

Exemples Scratch Marty

Ce document montre une sélection de choses que vous pouvez faire avec Marty à travers l'interface Scratch, du code très simple au plus complexe.

J'ai fait quelques Vidéos courtes montrant les effets qu'ont les scripts sur Marty. Ils ne sont vraiment pas fini (et toutes mes excuses pour la musique de Noël en fond), mais vous pouvez voir à quoi ressemble chaque script lorsqu'il est lancé :

<https://www.youtube.com/playlist?list=PLiBgsWjIRfInizbzagjLnoQjoeb0psS33>

Nous allons faire des projets pour débutants pour la plupart des exemples, pour les personnes qui débutent. Ils devraient apparaître sur www.robotical.io/scratch rapidement.

Commandes basiques

Habituez-vous à contrôler votre robot Marty avec Scratch en jouant avec les fonctions intégrées.

Contrôler Marty

Le bloc "Get Ready" redonnera sa position initiale à Marty et lui fera bouger les sourcils. Le bloc "Turn off motors" coupera les moteurs. C'est mieux de le placer à la fin du script.



Chaîne de commande plus longue ([Vidéo](#))



Amusez en faisant faire des choses plus compliquées à votre robot Marty. Faites une petite danse ou un jeu.

Boucle ([Vidéo](#))



Parfois vous voulez faire la même chose plusieurs fois, mais copier/coller le code ne fait pas propre et prends beaucoup de temps. Amusez-vous avec le bloc “repeat” afin de vous habituez aux boucles.

Définir une fonction ([Vidéo](#))



L'autre façon afin de ne pas répéter du code est de créer une “fonction”

Faites marcher Marty !

Il y a une fonction de marche déjà intégré, mais ce n'est pas drôle. Pour ce challenge, vous devez apprendre à Marty à marcher. Vous allez devoir penser à l'équilibre et comment chaque articulation doit bouger afin de créer un mouvement complexe.

Faites un pas ([Vidéo](#))



Le mouvement de base pour faire un pas - dans ce cas là, avec la jambe droite.

Faites-en deux ([Vidéo](#))



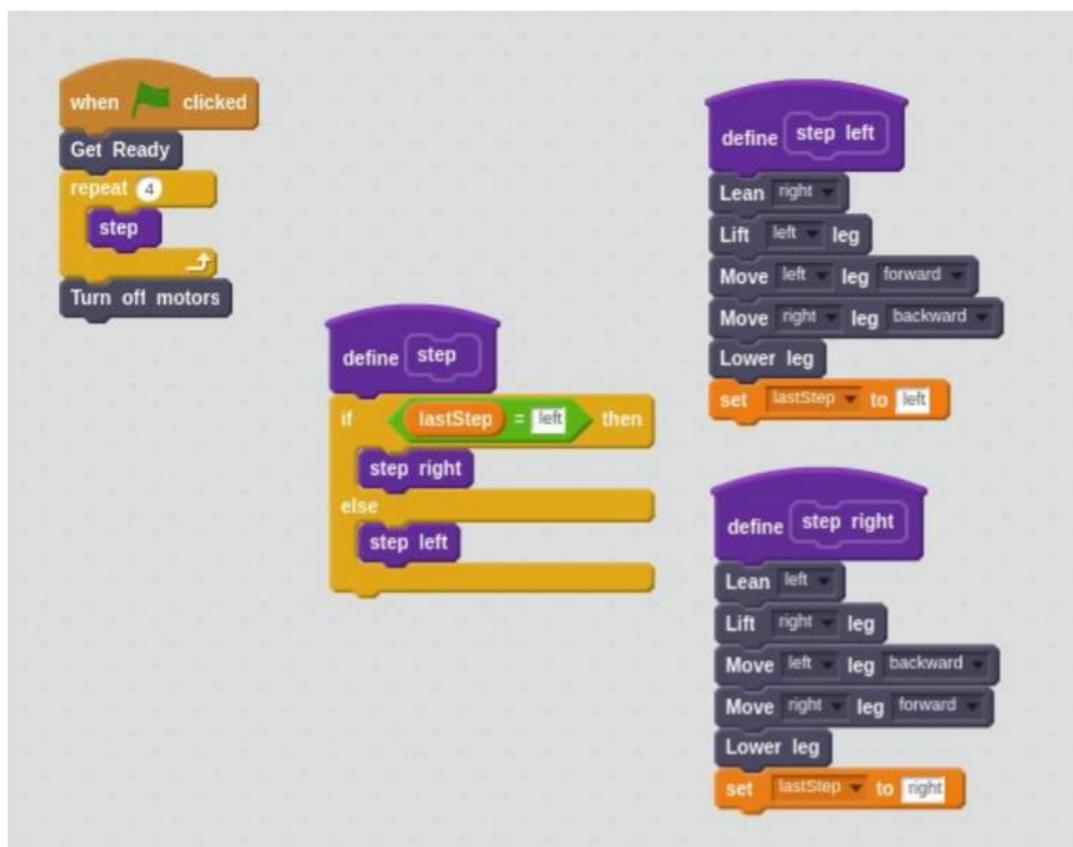
Et puis ajouter un pas avec l'autre jambe.

