

# Projet Arduino

# Réveil à Coucou

Filippone Adam

Baratli Wassim



# Sommaire

1. Objectifs – Cahier des charges
2. Vision globale du projet
3. Algorithme du projet
4. Plannings (initial et final)
5. Problèmes et résolutions
6. Développement de chacune des parties du projet.
7. Conclusion–perspectives
8. Bibliographie

# Objectifs – Cahier des charges

Notre projet consiste à fabriquer un coucou à l'aide d'Arduino. Ce dernier est fabriqué à l'aide du matériel électronique mis à disposition ainsi que de la Fabrique pour réaliser un nichoir en bois dans lequel réside l'oiseau. Notre coucou doit être capable de sortir de son nichoir et sonner à une heure précise que nous choisirons au préalable. Des portes sont supposées s'ouvrir pour laisser le coucou se déplacer et ruisseler. L'heure à laquelle le réveil sonne est programmée à l'aide d'un site web ou d'une application.

Objectifs :

- Respecter les délais imposés
- Inclure une connexion radiofréquence (Bluetooth, wifi)
- Réaliser un réveil
- Réaliser un mécanisme performant permettant de faire sortir et rentrer le coucou
- Activer et de régler le réveil à distance à l'aide d'un site web ou d'une application

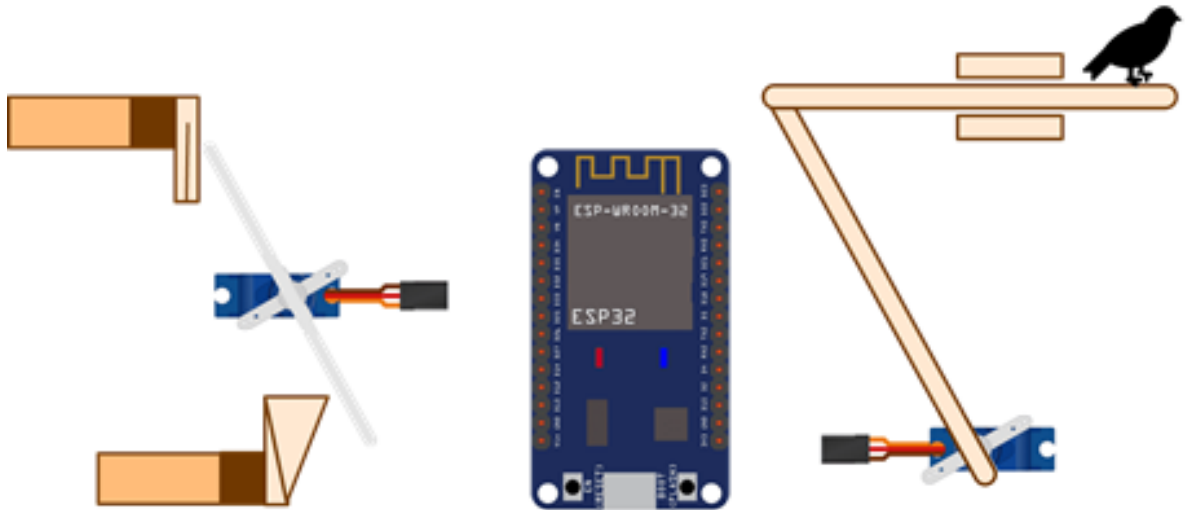
Limites

- Prendre en compte : le coût des composants, le temps de développement, la difficulté de certaines étapes et la taille du projet

Objectifs manquants et atteints :

Nous allons éclairer les différentes parties de notre projet qui fonctionne ainsi que les objectifs que nous n'avons pas réussi à atteindre. Nous avons réussi à réaliser un mécanisme complet pour imiter un véritable coucou. Il y a cependant certaines difficultés que nous n'avons pas réussi à atteindre.

