

PROJETS ARDUINO – PEIP2

Année scolaire 2021-2022

DOMOTECH

Etudiants : Axel Delille et Athéna Trastour–Lozano

**Encadrants : Pascal Masson, Christian Peter et
Amina Benouakta**

Introduction

Le projet consiste en une maquette de maison domotique connectée. Le principe est simple, à l'aide de plusieurs capteurs et de conditions posées dans le code, la maison est capable de prendre des décisions et d'effectuer des actions simples comme allumer le chauffage, ou encore contrôler l'éclairage et ouvrir les fenêtres.

L'objectif du projet était de réussir à reproduire à bas coût des fonctionnalités de produits assez onéreux, car sur internet, il est possible de trouver des produits effectuant les mêmes choses tout en dépassant les 1500€. Un lien vers un exemple de produit est présent dans la bibliographie.

Initialement, le projet était défini comme une maquette dont quelques fonctionnalités de bases seraient entièrement automatisées comme l'éclairage, le chauffage, l'aération. Avec la possibilité au fur et à mesure de lui ajouter des fonctionnalités plus complexes telles qu'un interphone, un détecteur de mouvement voire peut-être une caméra de surveillance. Avec comme liaison radiofréquence, l'utilisation du Wifi pour gérer un site web qui permet de récapituler les informations du système tout en permettant à l'utilisateur d'interagir avec celui-ci et ce peu importe son emplacement.

Le planning initial quant à lui a été schématisé sur le diagramme de Gantt (Figure 1) ci-dessous :



Figure 1 - Diagramme de Gantt initial du Projet

Le projet est divisé en plusieurs points qui seront abordés tout au long de ce rapport, le premier étant une explication de l'algorithme général du code, le second un passage sur la liaison radiofréquence.

Chapitre I : L'algorithme du code du projet

I.1 Vue d'ensemble de l'algorithme

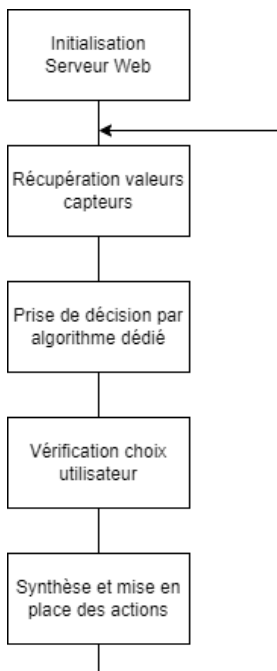


Figure 2 - Algorithme général du code du projet

Comme le montre le diagramme (Figure 2), au démarrage du projet, a lieu une initialisation de la connexion wifi avec un lancement du serveur et une instanciation de chaque méthode javascript permettant la communication entre code Arduino et site internet afin d'afficher des valeurs sur celui-ci ou de prendre la main sur l'algorithme et effectuer certaines actions.

À chaque instant a lieu une lecture des capteurs, et à l'aide des valeurs obtenues, un algorithme dédié permet de décider s'il faut effectuer ou non une action. Enfin, en faisant la synthèse des résultats de cet algorithme et des choix de l'utilisateur fait depuis le site internet, la maquette exécute certaines actions.

I.2 La partie prise de décision de l'algorithme

Lorsque la lecture d'informations des capteurs a été mise en place, le premier problème a été de savoir quand est ce que chaque action sera effectuée. Des conditions ont donc été posées sur l'algorithme. Cependant, l'utilisateur peut les outrepasser s'il contrôle la maison manuellement :

- Des conditions environnementales :
 - Il n'est pas envisageable d'ouvrir les fenêtres lorsque le chauffage est allumé, cela représenterait une perte d'énergie.
 - Il est préférable d'utiliser un éclairage naturel pour la maquette donc d'ouvrir les fenêtres plutôt que d'allumer la lumière.
- L'utilisation de valeurs seuils, qui vont par exemple permettre de dire à partir de quelle valeur de l'éclairage il est nécessaire d'ajouter un apport de lumière naturelle ou artificielle. Celles qui ont été utilisées sont les suivantes :
 - Un thermostat placé par défaut à 20°C mais modifiable depuis le site internet pour le chauffage.
 - La concentration en gaz inférieure à 1000 ppm (partie par million).
 - L'éclairage supérieur à 200 lux.