Fonction Pas ([Vidéo](#))



Déplacez chaque pas individuel dans leur propre fonction.

Mettez le meilleur pied devant ([Vidéo](#))



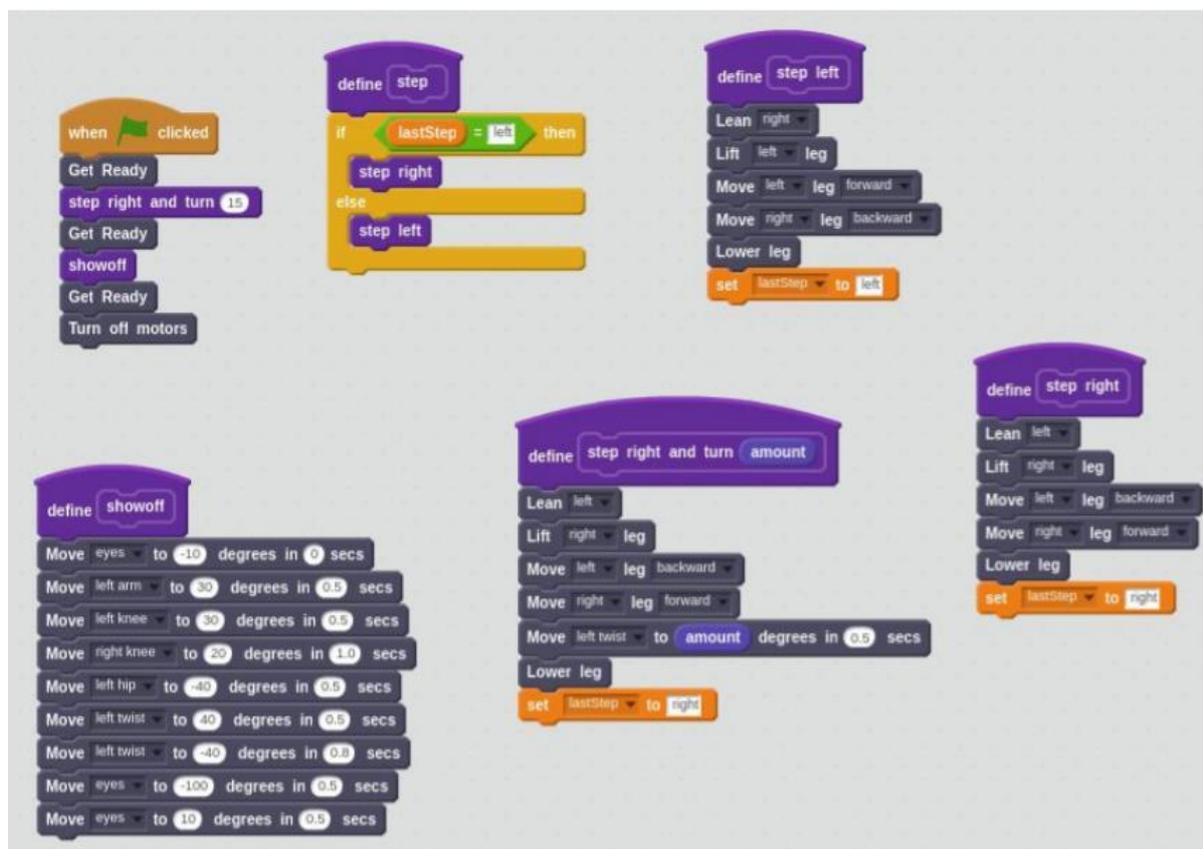
Ce serait mieux de simplement dire "step" plutôt que de dire spécifiquement quelle jambe doit bouger. Cela peut être fait en créant une variable afin de garder une trace de quel pied a été utilisé en dernier.

