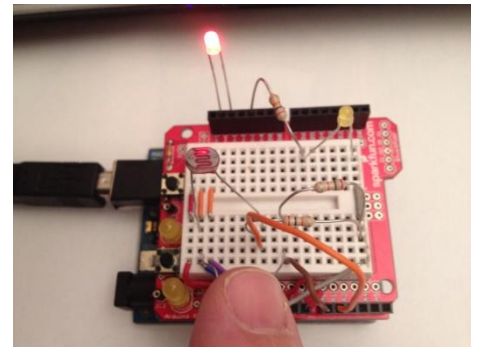
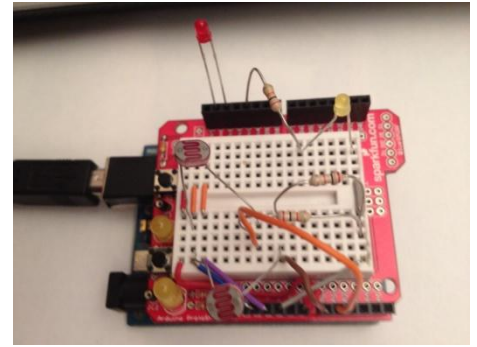
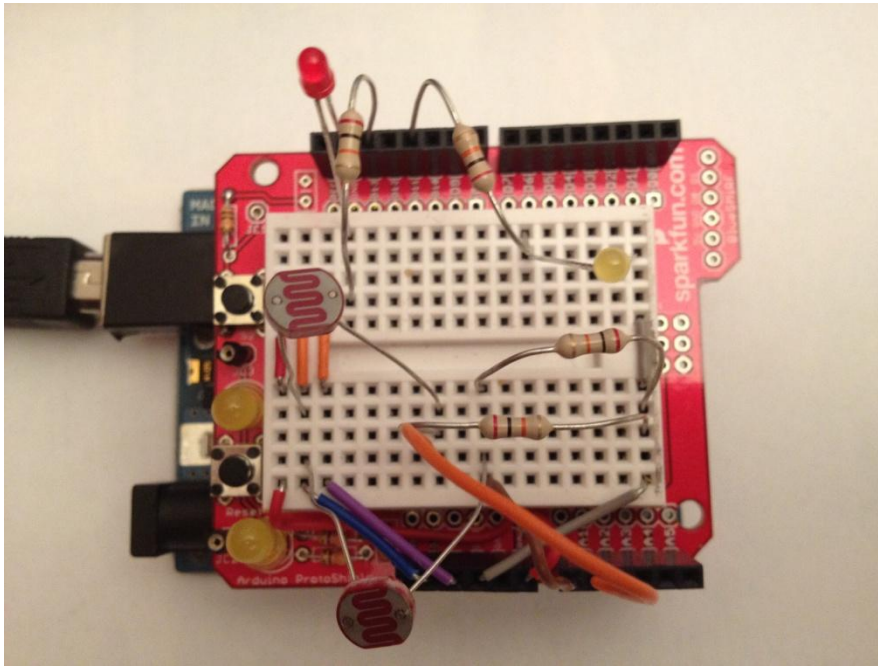
Pour le montage, les consignes sont les suivantes

En entrée du montage, donc pour les capteurs, les photorésistances ont une patte sur le +5V et la seconde reliée en même temps à une résistance et à un pin analogique. Par exemple, la photorésistance sur le pin A0 sera la photorésistance 1. On connecte donc les deux photorésistances de la même façon et leur valeur sera lue sur les pin A0 et A1.

