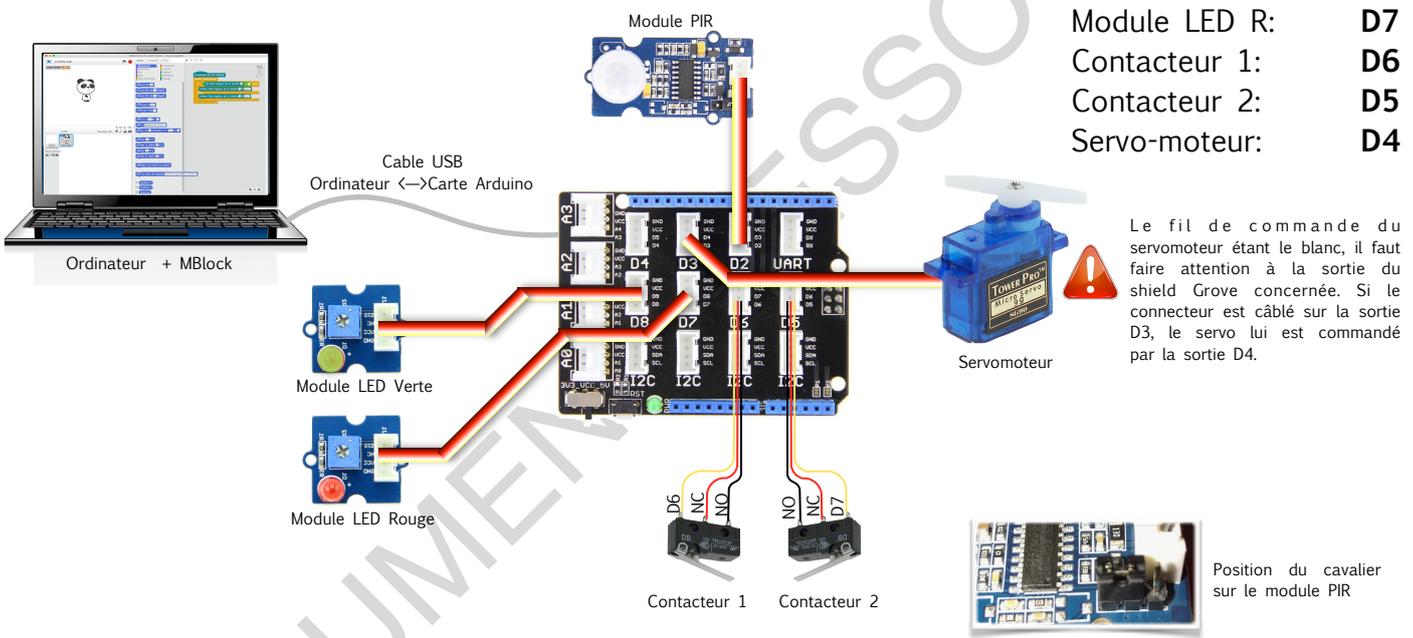


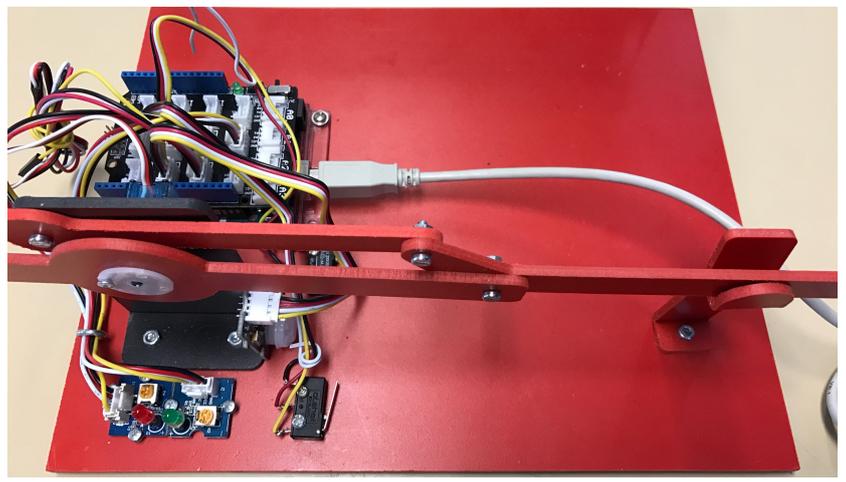
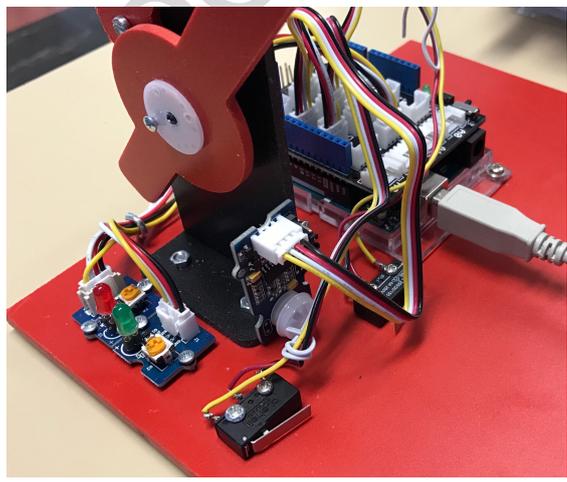


- Matériel nécessaire:
- 1 ordinateur avec MBlock installé et configuré,
  - 1 carte Arduino Uno
  - 1 Shield GROVE
  - 1 Module Grove PIR,
  - 1 module Grove LED Verte,
  - 1 module Grove LED Rouge
  - 2 contacteurs,
  - 1 barrière équipée avec son servomoteur.

