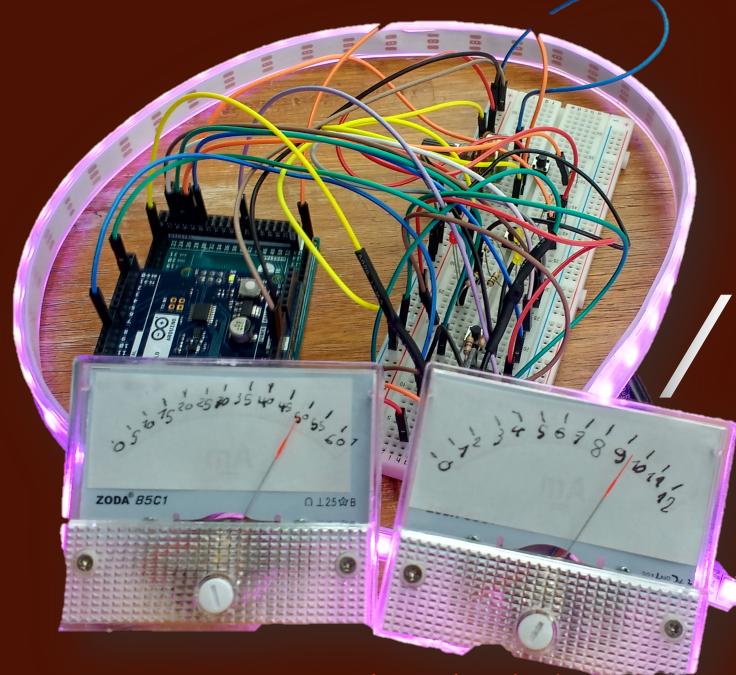


Théophane Durand

Éléonore de Rougé

ArduiWake



Projet de spécialité
Informatique et Sciences du numérique,
Année 2015-2016

Pourquoi faire un réveil ?

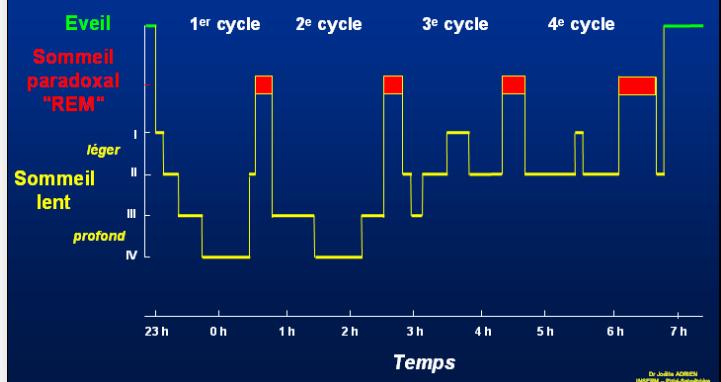
I existe aujourd’hui une multitude de réveils sur le marché. Des réveils sonores, odorants, lumineux... Cependant, parmi cette multitude, le choix se fait difficile. Comment trouver LE réveil qui nous corresponde ? Nous avons donc voulu créer un nouveau réveil à notre image.



