

# CRÉER UN MINUTEUR POUR LES ŒUFS

#R1AS14



## Disponible sur



## Prérequis

- R1AS13 - Les servos font bouger les choses !

## Matériel

- 1 carte programmable "**STM32 IoT Node Board**"
- 1 câble USB Micro-B
- 1 mini-servo SG-90 (1,6 kg)
- Câbles de connexion
- 1 petite feuille de carton (20cm\*10cm)
- Des bâtons de bois robustes (moins de 10 cm)

## De quoi parle-t-on ?

Créons un objet simple mais utile, un minuteur à œufs ! Cette activité permettra d'appliquer les connaissances acquises sur les servomoteurs, sur un problème du quotidien.

## Durée

35 minutes

## Niveau de difficulté

Avancé

## Activité étendue



## OBJECTIFS D'APPRENTISSAGE

- Créer une minuterie
- Utiliser un servomoteur pour afficher des données
- Effectuer un processus d'étalonnage pour améliorer la précision du chronomètre





Dans cette activité, nous allons créer un objet simple mais utile, une minuterie, en utilisant la programmation et la pratique du bricolage ! Après l'avoir réalisé, vous serez un vrai grand chef ! Pour faire bouillir correctement un œuf, nous utilisons la **règle appelée 3,6,9** ! Cette règle donne la durée correcte de cuisson exacte en minutes d'un œuf en fonction de vos objectifs de cuisson :

- 3 minutes pour les **œufs à la coque**
- 6 minutes pour les **œufs mollets**
- 9 minutes pour les **œufs durs**



## ETAPE 1 - CONSTRUIRE



### Préparer le matériel électronique

Câblez correctement votre carte et votre servomoteur en utilisant la fiche d'activité #R1AS13 - Les servos font bouger les choses !

### Créer l'aiguille de l'horloge et la fixer

Prenez les bâtons de bois robustes et fixez-les au bras du servomoteur.

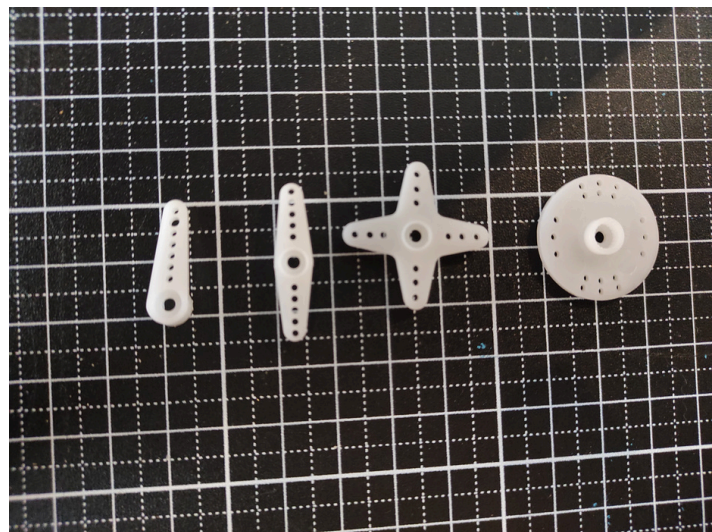
**i** Les bras de servomoteurs sont des accessoires qui s'adaptent sur l'arbre de sortie et vous permettent de relier mécaniquement la sortie du servomoteur au reste de votre mécanisme. Les servomoteurs sont généralement fournis avec un assortiment de bras de servomoteur.

Malheureusement, les bras exacts inclus ne sont généralement pas spécifiés et peuvent varier.

Et, comme les arbres de sortie des servomoteurs et leurs cannelures varient, les bras sont souvent incompatibles entre les marques et les modèles de servomoteurs.

La façon la plus simple de fixer l'aiguille de votre horloge est d'utiliser un élastique, mais vous pouvez aussi utiliser de la colle chaude ou du scotch.

1



2



Créez l'aiguille de l'horloge et fixez-la au bras du servomoteur.



## ETAPE 1 - CONSTRUIRE

### Créer le panneau avant du minuteur

Sur le carton, faites un petit trou de la taille de l'arbre de votre servomoteur. Le trou doit se trouver au milieu du côté le plus long de votre carton.

Mettez le servomoteur derrière et fixez l'aiguille d'horloge sur l'arbre du servomoteur.