**CABLAGE ET CONNECTIONS**



**MAQUETTE**



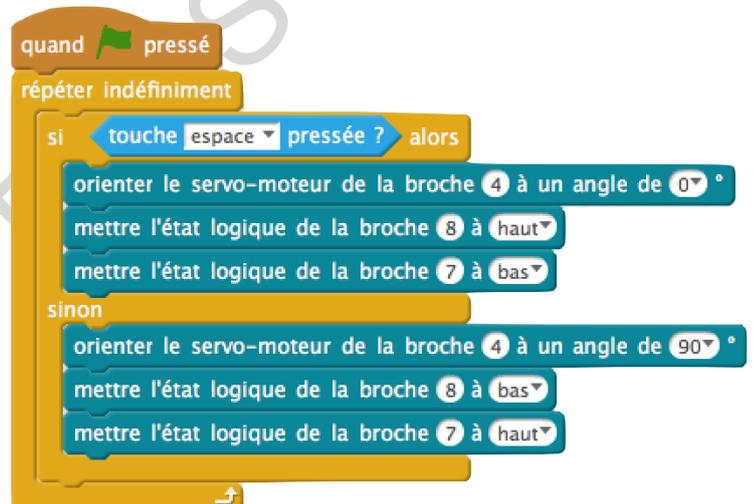
### Programme N°1

Scénario: On demande ouvrir la barrière avec la barre « espace » du clavier. Si la barre espace est maintenu, la barrière reste ouverte, si on relâche la barre la barrière se ferme.



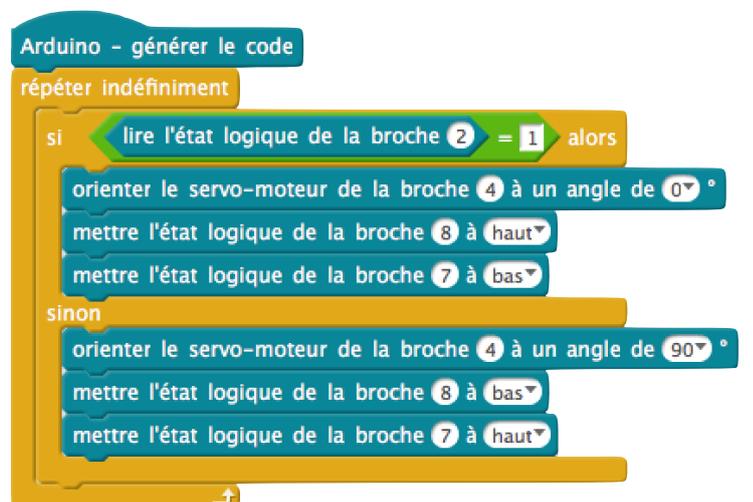
### Programme N°2

Scénario: Programme 1 avec ajout de LEDs; la LED Verte s'allume lorsque la barrière est ouverte, la LED rouge s'allume lorsque la barrière est fermée.



### Programme N°3

Scénario: La barrière s'ouvre si elle détecte une personne à l'aide du module Grove PIR. Les LEDs verte et rouge indiquent l'ouverture et la fermeture.



Ce programme permet à la maquette de fonctionner automatiquement, n'oubliez pas de le téléverser dans la carte Arduino.

### Programme N°4

Scénario: Nous allons simuler 2 barrières infra-rouges (une de chaque côté) avec les contacteurs (1 appui sur le contacteur = détection de la voiture). Lorsqu'une voiture coupe une barrière infra-rouge (de n'importe quel côté), celle-ci s'ouvre, si la voiture coupe à nouveau une barrière infra-rouge, la barrière se ferme. Les LEDs verte et rouge indiquent l'ouverture et la fermeture.

```
Arduino - générer le code
mettre ouverture/fermeture à 0
répéter indéfiniment
si lire l'état logique de la broche 5 = 1 ou lire l'état logique de la broche 6 = 1 alors
  ajouter à ouverture/fermeture 1
  attendre 0.5 secondes
si ouverture/fermeture modulo 2 = 0 alors
  orienter le servo-moteur de la broche 4 à un angle de 90°
  mettre l'état logique de la broche 8 à bas
  mettre l'état logique de la broche 7 à haut
sinon
  orienter le servo-moteur de la broche 4 à un angle de 0°
  mettre l'état logique de la broche 8 à haut
  mettre l'état logique de la broche 7 à bas
```

### EXPLICATIONS COMPLEMENTAIRES

Nous allons créer une variable « ouverture/fermeture » qui permettra de compter le nombre de fois ou nous passons devant une barrière infra-rouge. Considérant la barrière fermée au départ, si le nombre de passage est impair, la barrière s'ouvre; si le nombre de passage est pair, la barrière se ferme.

A chaque début de programme nous initialiserons cette variable à « 0 ».

La fonction « attendre 0,5s » qui suit l'incrémentement de la variable « ouverture/fermeture » est un système anti-rebonds, en effet, cela ne permet de compter qu'une impulsion par tranche de 0,5s, sinon, la variable s'incrémenterait très rapidement et de façon aléatoire (car le comptage est en fait très rapide).

La fonction « Modulo 2 » permet de vérifier que la variable « ouverture/fermeture » est paire ou impaire, et par conséquent de définir si la barrière doit s'ouvrir ou se fermer.

Principe du modulo: La fonction modulo ne fonctionne qu'avec des nombres entiers, elle donne le reste d'une division d'une variable par un nombre. Dans notre application, nous allons diviser la variable « ouverture/fermeture » par 2, la fonction modulo est égal au reste, donc si la variable est pair il n'y a pas de reste, si la variable est impair, le reste est 1:

Exemple= si ouverture/fermeture = 13 →  $13/2=6$  et il reste 1, la barrière s'ouvre,  
si ouverture/fermeture = 14 →  $14/2=7$  et il reste 0, la barrière se ferme.