Faites vos propres mouvements ([Vidéo](#))



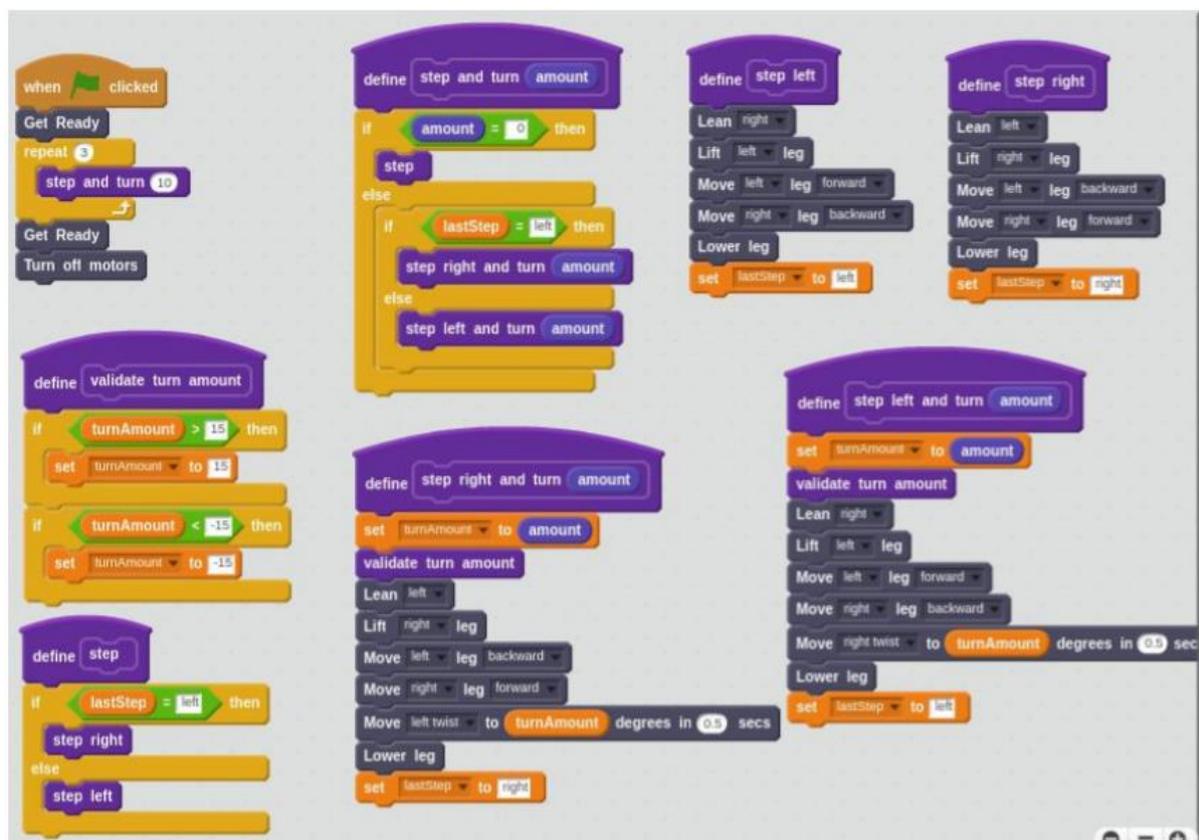
Amusez-vous avec le bloc “Move Joint” afin de créer une fonction de mouvement plus compliquée.