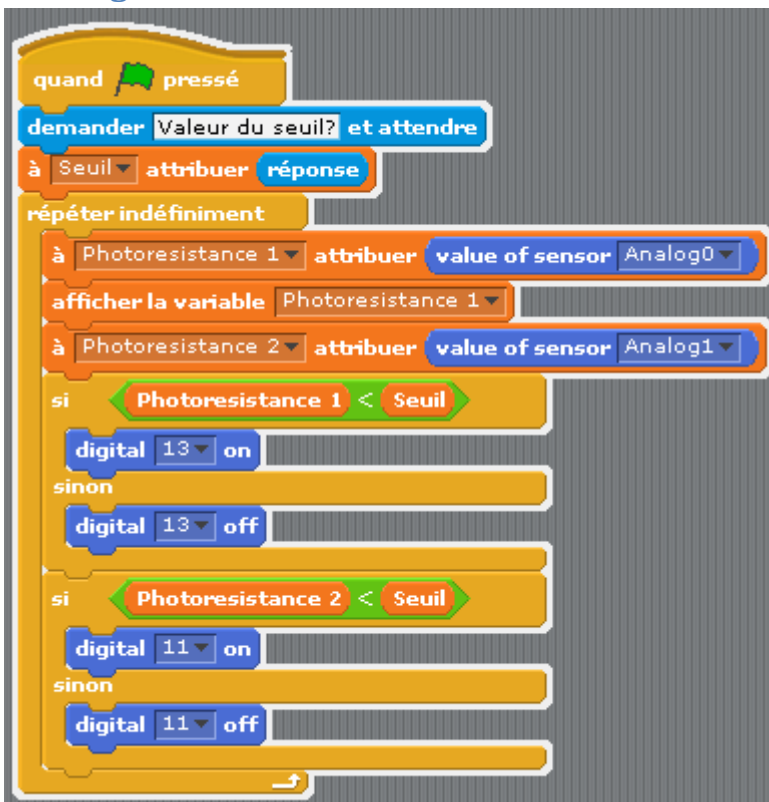
En sortie du montage, donc pour les actionneurs, les LED ont leur patte négative (cathode, la plus courte) reliée au pin GND (GROUND ou plus petite valeur) tandis que leur patte positive (l'anode, patte la plus longue) est reliée à une résistance qui est connectée au port de sortie. Sur la photo ci-dessous, la LED en haut à gauche est reliée via une résistance au pin 13. On peut simplifier le schéma en mettant directement la patte cathode dans le pin GND et la patte anode dans le pin 13 (car il y a une résistance sur le pin 13). Notez la différence de luminosité quand on se branche directement sur le pin 13 car le courant est plus fort.



Source craslab.org



2 - Programme S4A



On utilise une boucle infinie, cas d'un automate qui scrute la valeur d'un capteur pour déterminer une action à faire

L'algorithme est très simple est fait intervenir une condition : si la valeur de la photorésistance est inférieure à un seuil alors on fait une action sinon on fait une autre action.

On peut décider d'actions différentes pour chaque photorésistance.

On a donc 3 variables que l'on peut programmer :

- la valeur du seuil qui dépendra de la réponse à la question posée quand on souhaite que l'utilisateur fixe lui-même la valeur ;
- la valeur de chaque photo résistance telle qu'elle est lue par les pin analogiques de l'Arduino

3 - Expérimentation

photoresistance et LED

Arduino 1 port: COM5	
Valeur du seuil?	
Analog1	427
Analog2	420
Analog3	414
Analog4	407
Analog5	401
Digital2	false
Digital3	false

Seuil 400

Photoresistance 1 436 Photoresistance 2 424

Club Robotique
DON'T PANIC
du Val
Guermantes

L'instruction « Demander » ouvre une boîte de dialogue qui permet de saisir la valeur à attribuer à la variable « Seuil », ici 400.

On lit les valeurs des pin analogiques et les valeurs des variables.

Le code correspondant à la branche « si » de la condition, c'est-à-dire si elle est vraie, sera exécuté en fonction de la comparaison entre la valeur de chaque photorésistance et la valeur du seuil.

Dans le montage microbug, la valeur du seuil était fixée par un potentiomètre.

photoresistance et LED

Arduino 1 port: COM5	
Analog0	179
Analog1	440
Analog2	433
Analog3	425
Analog4	417
Analog5	409
Digital2	false
Digital3	false

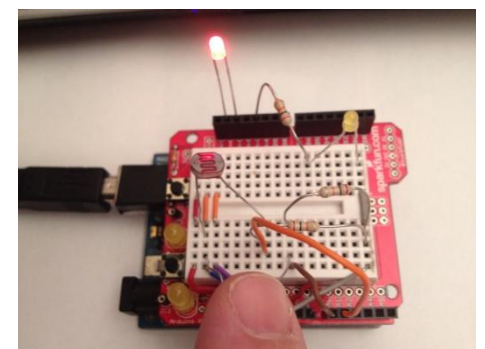
Seuil 400

Photoresistance 1 179 Photoresistance 2 440

Club Robotique
DON'T PANIC
du Val
Guermantes

Si on masque la photorésistance 1 alors la diode branchée sur le pin 13 est allumée

On remarque que la valeur de la photorésistance 1 est bien au dessous de la valeur du seuil.

