

# LES SERVOS FONT BOUGER LES CHOSES !

#R1AS13

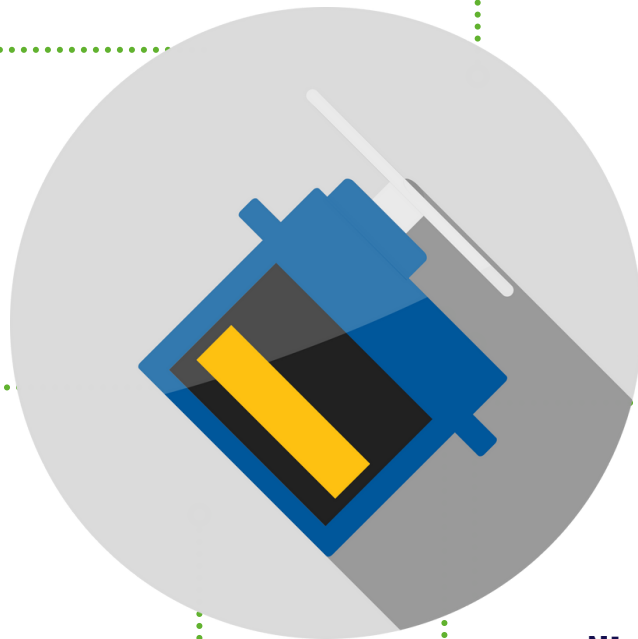


## Disponible sur



## Prérequis

- R1AS03 - Boutons et affichage LED



## De quoi parle-t-on ?

Le servomoteur est un moteur pilotable en position. Il permet de conserver un angle précisément. Il convient pour contrôler un système dont l'angle peut changer au cours du fonctionnement et doit maintenir cette position.

## Durée

25 minutes

## Niveau de difficulté

Intermédiaire

## Matériel

- 1 carte programmable "**STM32 IoT Node Board**"
- 1 câble USB Micro-B
- 1 mini-servo SG-90 (1,6 kg)
- Câbles de connexion

## OBJECTIFS D'APPRENTISSAGE

- Mettre un objet en mouvement





Un servomoteur est un moteur doté d'un ensemble de systèmes de contrôle automatique, qui se compose d'un **moteur ordinaire à courant continu** (moteur électrique rotatif qui convertit l'énergie électrique (courant continu) en énergie mécanique), d'un réducteur, d'un **potentiomètre** (diviseur de tension utilisé pour mesurer le potentiel ou la tension électrique) et d'un circuit de contrôle. Il peut définir l'angle de rotation de l'arbre de sortie en envoyant des signaux. En général, un servomoteur a un angle de rotation maximal (par exemple, 180 degrés). Les servomoteurs sont commandés par l'intermédiaire d'un **câble électrique à trois fils** qui permet d'alimenter le moteur et de lui transmettre des consignes de position sous forme d'un signal codé en largeur d'impulsion plus communément appelé PWM. Cela signifie que c'est la durée des impulsions qui détermine l'angle absolu de l'axe de sortie et donc la position du bras de commande du servomoteur. Le cycle d'un signal de référence servomoteur est de 20 ms. Sur le servomoteur proposé, la position du bras varie de 0 à 180°, pour des largeurs d'impulsion allant de 1 à 2 ms (respectivement).

*Ressources : [https://en.wikipedia.org/wiki/DC\\_motor](https://en.wikipedia.org/wiki/DC_motor), <https://en.wikipedia.org/wiki/Potentiometer>*



## ÉTAPE 1 - CONSTRUIRE



### Connecter le servomoteur à la carte

Il y a plusieurs façons de connecter un servomoteur à votre carte. Vous pouvez utiliser n'importe quelle *pin* de sortie analogique (broches PWM) pour connecter la *pin* de contrôle. Dans notre exemple, nous allons utiliser la *pin* **D2**. Le servomoteur sera connecté comme suit :

- **Noir** pour **GND**
- **Rouge** pour **V+ (3V3)**
- **Orange** pour **SIG (D2)**

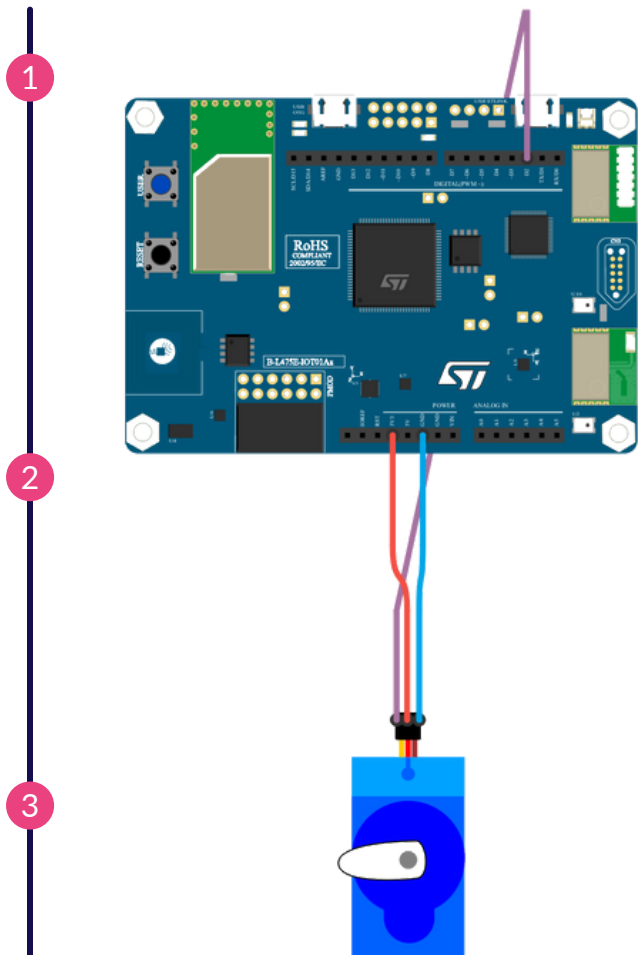
### Connecter la carte à l'ordinateur

Avec votre câble USB, connectez la carte à votre ordinateur en utilisant le **connecteur micro-USB ST-LINK** (sur le coin en haut à droite de la carte). Si tout se passe bien, vous devriez voir apparaître sur votre ordinateur un nouveau lecteur appelé **DIS\_L4IOT**. Ce lecteur est utilisé pour programmer la carte en copiant simplement un fichier binaire.