Un tournant décisif ([Vidéo](#))



Créez une fonction indiquant à quel pas tourner, cette fonction peut avoir un argument indiquant de combien le robot doit tourner.

Paramètres de validation, et fonction-ception ([Vidéo](#))



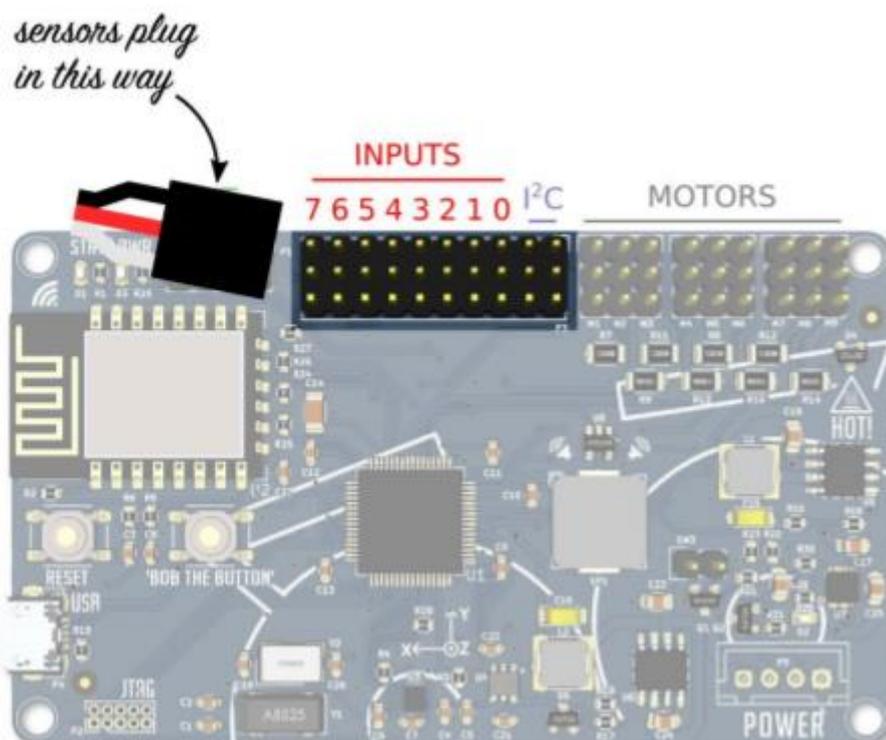
Il y a plusieurs choses qui se produisent dans ce code :

- Paramètres de validation : Si vous dites à Marty de trop tourner, ses pieds se heurteront l'un à l'autre et ce ne sera pas joli-joli. Afin d'éviter d'envoyer une mauvaise commande, on peut "limiter" la dose du virage.
- Faire des fonctions plus générales. Les fonctions peuvent appeler d'autres fonctions - dans ce cas la fonction "step and turn" englobe la fonction "step", en plus d'ajouter quelques fonctionnalités.

Capteurs

Le robot Marty a plusieurs capteurs intégrés - capteur de courant électrique sur la plupart des moteurs, accéléromètre pour calculer l'inclinaison et l'accélération, et un set de ports auxquels des capteurs comme des capteurs de contact peuvent être ajoutés.