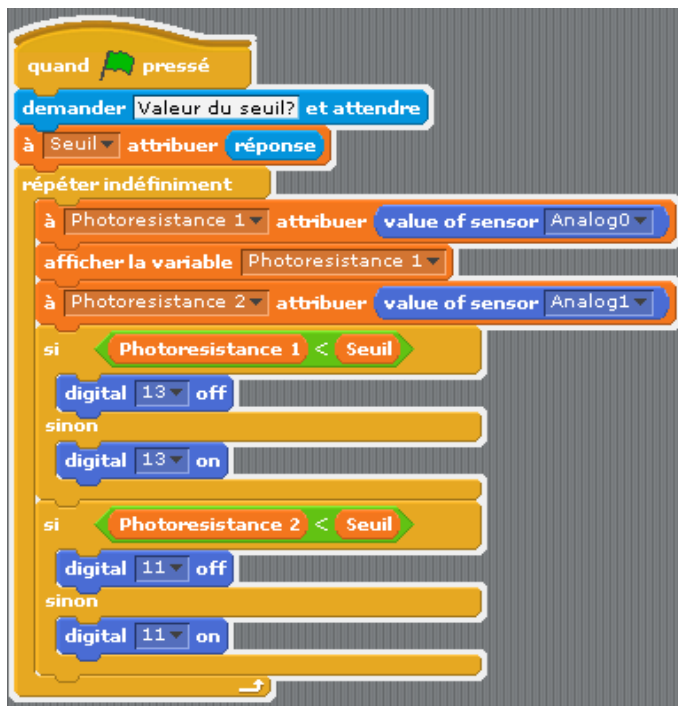


Evolution du montage : comment inverser le comportement en éteignant les LED lorsque la lumière descend en dessous du seuil ?

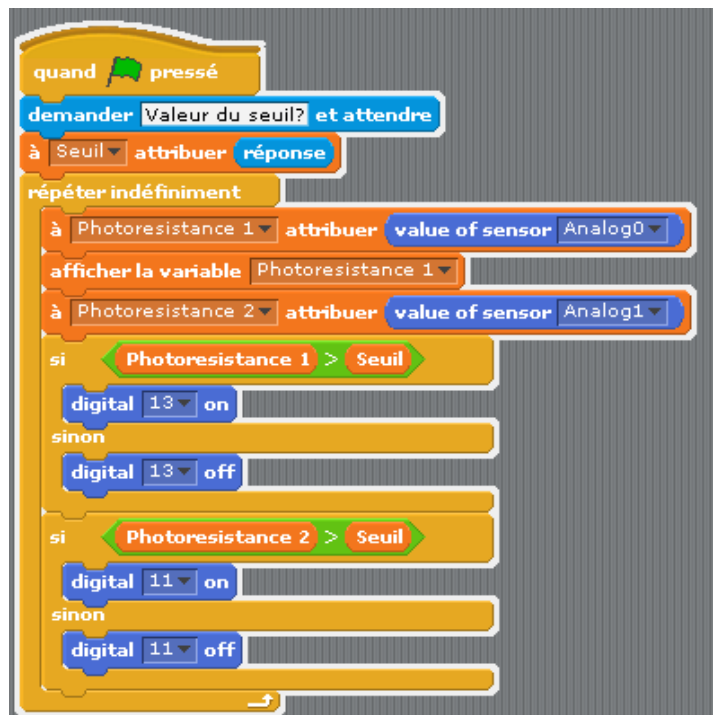
Votre réponse :

Solution : il suffit :

soit d'invertir les actions



soit de modifier le sens de la condition



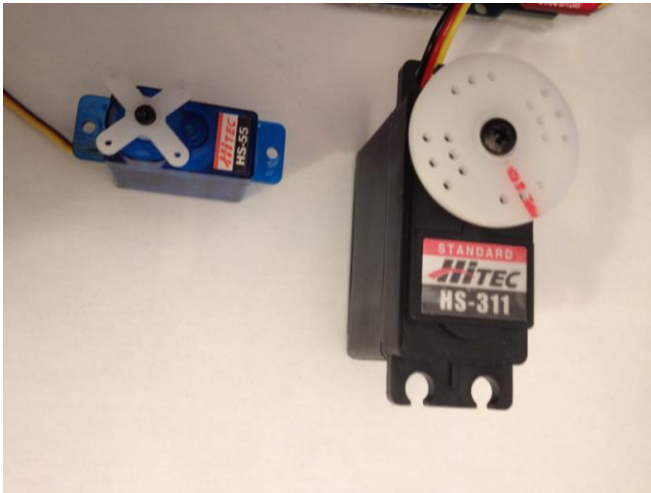
Application : ce comportement est bien adapté au pilotage automatique d'un robot suiveur de ligne : au lieu d'allumer des LED, le programme ajuste la puissance des moteurs de façon à ce qu'il suive la lumière (bande de couleur claire entre les deux capteurs) ou ce qu'il perçoit comme sombre (bande de couleur noire).