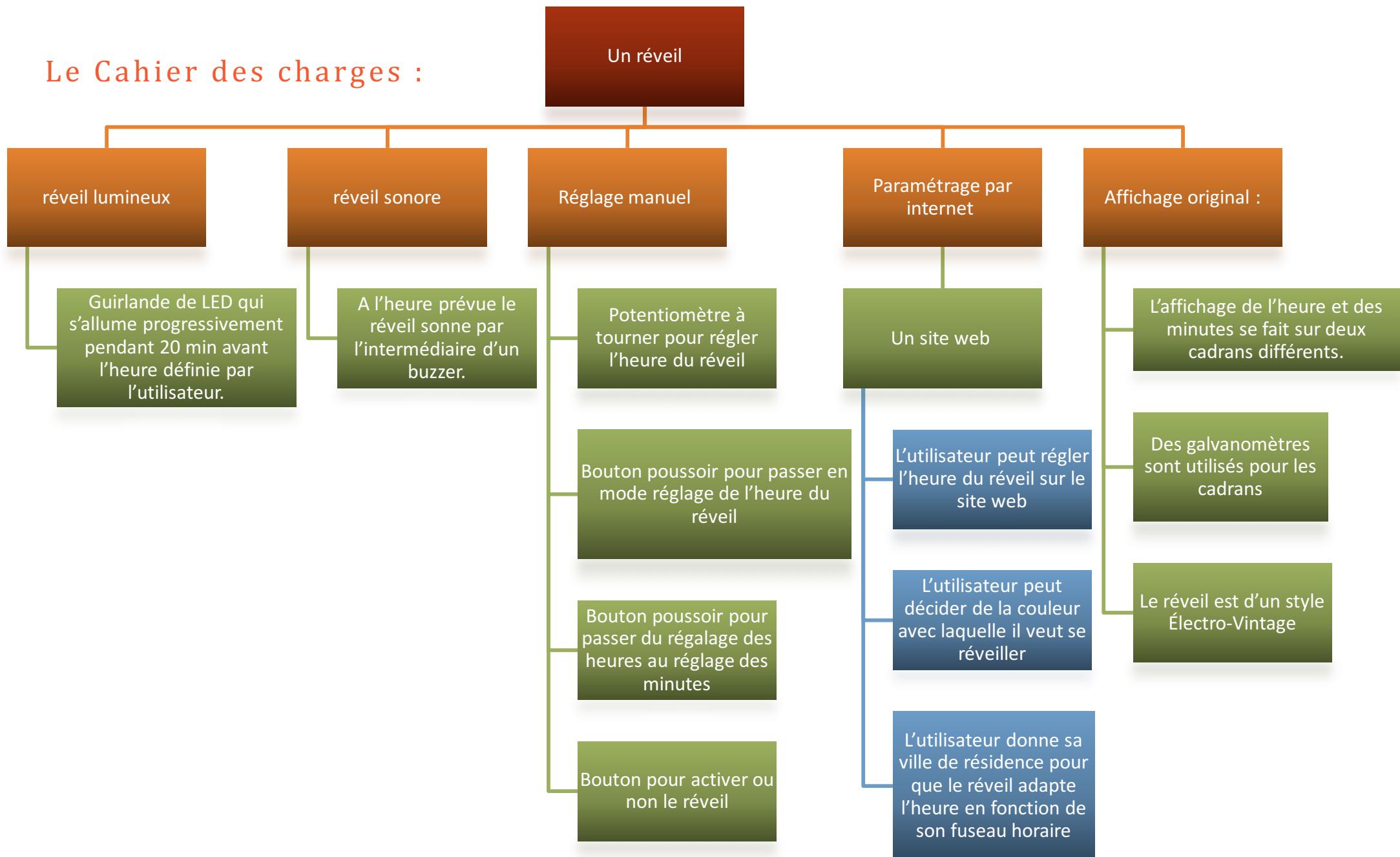
Notre réveil :

Nous voulions donc faire un réveil adapté à nos besoins. Nous voulions tout d’abord un design original. De plus, il est bien connu que la vie d’un lycéen est fatigante, il a besoin d’être opérationnel dès la première heure, il nous fallait donc un réveil doux. Nous sommes donc partis sur l’idée d’un réveil lumineux. Le nôtre s’allume progressivement durant les 20 minutes précédant l’heure définie. Grâce à ce système, la lumière ne réveille l’utilisateur que lorsqu’il est dans une phase de sommeil léger ou paradoxal, or plus le sommeil est léger moins le réveil est difficile.

Déroulement d'une nuit de sommeil (hypnogramme)



Le Cahier des charges :



Les objectifs :

Nous voulions faire de l'informatique sans avoir besoin d'un écran dans la production finale. C'est pourquoi nous utilisons des galvanomètres qui, en plus de donner un style électro-vintage, permettent d'afficher de manière simplifiée et originale des informations, ici l'heure et les minutes.

L'environnement :

Avec un tel cahier des charges, la solution la plus réalisable était un environnement Arduino pour sa puissance algorithmique et sa simplicité d'utilisation. Le langage Arduino est basé sur le langage C++.

L'idée de ce projet nous est venue grâce au magazine HACKABLE n°10¹ dans lequel, il était expliqué comment créer une horloge simple avec des galvanomètres.