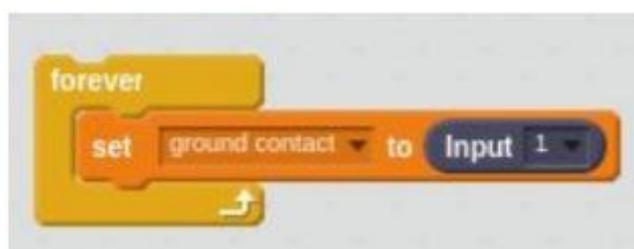
Le schéma de câblage électronique ci-dessous montre comment les capteurs peuvent être fixés, mais ces exemples supposent qu'il y a un capteur de contact fixé au port 0 qui se trouve à l'avant d'un pied, et un capteur de contact fixé au port 1 qui en bas d'un pied.



Connecter les capteurs au robot Marty. Les câbles vont avec le fil noir en haut, et les nombres correspondent au bloc "Input *n*" dans Scratch.

Lire un capteur ([Vidéo](#))

Le bloc "Input" vous laissera lire l'état actuel d'un switch. En assignant la lecture d'un capteur à une variable, vous pouvez observer le résultat dans Scratch. Le switch lira "1" lorsqu'il est activé, et "0" lorsqu'il ne l'est pas.



Marcher jusqu'au bord ([Vidéo 1](#)) ([Vidéo 2](#))

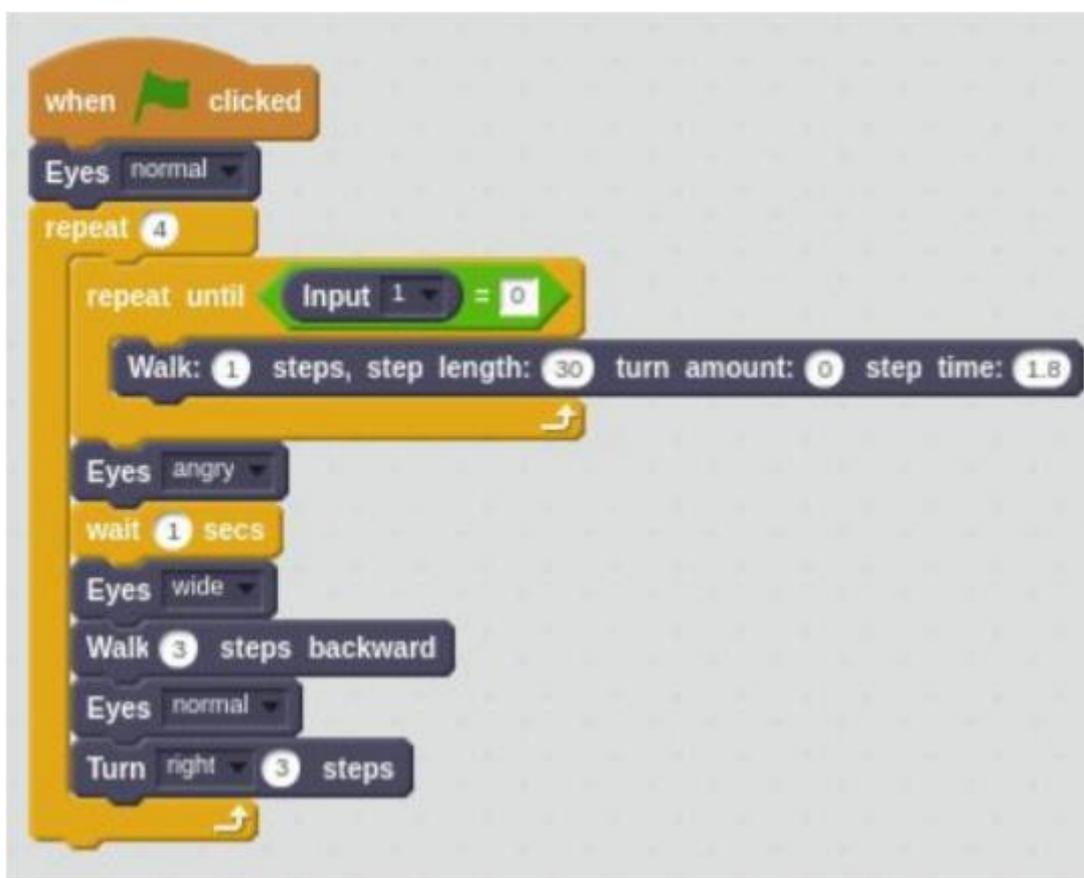
Cet exemple part du principe qu'il y a un capteur de contact sous l'un des pieds et fixé au port 1.