Application 3 – Vider un bac de ramassage

Un robot ramasse des objets à l'aide d'une pince et les dépose dans un bac situé sur son châssis. Quand le bac est plein, l'appui sur un bouton déclenche une séquence permettant de vider le bac par une rotation de 0 à 105° (un peu plus de 90° pour être sûr que tous les objets qu'il contient vont glisser en dehors du bac), attendre 2 secondes puis revenir à sa position initiale.

1 – Montage

On utilise un servomoteur qui permet une rotation de 0 à 180°. Il en existe de plusieurs sortes. Par exemple, mini et standard comme sur la photo ci-dessous :



Un servomoteur a toujours 2 fils d'alimentation (rouge positif et noir négatif) et un fil (jaune) pour la commande. Il est connecté à un pin de l'Arduino.

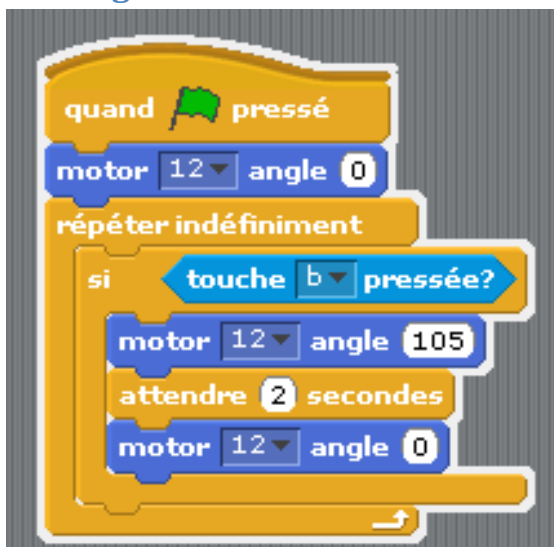
Le servo moteur



Le servo-moteur est un moteur (rotatif) qui peut effectuer des rotations très précises (dans une portion de tour seulement) et en un certain nombre de pas (de micro-déplacements). Il y a toutes sortes de servo moteurs. Un des avantages des servo moteurs est sa possibilité de maintenir avec force une position donnée. On peut piloter des rotations avec l'Arduino, quelques fois directement avec la carte si le moteur n'est pas trop puissant, sinon en passant par un montage associé.

Source craslab.org

2 – Programme S4A



3 – Expérimentation

Elle est assez triviale car la séquence est très répétitive...

Le programme est simplissime.

On retrouve une boucle infinie qui permet de scruter le clavier pour détecter l'appui sur la touche « B » (comme bac).

Elle déclenche une simple séquence affectant 105° comme valeur à l'angle du servomoteurs, afin de faire une rotation plus grande qu'un angle droit et de le vider complètement.