# Vision globale du projet avec les différents modules



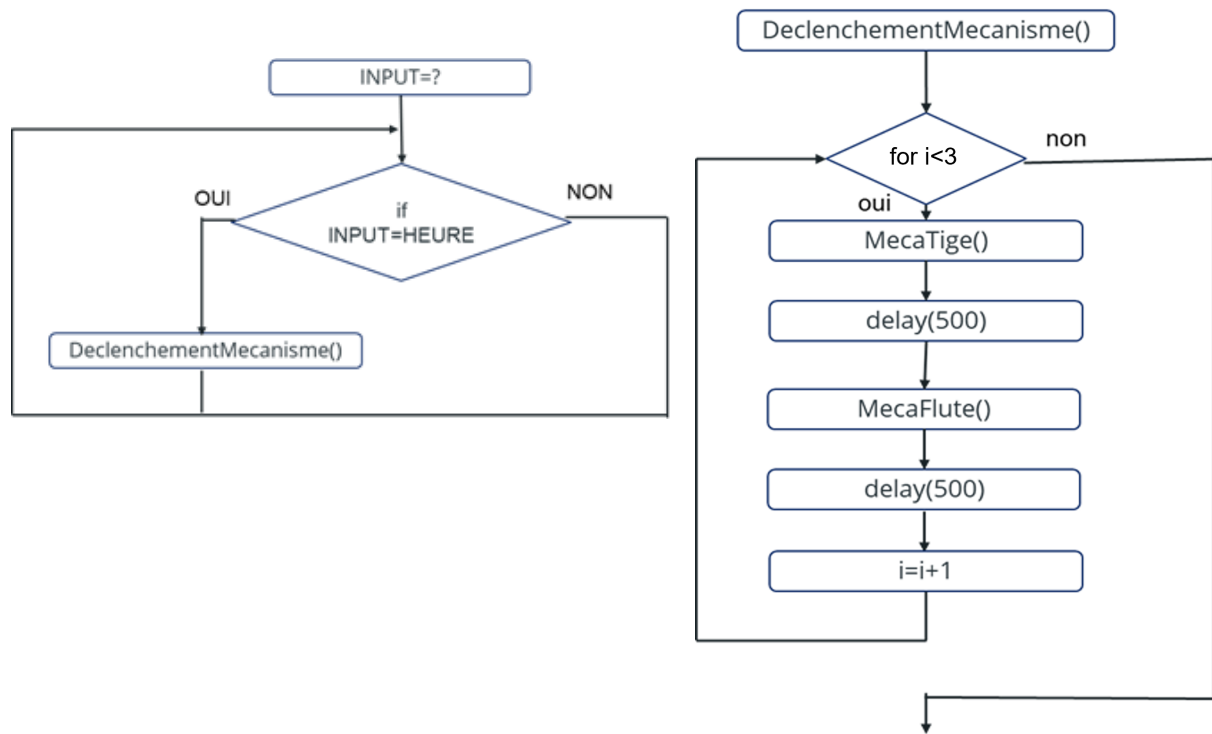
Au cours de notre projet, nous avons utilisé différents modules qui nous ont permis d'atteindre nos objectifs :

- Module ESP32
- Carte Arduino Uno
- Servo-moteurs
- Module RTC
- Ecran LCD

Quelques exemples de modèles :