¹ HACKABLE n°10 : <http://www.hackable.fr/?p=746>

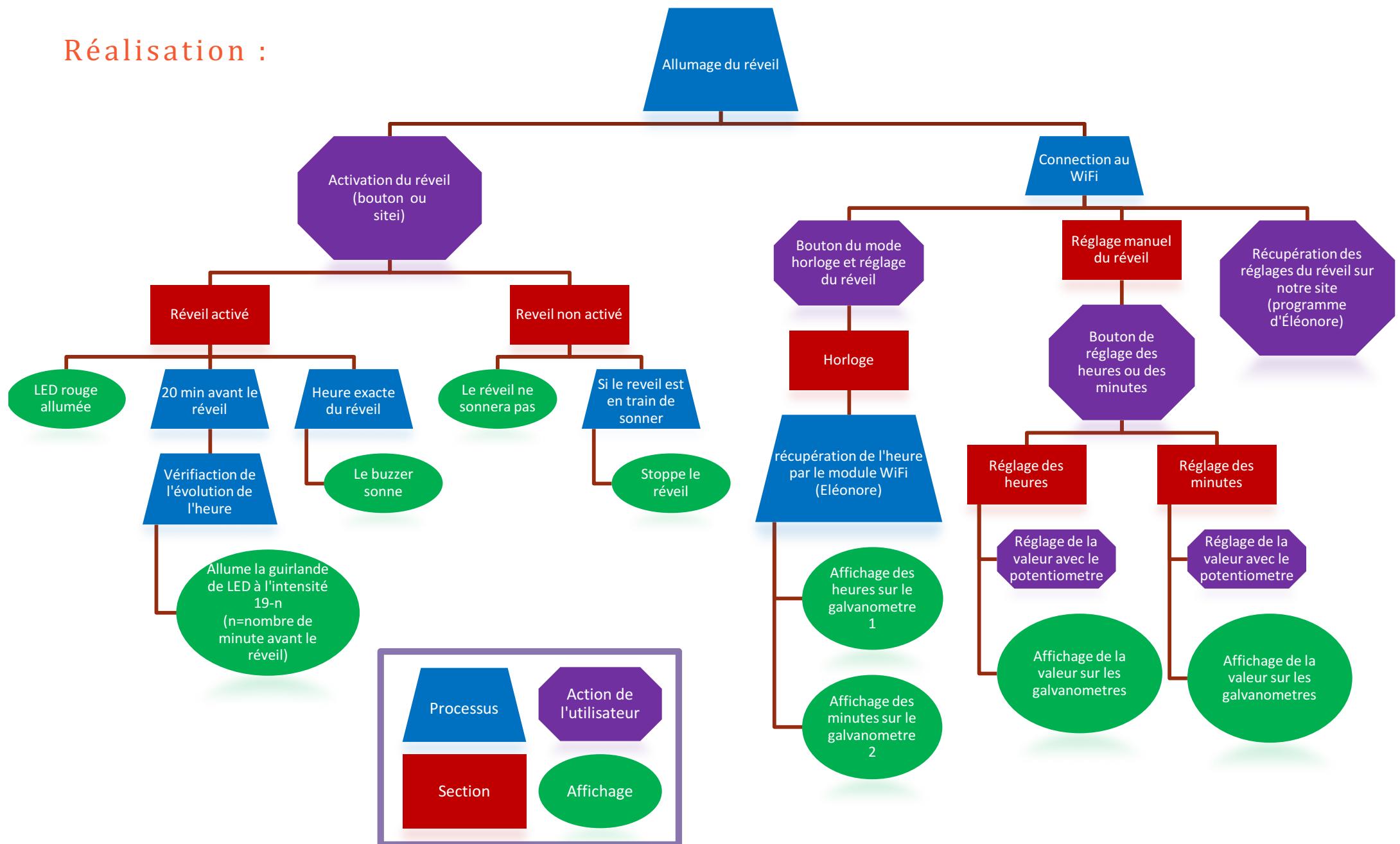
Répartition des tâches :

Tâches	Planning	Nom
Phase de recherche	Début Janvier Début Février	Théophane Durand Eléonore de Rougé
Familiarisation, découverte de l'outil Arduino.	Début Février	Théophane Durand Eléonore de Rougé
Récupération de l'heure du réveil / Début du code	4 Février 2016 22 Février	Théophane Durand
Recherches pour commander le réveil via Twitter ECHEC due à un changement de version Twitter / Recherches pour commander le réveil via Internet	4 Février 2016 22 Février	Eléonore de Rougé
Début cahier des charges / Commandes des composants	Fin Février début Mars	Théophane Durand Eléonore de Rougé
Création d'un site web pour ArduiWake	Début Mars	Eléonore de Rougé
Déclenchement du réveil : buzzer + LEDs	Début Mars	Théophane Durand
Contrôle de l'ArduiWake par le site web	Avril	Eléonore de Rougé
Réalisation du programme Horloge et Réveil	Avril	Théophane Durand
Liaison entre les deux programmes + entre les composants matériels	Fin Avril Début Mai	Théophane Durand Eléonore de Rougé
Problème d'incompatibilité du shield WiFi et du shield Horloge donc restructuration du programme	Fin Avril Début Mai	Théophane Durand Eléonore de Rougé