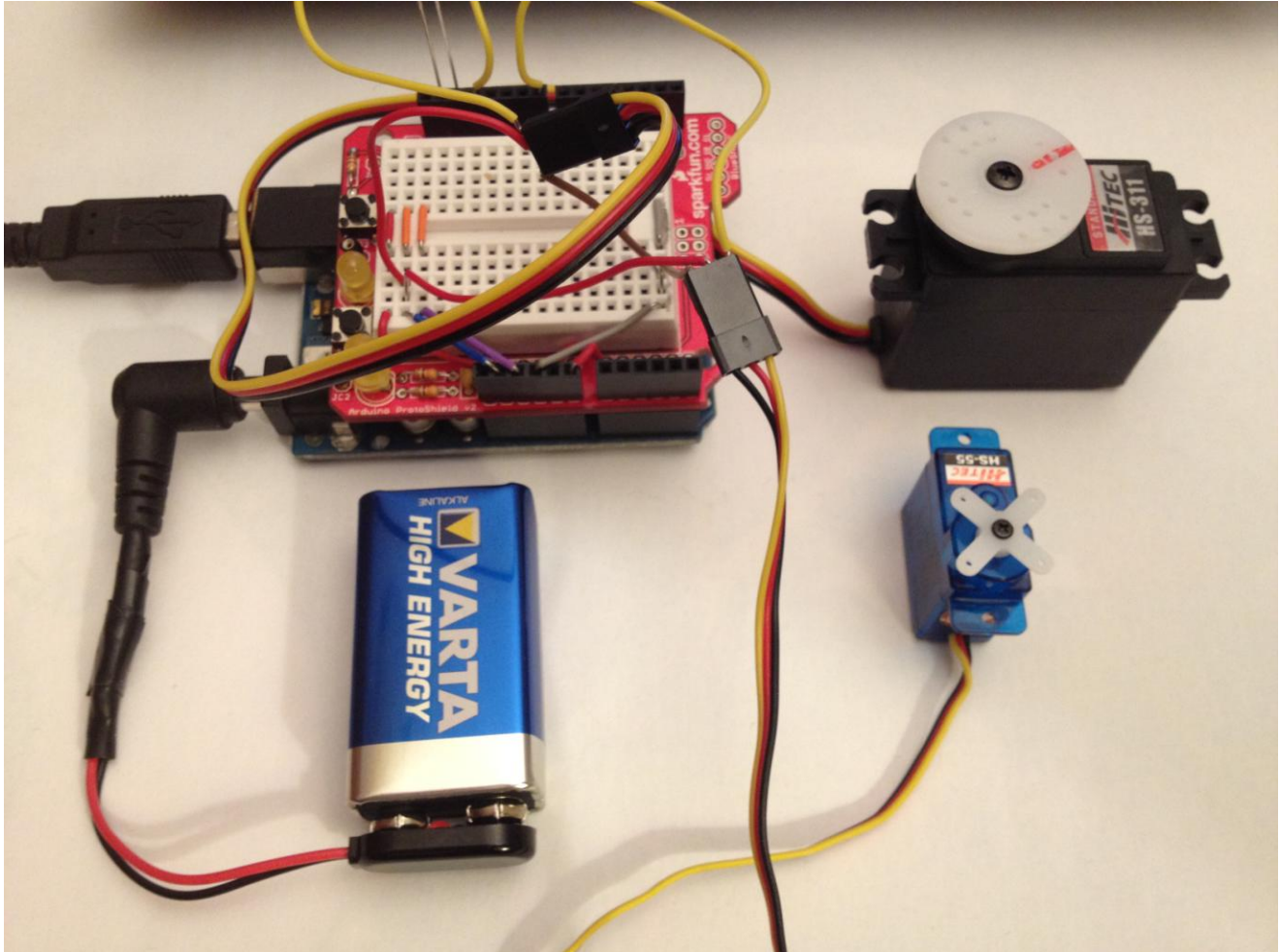
Puis, après une attente de 2 secondes, on envoie un ordre de retour à la position de départ, soit 0°.

Application 4 – Commander les servomoteurs d'un bras robotisé

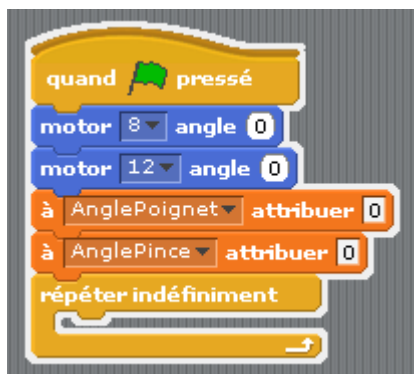
Un bras robotisé fait intervenir plusieurs servomoteurs pour animer ses articulations.

1 – Montage

Il fait appel à 2 servomoteurs (poignet + pince). Le câble de commande est connecté au pin 12 pour le poignet et 8 pour la pince. L'Arduino est alimenté avec une source externe pour ne pas risquer de dégrader le port USB de l'ordinateur