De ses conditions, a alors émergé une sorte de hiérarchie entre actions, et l’algorithme suivant (Figure 3) en est directement issu :

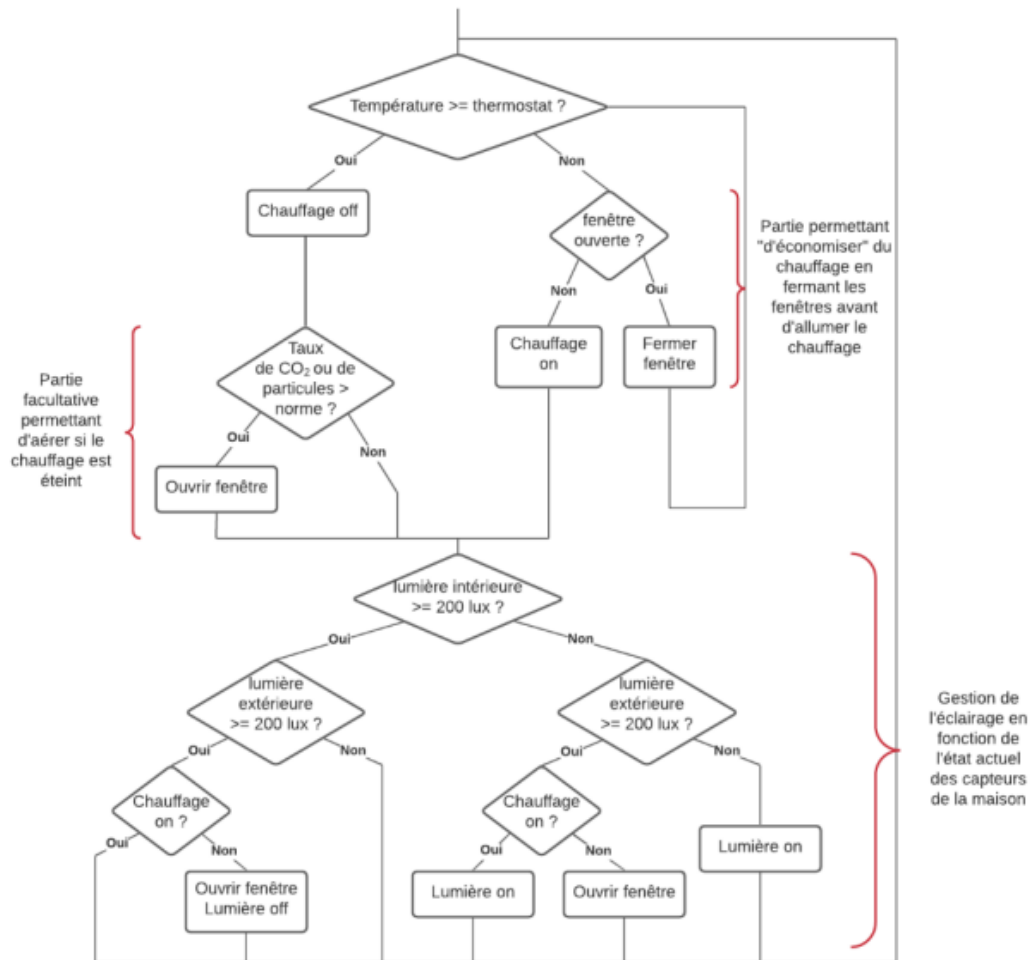


Figure 3 - Algorithme détaillé pour la partie prise de décision de l'algorithme du code du projet

Chapitre II : la liaison radiofréquence

II.1 Son fonctionnement général

Le principe de fonctionnement est simple, la maquette se connecte à un réseau Wifi au démarrage, qui, via son adresse IP, laisse accès au site internet qui contient à la fois les données des capteurs ainsi qu'une interface pour interagir directement avec le projet en ouvrant la fenêtre par exemple. Le site permet aussi d'agir de façon indirecte en modifiant la valeur du thermostat pour gérer l'allumage du chauffage. A l'échelle du projet d'Arduino, le site n'est accessible qu'en local et ne possède pas d'identifiant de connexion, et ne présente pourtant pas de problèmes de sécurité.