Fonctionnement de l'équipe :

E Léonore et moi-même avions la chance d'avoir à notre disposition deux cartes Arduino. Nous avons donc pu travailler chacun chez nous. Je travaillais avec l'Arduino Mega et les composants électroniques, tandis qu'Eléonore travaillait avec une carte Arduino Uno et le module WiFi. Ce n'est que fin Avril/début Mai que nous avons réuni nos deux programmes et donc posé le WiFi shield sur l'Arduino Mega, ce qui nous a permis de n'avoir plus qu'un seul outil de travail.

Réalisation :

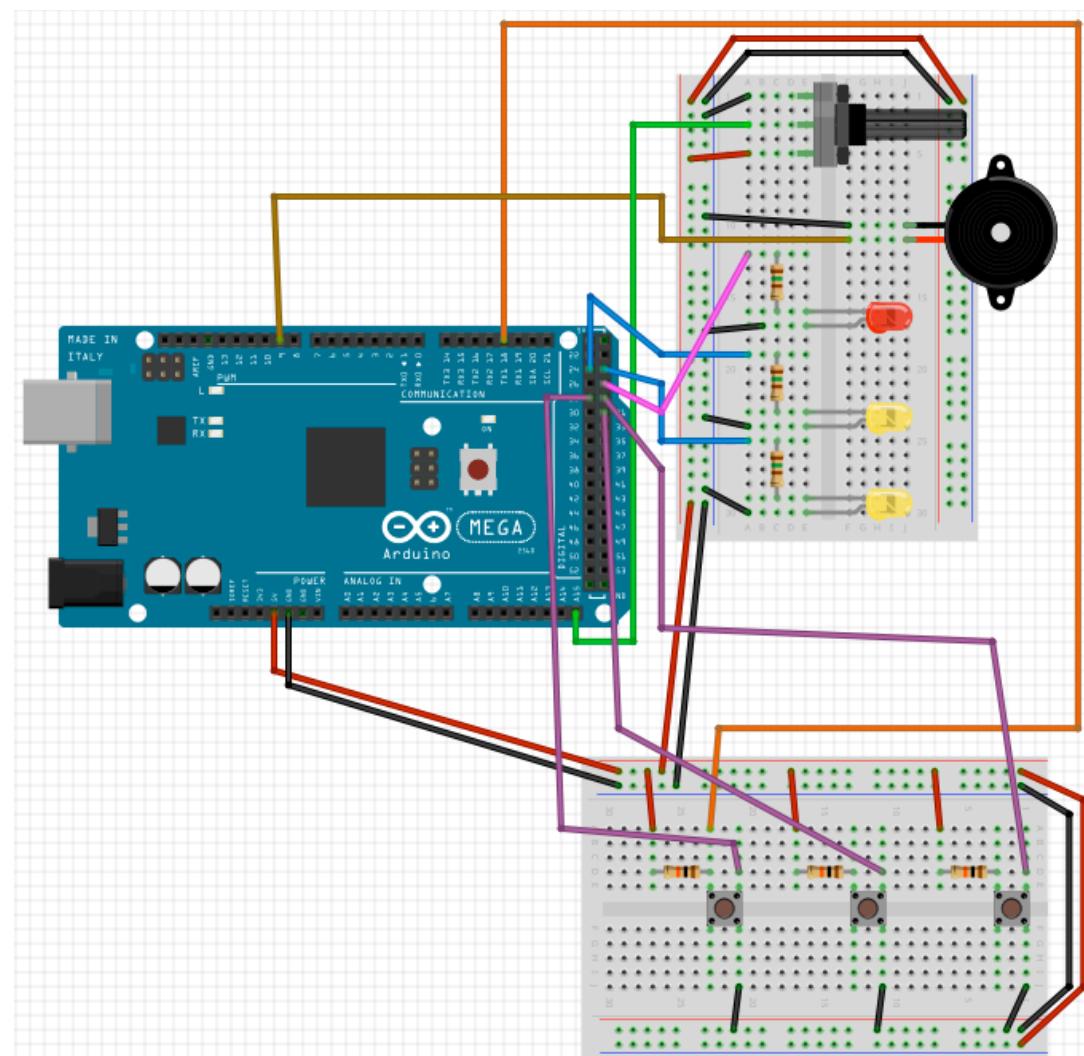


Introduction :

Mon rôle dans l'équipe a été de faire toute la partie électronique. J'avais déjà pratiqué l'Arduino et le langage C++ l'an dernier pour mon TPE. L'environnement m'était donc familier. Éléonore n'ayant jamais fait d'électronique ni d'Arduino, nous avons convenu ensemble qu'elle s'occuperait du site et de la récupération des données, moi de l'électronique ainsi que du paramétrage manuel du réveil.