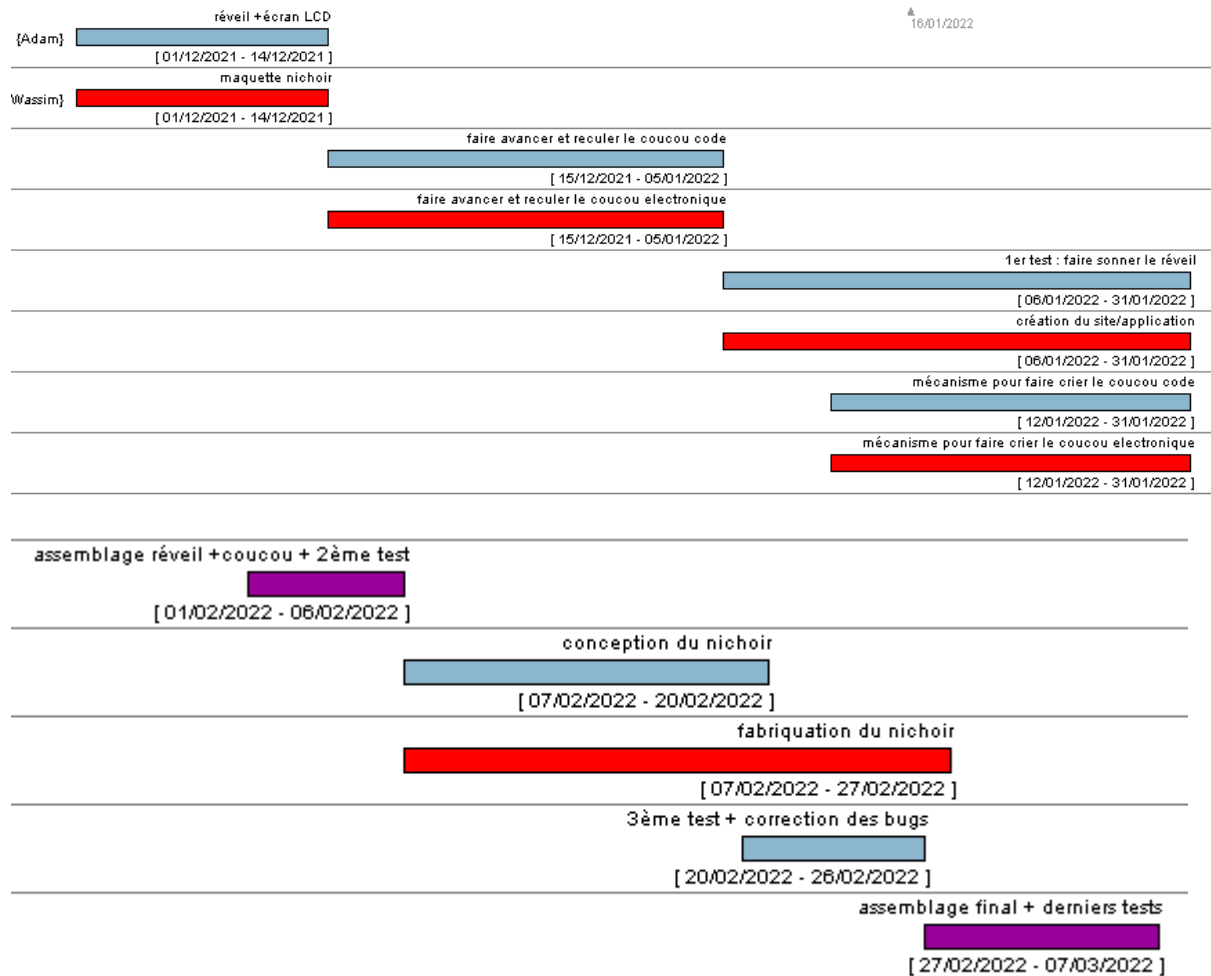


# Algorithme du projet

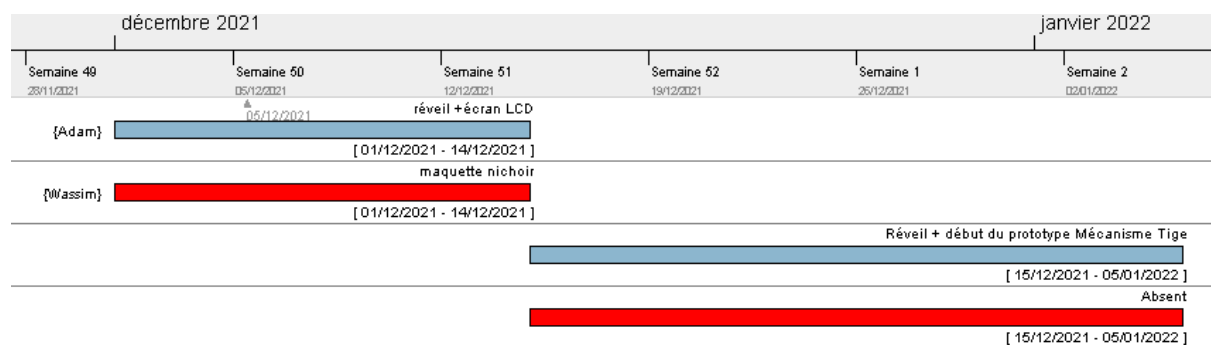


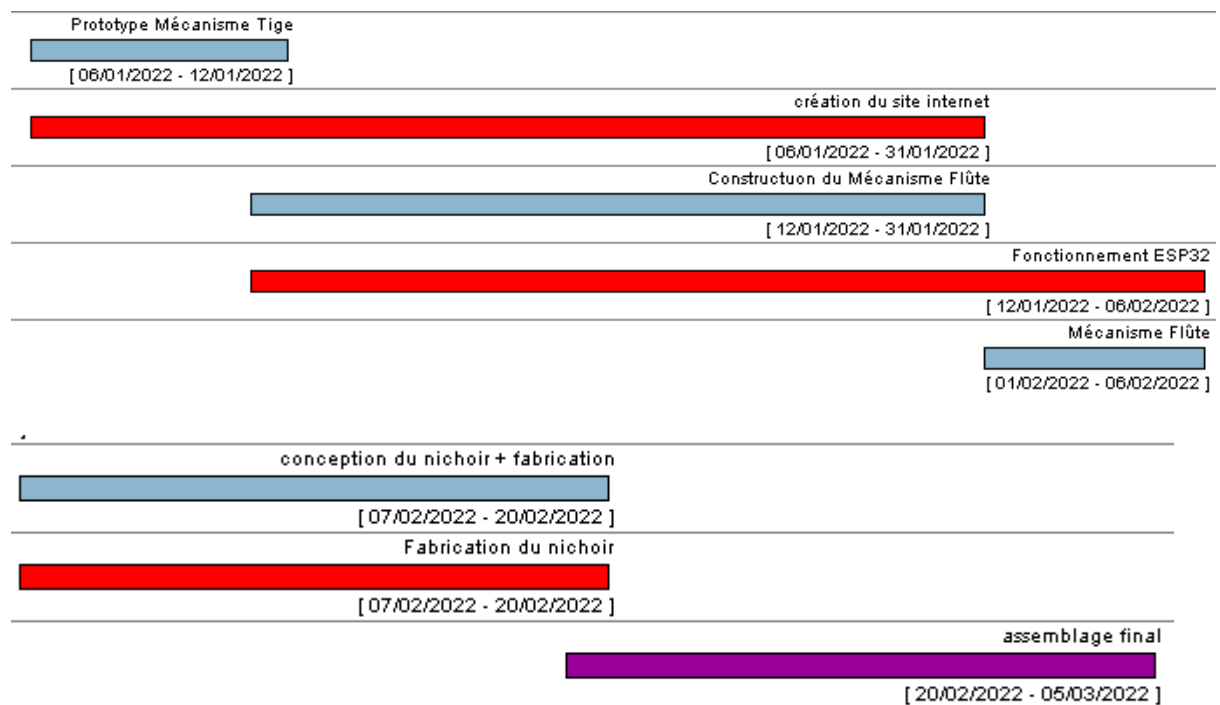
Ci-dessus se trouve l'algorithme de notre projet. Si l'heure récupérée sur le site est atteinte, on déclenche le mécanisme complet qui permet d'actionner le réveil. Deux fonctions règles respectivement le mécanisme des soufflets et le mécanisme de la tige.

# Planning Initial



# Planning Final





# Développement du projet

## Horloge

Dans un premier temps nous avons réalisé une horloge numérique grâce à un écran LCD, un module RTC et un potentiomètre pour régler la luminosité. Sur cette horloge était affichée l'heure, les minutes, les secondes mais également la date du jour.

## Nichoir

Nous avons au même moment conçu une maquette du nichoir en carton pour avoir une idée de la taille finale du projet. Cette maquette a évidemment été sujette à de nombreuses modifications car nous n'avions pas créé les mécanismes qui sont actuellement dans la maisonnette. Les dimensions du nichoir sont les suivantes. 