

Cahier des charges

Un réveil intelligent

Antoine AUGUSTI
Etienne BATISE
Jean Claude BERNARD
Thibaud DAUCE

1 Électronique

- Faire un circuit imprimé pour remplacer le shield actuel
- Implémenter un système d'ampoule
- Implémenter une alimentation autonome
- Effectuer tous les branchements

Le shield existant permet, grâce à une photorésistance, de déterminer la luminosité dans la pièce où se situe le réveil. Il est fonctionnel mais nous devons créer notre propre carte pour répondre aux besoins du projet. Cette carte aura les mêmes objectifs que le shield, c'est-à-dire retourner une valeur analogique en fonction de la luminosité de la pièce. L'ancien shield possède une LED, contrôlée par la Arduino, permettant de savoir quand la luminosité dépasse un certain seuil. Pour notre projet, nous comptons intégrer le réveil dans une boîte et nous pensons réutiliser la LED afin de prévenir l'utilisateur que la batterie du réveil est faible.

Nous devons choisir l'ampoule et la connecter à notre circuit imprimé afin de pouvoir contrôler la quantité de courant à envoyer et ainsi proposer un éclairage progressif.

Le dernier objectif de cette partie d'électronique est d'effectuer des montages et par extension des câblages propres afin de produire un travail de qualité et de faciliter l'intégration des différentes cartes dans un boîtier.

2 Arduino

- Implémentation de la gestion de l'ampoule
- Réutiliser la LED pour signaler que la batterie est faible

Au niveau de la Arduino, le code présent permet déjà de déterminer lorsque le seuil de luminosité est dépassé. Il nous faudra donc ajouter l'allumage progressif de l'ampoule. Nous devons aussi résoudre les conflits entre l'allumage de l'ampoule et le capteur de luminosité afin de ne pas précipiter l'arrêt du réveil. Nous pourrions pour cela adapter le code déjà produit par l'équipe précédente et implémenter, par exemple, une variation du seuil de luminosité en fonction du stade d'allumage de l'ampoule.

La gestion de la batterie est importante et nous devons trouver un moyen de déterminer quand le niveau de batterie du réveil est faible pour pouvoir prévenir l'utilisateur (afin de ne pas se retrouver sans réveil du jour au lendemain). À ce stade de la conception, nous ne savons toujours pas si cette information pourra être récupérée directement via la Arduino ou si (plus probablement) nous devrions récupérer certaines informations via notre circuit imprimé. Cette batterie va nous permettre d'alimenter le Raspberry Pi et la Arduino sans câble, si nous voulons un boîtier autonome nous avons aussi besoin de supprimer le câble RJ-45 permettant l'accès au réseau. Nous devons donc rajouter un dongle Wi-Fi qui peut consommer très rapidement beaucoup de courant. Il faut donc aussi implémenter une gestion du Wi-Fi : bouton on / off ou allumage intelligent (un peu avant le déclenchement du réveil par exemple).

3 Raspberry Pi

- Affichage des prochains métros
- Gestionnaire de sons
- Implémentation de l'affichage LCD
- Paquet magique sur le réseau (broadcast)

L'affichage des prochains métro semble ne pas fonctionner actuellement sur le projet, sans doute à cause d'un changement sur le site de la TCAR. Il nous faudra réparer cette fonctionnalité.

Au niveau des améliorations, il n'existe pas de gestionnaire de son sur l'interface web permettant l'administration du réveil et donc, le changement de la sonnerie du réveil est laborieux. Nous comptons implémenter une nouvelle page web permettant d'ajouter des nouveaux sons et de choisir celui à jouer au prochain réveil.

Les deux dernières fonctionnalités sont optionnelles, même si nous aimerions les implémenter, il est peu probable que nous ayons le temps vu les délais à respecter pour le projet. La première consiste à gérer l'affichage de l'heure via un écran LCD et la deuxième, un peu plus technique mais très utile, permettrait d'envoyer un paquet "magique" TCP / IP sur le réseau local de l'utilisateur lors de la sonnerie du réveil (et pourquoi pas lors de son extinction) afin de donner la possibilité de réutiliser ce projet. En effet, un paquet "magique" est un paquet qui peut être lu par tous les appareils connectés au réseau, cela donne donc la possibilité de créer un autre projet d'électronique (par exemple un autre Raspberry Pi) analysant le réseau et exécutant une action lors de la détection de ce paquet (par exemple, allumage de l'ordinateur, de la cafetière ou autre...).

4 Bricolage

- Construire une boîte

Il ne faut pas oublier de prendre le temps de construire une boîte afin de protéger les circuits et rendre le projet plus esthétique.