Tourner le bras en position minimale (angle 0°) et fixer le servomoteur de façon à ce que l'aiguille de l'horloge soit horizontale. Avec un stylo, faites une petite marque pour indiquer 0 seconde. Tournez le bras en position maximum (angle 180°) et faites une petite marque pour indiquer 180 secondes.

### 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://letssteam.com/makecode). Sur la page d'accueil, créez un nouveau projet en cliquant sur le bouton "Nouveau projet". 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)

### Programmer votre carte

Dans l'éditeur JavaScript de MakeCode, copiez/collez le code disponible dans la section "**Programmer**" ci-dessous. Si ce n'est pas déjà fait, pensez à donner un nom à votre projet et cliquez sur le bouton "**Télécharger**". Copiez le fichier binaire sur le lecteur **DIS\_L4IOT** et attendez que la carte finisse de clignoter et votre servomoteur commencera à bouger ! Avant d'essayer ce programme sur la carte, vous pouvez l'essayer directement dans le simulateur. Si vous cliquez sur le bouton USER, vous verrez votre minuterie démarrer.

### 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 commencera à bouger.

3



Créer le panneau avant du minuteur

4

5

6

7



## ETAPE 2 - PROGRAMMER



```
input.buttonUser.onEvent(ButtonEvent.Click, function () {
  for (let pos = 0; pos <= 179; pos++) {
    pins.D3.servoWrite(pos)
    pause(1000)
  }
  for (let i = 0; i < 5; i++) {
    pins.D4.servoWrite(0)
    pause(1000)
    pins.D4.servoWrite(180)
    pause(1000)
  }
})
```

### Comment cela fonctionne-t-il ?

La partie principale du code concerne les interactions avec les boutons. Ces interactions sont faites avec la fonction **input.buttonUSER.onEvent**.

Lorsque vous cliquez sur le bouton **USER**, vous démarrez la minuterie en modifiant la position du servomoteur d'un degré par seconde.

Lorsque vous avez fini de compter de 179 à 0, vous commencez à déplacer rapidement votre servomoteur pour signaler la fin de la minuterie.



## ETAPE 3 - AMÉLIORER



En **ajoutant un servomoteur**, réalisez un second indicateur permettant de connaître l'état de la cuisson de votre œuf (*cru, à la coque, cuit, dur*).

Changez l'animation finale du minuteur en **ajoutant un buzzer** pour faire plus de bruit.

La version actuelle du programme n'est pas calibrée, votre minuteur vous donnera une valeur approximative. Si vous voulez être un cuiseur d'œufs plus scientifique, vous devez suivre un **processus de calibrage**. Pour calibrer un minuteur, utilisez une **horloge de référence**. Vous pouvez facilement utiliser l'horloge de votre smartphone par exemple pour mesurer la durée du minuteur. Pour réduire l'incertitude, vous répétez la mesure de nombreuses fois (dix fois suffisent par exemple) pour pouvoir **calculer la valeur moyenne et utiliser un produit en croix pour trouver la valeur correcte du délai**.

1



2



3

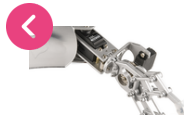


## ALLER PLUS LOIN



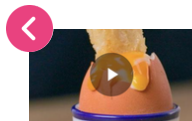
**Modulation de largeur d'impulsion** - En savoir plus sur le signal numérique de modulation de largeur d'impulsion.

<https://learn.sparkfun.com/tutorials/pulse-width-modulation/all>



**Comment faire bouillir un œuf à la perfection** - Apprenez combien de temps il faut faire bouillir un œuf pour obtenir la consistance parfaite.

<https://www.bbcgoodfood.com/howto/guide/how-boil-egg-perfectly>



**Compte à rebours** - Créer un compte à rebours et regardez les secondes défilier sur la montre micro:bit.

<https://makecode.microbit.org/projects/watch/timer>



**Micro:bit Egg Timer** - Réalisez un minuteur amusant pour garantir le temps de cuisson parfait des œufs en utilisant l'impression 3D et micro:bit.

<https://www.myminifactory.com/object/3d-print-micro-bit-egg-timer-18361>



## Explorer d'autres fiches d'activité

**R1AS15 - Collecter des données**