En effet, afin de se connecter au site web, il faut être sur le même réseau que l'ESP32 qui gère le serveur web car nous ne sommes pas sur un vrai routeur wifi mais bien sur un partage de connexion internet effectué par un smartphone. Par conséquent, il est impossible d'ouvrir un port afin de rendre le site public et donc disponible depuis n'importe quel réseau. Ainsi, se connecter au site nécessite de se connecter au réseau internet du possesseur de Domotech.

De nos jours, le cryptage du mot de passe de nos modems internet est effectué par l'un des services suivants : WEP, WPA, WPA2 qui sont extrêmement protégés. Donc, à échelle locale, une sécurité supplémentaire n'est pas requise. En revanche, si ce projet devait être installé dans une vraie maison, l'ouverture de l'un des ports de la box internet du client serait envisageable afin de rendre le site public et accessible depuis n'importe quel réseau n'importe où dans le monde. Dans ce cas précis, l'ajout de système d'identifiant et de mot de passe serait obligatoire pour garantir la sécurité de l'utilisateur.

Le serveur web est composé de 4 fichiers gérés par le gestionnaire de fichiers de l'ESP32 appelé Serial Peripheral Interface Flash File System (SPIFFS). Parmi ses 4 fichiers, il y a un CSS personnalisé de w3school qui permet de modifier l'affichage de la page html avec de simples noms de classes afin d'économiser du temps sur le design tout en ayant un rendu de bonne qualité. On retrouve également une librairie javascript ainsi qu'un fichier javascript réalisé pour le projet. Ses 2 fichiers ensembles permettent, en communiquant avec la page html, d'échanger des informations entre le code et le site web pour afficher des valeurs ou pour dire que l'utilisateur a appuyé sur un bouton. Et enfin, nous trouvons la page html en elle-même.

II.2 Les problèmes rencontrés

Premièrement, par manque d'informations, lorsque l'on a reçu l'ESP32 qui est le composant permettant de gérer le wifi de notre projet, nous avons pensé qu'il s'agissait d'un module à câbler sur l'Arduino, donc il a été mis de côté jusqu'à janvier. Malheureusement, il s'avère que, lors du passage à la radiofréquence, le projet a dû être repris du début pour s'adapter aux spécificités de cette carte plutôt capricieuse, qui, sans raisons apparentes, refusait d'exécuter les codes les plus simples malgré un câblage correct des composants. Mais après quelques vérifications, l'ESP32 que l'on utilisait alors avait un bus I2C défectueux ce qui perturbait les tests avec le capteur de lumière BH1750 qui est en I2C. Un changement d'ESP32 a alors été effectué.

Le problème suivant fut la lecture analogique. Il se trouve que la conversion analogique sur ESP32 est faite sur 12 bits donc 4096 valeurs au lieu de 10 bits sur un Arduino classique donc 1024 valeurs. Mais malgré cela, la lecture analogique de l'Arduino reste plus précise et plus fiable que celle d'un ESP32. Cette lecture analogique nécessite également une

tension d'entrée de 3.3 volts au lieu de 5 volts et c'est cette information qui fut plus compliquée à trouver. Pour remédier à cela, l'installation de ponts diviseurs de tension pour chaque entrée analogique fut effectuée avec des résistances de 1000 Ohms et 2000 Ohms.

Enfin, un problème qui fut résolu sans solution apparente est le suivant : malgré le pinout fourni par le constructeur, plusieurs pins n'agissent pas comme ils le devraient même sur l'ESP32 neuf que l'on a reçu après changement du premier. En effet, d'un pin à l'autre certains composants fonctionnent et d'autres non, sans aucune modification autre qu'un changement de pin sur le circuit et dans le code. Il fut donc nécessaire d'effectuer une batterie de tests afin de déterminer quel pin acceptait quel composant de notre projet. En effet, deux pins permettant par exemple d'être utilisés comme sorties ne vont pas réagir de la même façon selon le composant que l'on fera fonctionner avec, même si cela reste surprenant.