A la fin de chaque pas, le switch peut être vérifié pour voir s'il est activé. S'il est à "1", le pied touche le sol, s'il est à "0" alors il ne le touche pas !

Nous pouvons utiliser ce code pour voir si Marty est arrivé à un rebord.

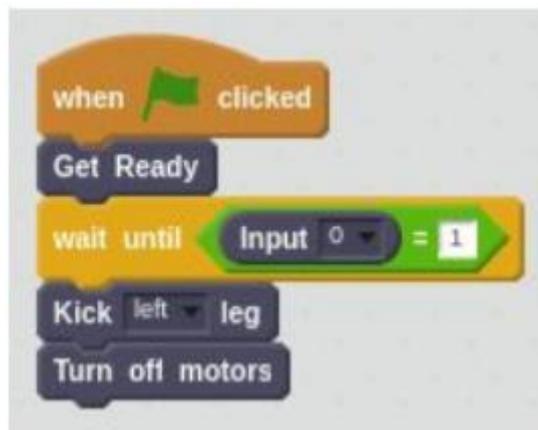


Revenir du bord ([Vidéo](#))



C'est utile de faire autre chose que des réactions. Ajoutez quelques blocs pour faire faire quelque chose d'utile à Marty lorsqu'il arrive à un bord. Dans cet exemple, il reculera et se tournera avant de reprendre sa marche.

Réagir à la pression d'un bouton ([Vidéo](#))



Il s'agit d'un exemple simple d'une réaction à un bouton poussé. Marty attendra jusqu'à ce que quelque chose touche le capteur de contact et donnera un coup de pied.