29 cm par 29 cm pour chaque mur de 5mm d'épaisseur. Le tout est surplombé par un toit de 15 cm de hauteur. Sur la face arrière on a placé une trappe qui permet d'accéder à l'intérieur de la petite maison. Elle nous sert à accéder aux mécanismes pour les réparer si l'on remarque un dysfonctionnement. Un fil qui se serait décroché, ou une fixation décollée. Les portes et la trappe arrière ont été fixées à l'aide charnières

## Site Internet

Nous avons pensé à créer une interface pour pouvoir paramétrer le réveil. Nous avons pensé à un site Web ou une application. Finalement, notre choix s'est porté sur le site web, qu'on a créé entièrement en langage HTML/CSS et qui s'affiche juste ici. Donc le site affiche une maison en bois sur laquelle est affichée la date et l'heure en temps réel. On voulait vraiment créer un site web qui soit esthétiquement beau et qui soit reconnaissable. Il y a un petit formulaire qui permet à l'utilisateur de définir une heure bien précise, (heure et minutes) et c'est cette information qui sera conservée par l'Arduino.

Au départ on comptait utiliser un module Bluetooth pour assurer cette liaison site-web Arduino, puis notre choix s'est porté sur le module ESP32, il a fallu au préalable, nous familiariser avec cette carte, l'installer, prendre connaissance de ses librairies, cela n'a pas toujours été facile.