Le dernier souci que nous avons rencontré était l'utilisation de deux BH1750 simultanément qui était impossible sur Arduino et esp32 car, malgré le fait que les 2 capteurs avaient une adresse I2C différente, ils renvoyaient la même valeur. On a donc fini par en remplacer un par une photorésistance qui permet également de mesurer une intensité lumineuse.

Conclusion

Le projet actuellement est une maquette dont l'éclairage, le chauffage, et l'aération sont automatisés comme expliqué précédemment. Et l'on peut interagir avec cette maquette via un site internet afin de vérifier les valeurs des capteurs, ou de prendre la main sur l'algorithme.

Dans l'ensemble, chaque fonctionnalité sur laquelle nous avons travaillé a été implémentée. En revanche, le temps a été un facteur limitant à cause duquel certaines fonctionnalités n'ont pas vu le jour. L'aspect confort de la maquette domotique a été respecté mais aucun élément relatif à la sécurité de celle-ci n'a pu être ajouté. Il manque donc un système de surveillance, une caméra ou un digicode, et la partie sécurisation du site internet n'a pas été traitée, car, comme expliqué dans la partie dédiée à l'échelle du projet, elle n'était pas pertinente.

Si l'on devait poursuivre ce projet, il faudrait donc potentiellement commencer par s'occuper peut-être de la sécurisation de l'accès à la maison. Un système de digicode avec un mécanisme d'ouverture et de fermeture de la porte peut rapidement être mis en place. Pour le relier à la liaison radiofréquence, il serait possible de permettre à l'utilisateur de rentrer le code depuis le site internet également.

Il serait également judicieux de mettre en place un détecteur de mouvement qui, lorsqu'un mouvement est détecté et que l'utilisateur a indiqué sur le site internet ne pas être chez lui, permettrait au serveur web d'envoyer un mail à l'utilisateur ou tout autre forme de notification pour le prévenir. En poussant à l'extrême, il serait possible via une caméra, d'envoyer une photo prise dès que le capteur est déclenché pour permettre l'identification d'un intrus, et après cela, à l'aide d'un buzzer, de lancer un enregistrement audio prédéfini afin de dissuader le cambrioleur.

Bien entendu, remplacer l'utilisation d'un partage de connexion internet par un vrai routeur internet permettrait, en ouvrant le port requis, de rendre le site internet public et il faudrait alors s'occuper de créer un système de connexion au site internet avec un niveau de complexité de mot de passe minimum pour ne pas permettre à n'importe qui de s'y connecter.

Pour ajouter plus de fonctionnalités à la maison, il faudrait davantage de pins disponibles ce qui n'est plus le cas sur l'esp32. Il faudrait donc utiliser une carte Arduino et mettre en place une communication entre ESP32 et Arduino pour faire communiquer davantage de capteurs.

Enfin, il serait nécessaire de modifier la source d'alimentation de la maquette pour lui permettre de fonctionner sans être branchée à un ordinateur.

Avec du recul, il aurait été préférable de commencer par se renseigner sur l'intégralité du matériel mis à notre disposition, cela nous aurait permis de commencer directement par l'ESP32 et, par conséquent, d'éviter toute la partie où l'on est reparti de zéro, ce qui aurait contribué au développement d'autres fonctionnalités.

Dans l'introduction de ce rapport, le diagramme de Gantt initial a été fourni (Figure 1), voici le diagramme de Gantt représentant le réel travail effectué tout au long de ce projet (Figure 4) :