2 – Programme S4A

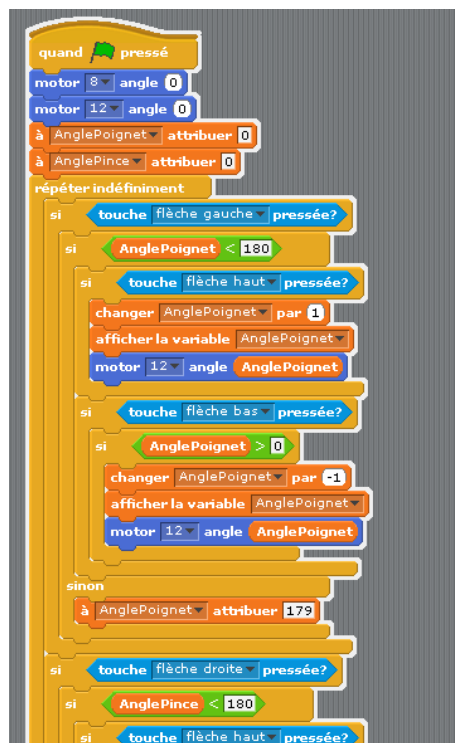


Après initialisation des 2 servomoteurs à leur valeur d'angle de départ, on utilise une boucle infinie, pour capturer les événements au clavier.

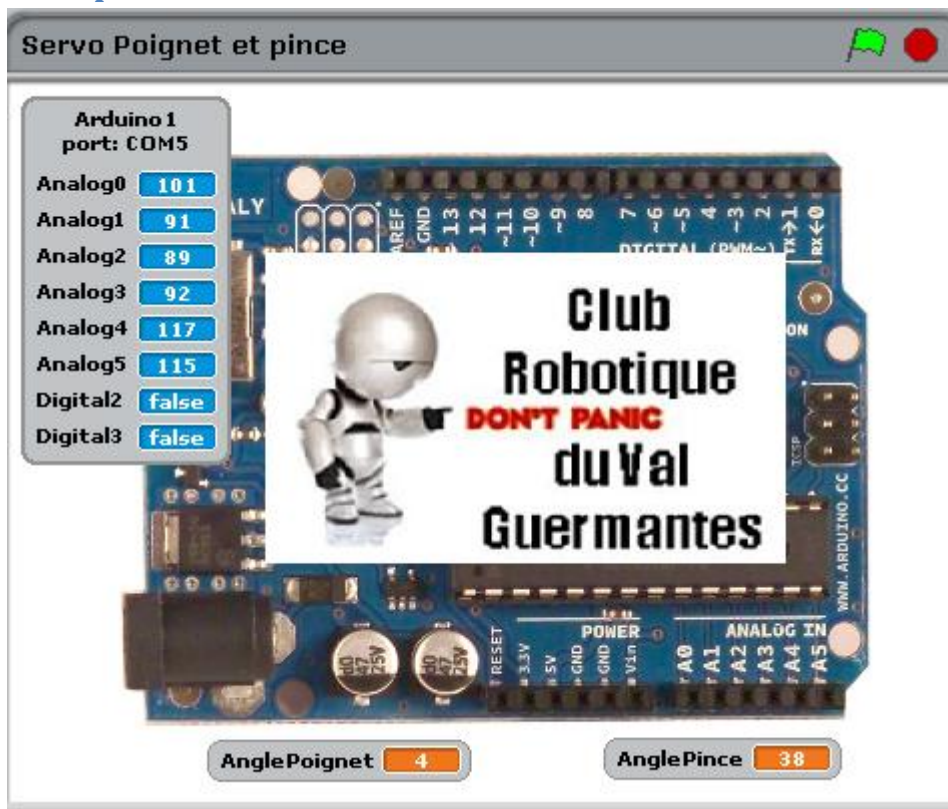
Selon qu'on appuie sur la touche gauche ou droite, on obtient le même comportement mais pour un servomoteur différent (dans le programme, seuls le nom de la variable et le numéro de pin changent).



Le bloc « motor » permet de spécifier l'angle pour un servomoteur branché sur le pin indiqué. Lorsqu'on appuie sur la flèche haut, on augmente la valeur de cet angle avec un incrément de 1. On contrôle que la valeur de l'angle ne soit pas supérieure à 180 ni inférieure à 0, auxquels cas, elle est fixée à la valeur mini ou maxi.



3 - Expérimentation



En sélectionnant le servomoteur par un appui sur la flèche gauche du clavier (poignet) ou la flèche droite (pince), on peut faire varier sa rotation de 0 à 180° à l'aide des touches haut et bas.

L'angle courant de chaque servomoteur est affiché dans l'interface.

Ce principe se généralise pour n servomoteurs dans la limite de ce que le microcontrôleur peut supporter.

Noter qu'en appuyant sur les touches gauche et droite en même temps, on sélectionne les deux servomoteurs pour leur appliquer la même commande de rotation.

*

* *