### Ouvrir MakeCode

Allez dans [l'éditeur MakeCode de Let's STEAM](https://makecode.lets-steam.eu). Sur la page d'accueil, créez un nouveau projet en cliquant sur le bouton "New Project". Donnez à votre projet un nom plus expressif que "Sans titre" et lancez votre éditeur.

*Ressource : [makecode.lets-steam.eu](https://makecode.lets-steam.eu)*



Connecter le servomoteur à la carte



## ÉTAPE 1 - CONSTRUIRE



Après avoir créé votre nouveau projet, vous accéderez à l'éditeur par défaut ci-contre.

### Programmer la carte

Dans l'éditeur JavaScript de MakeCode, copiez/collez le code disponible dans la section "**Programmer**" ci-dessous.

Avant d'essayer ce programme sur la carte, vous pouvez l'essayer directement dans le simulateur. Si vous changez les valeurs 0 et 180, vous verrez directement le résultat.

Si ce n'est pas déjà fait, pensez à donner un nom à votre projet et cliquez sur le bouton "**Download**". Copiez le fichier binaire sur le lecteur **DIS\_L4IOT** et attendez que la carte finisse de clignoter et votre servomoteur commencera à bouger !

### Exécuter, modifier, jouer

Votre programme s'exécutera automatiquement chaque fois que vous le sauvegarderez ou que vous réinitialiserez votre carte (appuyez sur le bouton intitulé RESET).

Si tout fonctionne bien, votre servomoteur va commencer à bouger.

Essayez de comprendre l'exemple et commencez à le modifier en changeant la période entre les deux mouvements.

4



Editeur par défaut

5



Editeur Javascript Makecode



Votre servomoteur se met à bouger



## ÉTAPE 2 - PROGRAMMER



```
// aller de 0 à 180 degrés
forever(function () {
  // dire au servomoteur d'aller à la position 180 degrés
  pins.D2.servoWrite(180)
  // attendre que le servomoteur atteigne la position
  pause(1000)
  // dire au servomoteur d'aller à la position 0 degré
  pins.D2.servoWrite(0)
  // attendre que le servomoteur atteigne la position
  pause(1000)
})
```

### Comment cela fonctionne-t-il ?

Cet échantillon est assez simple puisqu'il s'agit du classique clignotement adapté à un servomoteur.

L'instruction principale est **pins.D2.servoWrite(XXX)**. Cette instruction demande au servomoteur de tourner jusqu'à un angle de XXX degrés (comme défini par vos besoins spécifiques en fonction du projet que vous développez).

Pour passer d'une position à l'autre, le servomoteur prend un certain temps. Il faut donc toujours ajouter un délai avant de commencer un autre mouvement.

**Ce programme ne fait que balayer de gauche à droite sans s'arrêter !**



Par rapport à un moteur à courant continu ordinaire, un servomoteur ne tourne que dans une certaine plage d'angles, alors qu'un moteur à courant continu ordinaire (CC) tourne sur un cercle.

Un servomoteur ne peut pas tourner sur un cercle. Un moteur CC ordinaire ne peut pas nous donner un retour sur l'angle de rotation, mais un servomoteur peut le faire. Leurs utilisations sont donc différentes.

Les moteurs CC ordinaires s'utilisent comme source d'énergie rotative, tandis que les servomoteurs s'utilisent pour donner un certain angle à un objet qu'ils contrôlent, comme faire bouger une articulation de robot.



## ÉTAPE 3 - AMÉLIORER



Essayez de **réduire autant que possible la valeur de la pause** pour supprimer tout arrêt dans le mouvement.

Ajoutez des instructions pour **faire un arrêt court en position médiane**. Adaptez le délai de la pause pour être sûr que l'arrêt soit très court.

Transformez ce programme pour **réaliser une minuterie avec un servomoteur**. A chaque pas, déplacez le servo de 3 degrés. Adaptez le délai de sorte que chaque étape prenne approximativement 1s.

Démarrer le **mouvement de balayage** uniquement **lorsque le bouton USER a été poussé**.

1



2



3



4



## ALLER PLUS LOIN



**Servomoteur** - Apprenez-en plus sur le mécanisme et le fonctionnement du servomoteur.  
<https://en.wikipedia.org/wiki/Servomotor>



**Servomoteurs sur micro:bit** - Tout sur les boutons et leur utilisation dans MakeCode avec Shawn Hymel, influenceur en nouvelles technologies.  
<https://www.youtube.com/watch?v=okxooamdAP4&t=200s>, <https://shawnhymel.com>



**Bras robotique trieur de couleurs DIY** - Apprenez à fabriquer votre propre bras robotique de triage des couleurs à l'aide de capteurs ultrasoniques et IR.  
<https://thetempedia.com/project/diy-color-sorting-robotic-arm/>



### Explorer d'autres fiches d'activité

**R1AS14 - Créer un minuteur pour les œufs**