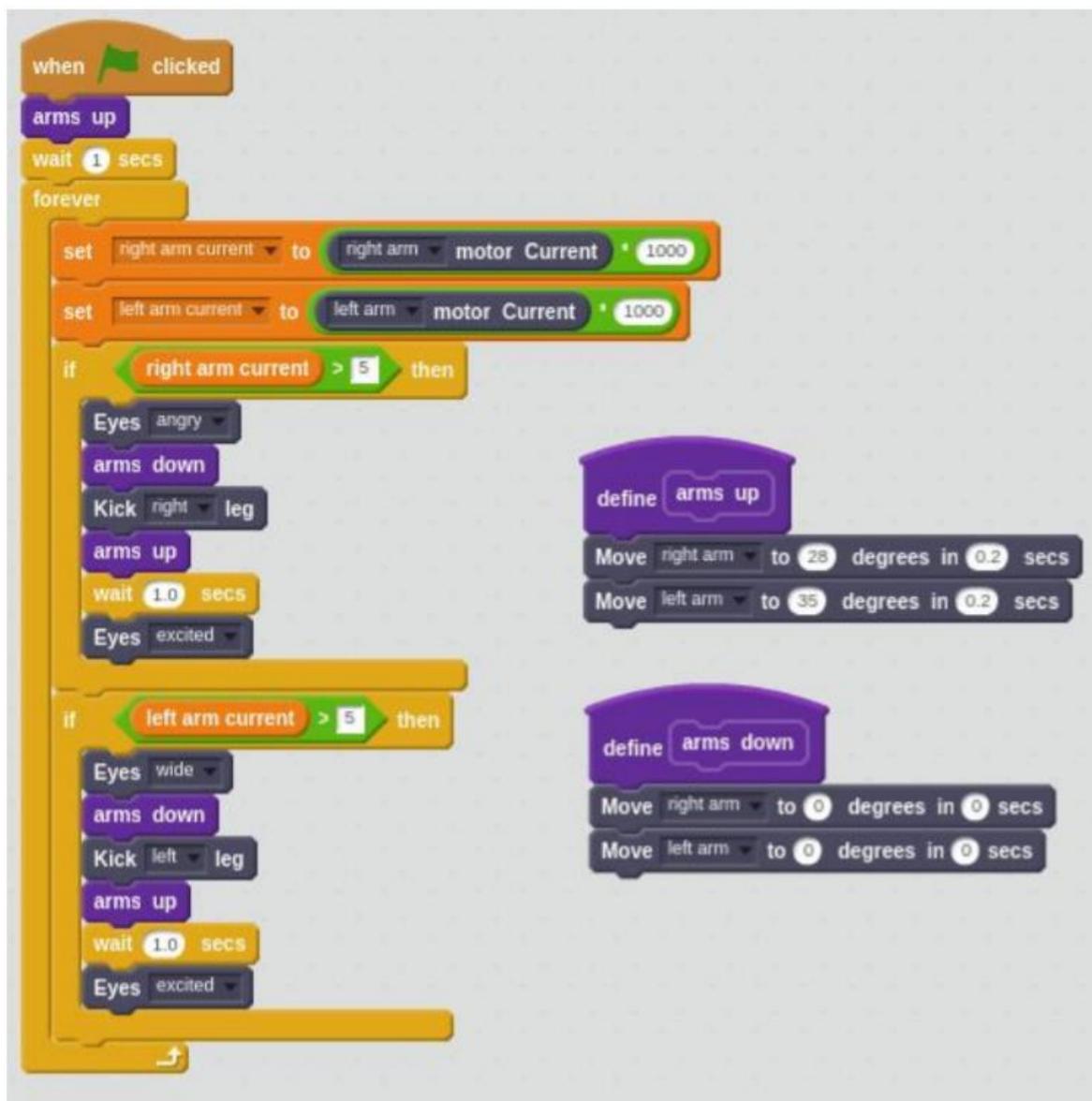
Lire le courant d'un moteur ([Vidéo](#))



Avoir un aperçu du courant d'un moteur vous laisse voir à peu près combien de couple un moteur produit. Le capteur renvoie une petite valeur, vous devrez donc la multiplier comme montré ci-dessus.

Dans cet exemple, nous lions le couple du bras droit à la position du sourcil - poussez le bras de Marty ou mettez quelque chose de lourd dessus et il se mettra en colère !

Utiliser le courant du moteur comme entrée ([Vidéo](#))



```

when clicked
  arms up
  wait 1 secs
  forever
    set right arm current to right arm motor Current * 1000
    set left arm current to left arm motor Current * 1000
    if right arm current > 5 then
      Eyes angry
      arms down
      Kick right leg
      arms up
      wait 1.0 secs
      Eyes excited
    if left arm current > 5 then
      Eyes wide
      arms down
      Kick left leg
      arms up
      wait 1.0 secs
      Eyes excited

define arms up
  Move right arm to 28 degrees in 0.2 secs
  Move left arm to 35 degrees in 0.2 secs

define arms down
  Move right arm to 0 degrees in 0 secs
  Move left arm to 0 degrees in 0 secs
  
```

Dans cet exemple, nous laissons quelqu'un donner des instructions à Marty en tirant l'un de ses bras vers le bas. Tirer le bras droit lui fera donner un coup de pied avec sa jambe droite, et tirer le bras gauche lui fera donner un coup de pied avec sa jambe gauche.