## Mécanismes

Le premier mécanisme à avoir été créé est celui qui permet de faire sortir le coucou. Nous avons collé un morceau de polystyrène sur le socle de la maison sur lequel est collé un servo-moteur. Ce servomoteur est relié à une tige en bois elle-même reliée



à une autre tige en bois. C'est cette dernière qui sort de la maisonnette. Elle est fixée sur un support en bois pour ne pas tomber. Puis à l'aide d'un programme on fait varier l'angle du servo, ce qui va permettre cet effet de va et vient. C'est une sorte de bras mécanique.

Une séance après, nous avons conçu le 2ème mécanisme qui permet d'imiter le ruissellement d'un oiseau.

Nous avons la même base que le mécanisme précédent. Un servo moteur encastré dans un bloc de polystyrène lui-même collé au sol. Sur le servo est fixée une tige en bois parallèle au sol sur laquelle sont insérés 2 fils métalliques. Ces 2 fils métalliques sont attachés (grâce à 2 crochets) à des flûtes (soufflets) spéciales coucou de tailles différentes. Encore une fois à l'aide d'un programme, on déclenche le mécanisme qui en fait une sorte de balance. Lorsqu'un soufflet s'ouvre un autre se ferme, cela produit un son et on répète ce schéma plusieurs fois.

En combinant les fonctions des 2 mécanismes dans la même boucle, on arrive à faire sortir et rentrer le coucou et entendre un son simultanément.

## Problèmes

Nous avons rencontré quelques problèmes que nous n'avons réussi à surmonter. Les portes ne se referment pas automatiquement lorsque le coucou rentre "chez lui". En effet, nous avons construit un mécanisme très simple à l'aide d'élastique pour ramener les portes en position fermée après la sortie de la tige. Le problème est que les portes se referment sur la tige en bois qui frottait dessus donc. Cela bloquait le mécanisme. Il manquait de temps pour trouver une solution.

Nous avons rencontré de nombreux problèmes lors de l'utilisation de l'ESP 32. Notamment dans l'utilisation des bibliothèques des servo-moteurs. Nous avons eu beaucoup de mal à assurer la connexion entre le site internet et la carte ESP32 Arduino. Nous avons dû faire face à beaucoup de bugs et d'erreurs de code.

Un problème classique que nous avons surmonté est celui des fils qui ne marchent pas. Certains montages réalisés ne fonctionnaient plus sans que l'on ne touche à rien. Il a fallu effectuer des tests avec des LED pour comprendre que ces fils étaient défectueux ou partiellement défectueux.

# Conclusion-perspectives

En conclusion, nous pouvons dire que nous avons réussi à atteindre la majorité des objectifs que l'on s'était fixés. Nous avons construit un vrai réveil, un coucou avec un mécanisme d'entrée-sortie et un mécanisme de sifflement, le tout grâce à Arduino. Les portes ne se referment pas automatiquement, il y a des problèmes quant à l'exécution du programme qui permet de récupérer l'heure du site.

Avec 9 semaines supplémentaires nous aurions fabriqué une réelle horloge (avec 2 aiguilles) à l'aide d'Arduino que nous aurions fixé sur notre coucou pour le rendre réaliste et esthétiquement plus beau. Nous aurions aussi pu ajouter d'autres atouts au réveil comme des leds ou même créer une application en parallèle du site internet..

## Bibliographie

<http://gilles.thebault.free.fr/spip.php?article53>  
<https://tutoduino.fr/blog-rtc/>  
<https://learn.adafruit.com/adafruit-ds3231-precision-rtc-breakout>  
<https://forum.arduino.cc/t/reveil/529500>  
<https://randomnerdtutorials.com/esp32-esp8266-input-data-html-form/>  
<https://www.tubefr.com/pendule-a-coucou-arduino.html>  
<https://arduino103.blogspot.com/2018/04/mon-reveil-1-tutoriel-dun-reveil.html>  
<https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/>  
<http://emery.claude.free.fr/esp32-serveur-web-simple.html>  
<https://gammatroniques.fr/fabriquer-un-reveil-avec-un-arduino-2-wifi/>  
<https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/>  
<http://emery.claude.free.fr/esp32-serveur-web-simple.html>  
<https://randomnerdtutorials.com/esp32-esp8266-input-data-html-form/>