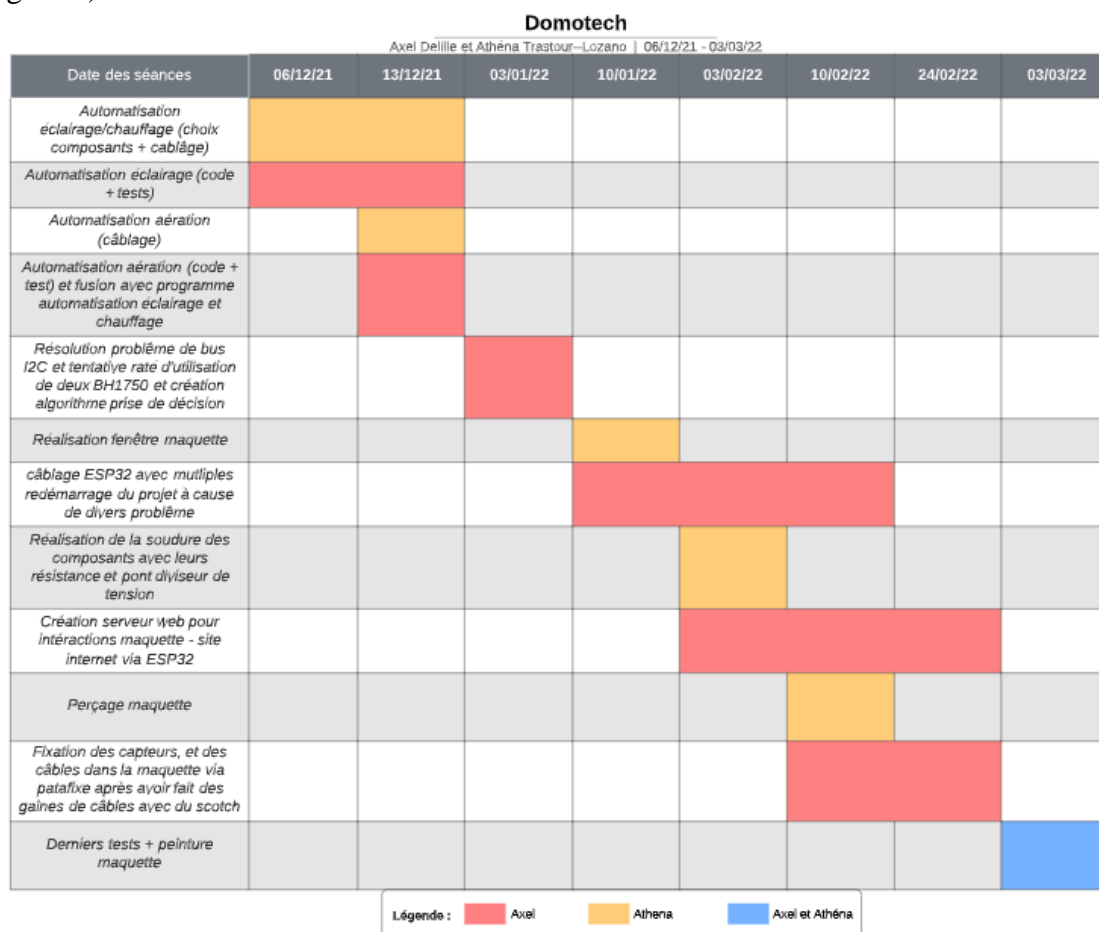


Figure 4 - Diagramme de Gantt final du projet

Enfin, nous tenons à remercier les différents enseignants qui nous ont encadrés tout au long de ce projet et qui ont été un soutien indispensable au bon déroulement de celui-ci.

BIBLIOGRAPHIE

- Exemple de produit domotique : <https://www.manomano.fr/p/kit-domotique-z-wave---pilote-volets-roulants--radiateurs--clairage--prise---fibaro-36984111>
- Informations relatives à l'esp32 : <https://www.upesy.fr/blogs/tutorials/list-tutorials-for-esp32-on-arduino-code-and-micropython>
- Information BH1750 : <https://github.com/claws/BH1750>
- Information DHT22 : <https://passionelectronique.fr/tutorial-dht22/>
- Information MQ2 : <http://noveselectric.com/detection-de-fumee-a-laide-du-capteur-de-gaz-mq-2/>
- Librairie pour avoir esp32 et servomoteur en même temps car la librairie initiale Servo est incompatible avec esp32 : <https://github.com/madhephaestus/ESP32Servo>
- Utilisation photorésistance comme capteur de luminosité : <https://www.aranacorp.com/fr/mesure-de-luminosite-avec-une-photoresistance/>
- Lien documentation du CSS de w3school utilisé dans la page html : <https://www.w3schools.com/w3css/default.asp>