Comportement AND ([Vidéo](#))

```

when clicked
  arms up
  wait 1 secs
  forever
    set right arm current to right arm motor Current * 1000
    set left arm current to left arm motor Current * 1000
    if left arm current > 5 and right arm current > 5 then
      Wiggle
    else
      if right arm current > 5 then
        Eyes angry
        arms down
        Kick right leg
        arms up
        wait 1.0 secs
        Eyes excited
      if left arm current > 5 then
        Eyes wide
        arms down
        Kick left leg
        arms up
        wait 1.0 secs
        Eyes excited
  
```

define arms up

Move right arm to 28 degrees in 0.2 secs

Move left arm to 35 degrees in 0.2 secs

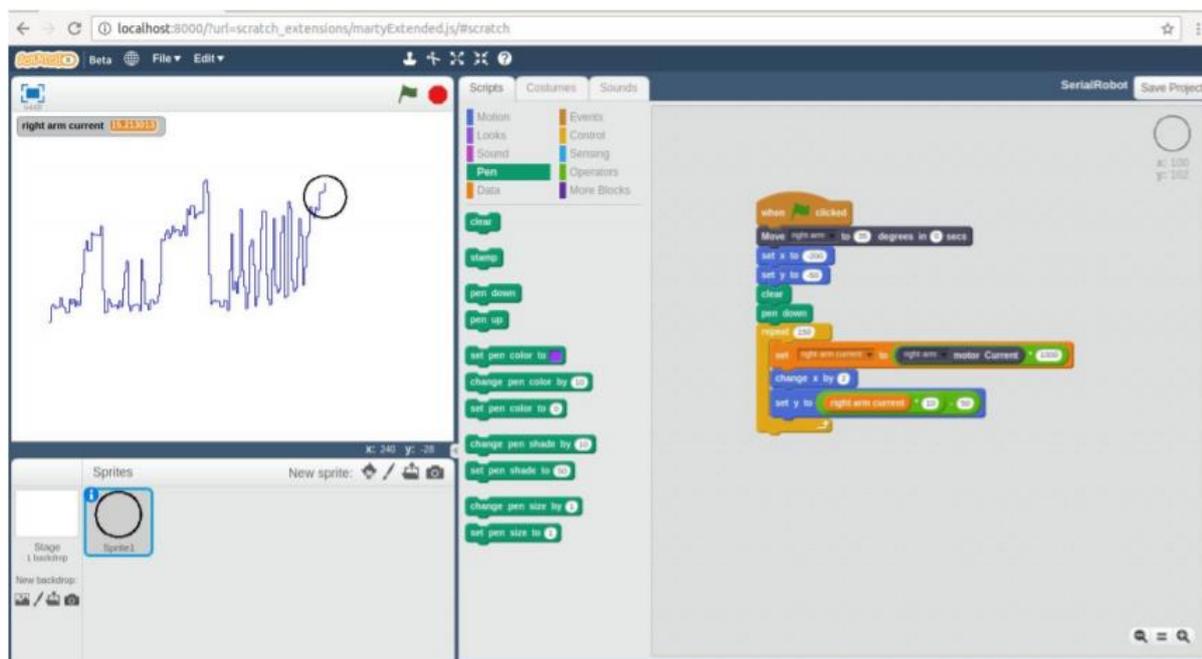
define arms down

Move right arm to 0 degrees in 0 secs

Move left arm to 0 degrees in 0 secs

Si vous voulez être vraiment dans le tape-à-l'œil, vous pouvez dire à Marty de faire quelque chose de différent lorsque *les deux* bras sont tirés en même temps.

Obtenir des graphiques ([Vidéo](#))



Scratch est aussi cool pour sa capacité à afficher des lectures de capteurs. Dans cet exemple, nous montrons la sortie de l'un de capteur de courant du moteur en tant que graphique.

Pour faire ceci, nous créons un sprite qui est un simple cercle, puis on l'anime avec un stylo.