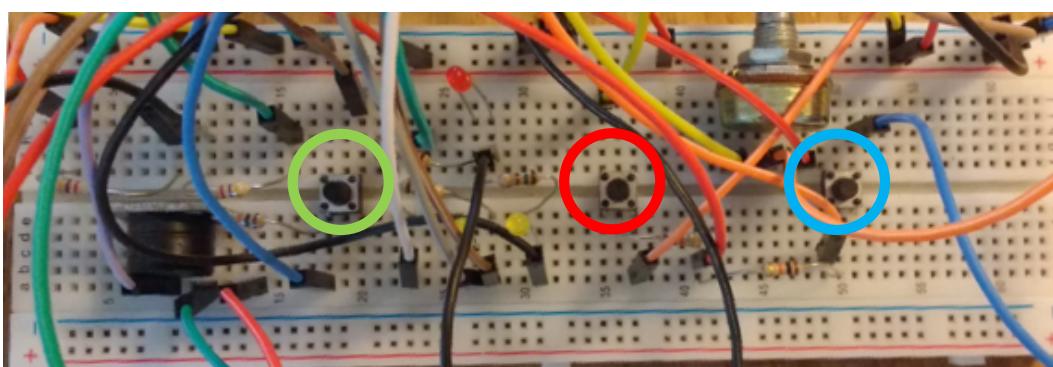
I/L'électronique :

La partie électronique du réveil est constituée de trois boutons poussoirs, deux LEDs pour afficher l'après-midi ou le matin; une pour savoir si le réveil est activé, un potentiomètre pour le réglage des heures et des minutes, un buzzer pour réveiller, une guirlande de LEDs (non représentée) pour réveiller, ainsi que deux galvanomètres (non représentés) pour l'affichage des heures et des minutes. J'ai tout branché sur une plaque de prototypage pour éviter les soudures. Tous les composants d'entrée et de sortie sont branchés sur les pins de l'Arduino. On retrouve les valeurs des pins dans le programme.



II/Quelques fonctionnalités de mon code :

Comme je le disais dans l'introduction, j'étais déjà familier du langage Arduino (qui n'est qu'une dérivée du langage C++), je trouve cet outil fantastique, car il n'est pas très compliqué mais très puissant ! On m'a offert un magazine HACKABLE dans lequel j'ai vu un article sur une horloge électro-vintage. Il était expliqué comment faire une horloge avec des galvanomètres. J'ai donc proposé à Éléonore que l'on prenne cet article comme base de notre projet. J'ai récupéré une partie du code pour le réveil. Étudions quelques parties de mon programme.



A) Le mode affichage de l'heure² :

Au départ, le code vérifie l'état d'une variable pour savoir si l'on est en mode affichage de l'heure ou réglage de l'heure du réveil. Ce paramètre est modifiable en appuyant sur le bouton (rouge) le programme est fait de telle sorte qu'une fonction vérifie si le bouton a été appuyé. Le cas échéant, elle change la valeur de la variable qui a appelé cette fonction³. A l'état initial, le réveil est en mode affichage de l'heure. Dans ce mode, la carte Arduino récupère la valeur de l'heure et des minutes et les affiche sur les galvanomètres. Ceux-ci peuvent recevoir une valeur entre 0 et 1023. Par un produit en croix, on affiche la juste valeur de l'heure (à laquelle on ajoute le décalage horaire si le réveil est réglé par le WiFi) et des minutes. Pour l'heure, par souci de lisibilité sur le galvanomètre, on n'affiche que 12 valeurs donc si l'heure est supérieure à 11, on revient à 0 sur le galvanomètre et on allume une LED pour indiquer que nous sommes l'après-midi.

² Annexe Figure 3

³ Annexe Figure 4

B) Le mode réglage de l'heure du réveil⁴:

Si la variable dont je parlais précédemment est en mode réglage de l'heure du réveil, on entre dans une autre fonction. Une autre variable entre en jeu, une variable pour vérifier si on souhaite modifier les heures ou les minutes. Cette variable est modifiable en appuyant sur un autre bouton (vert) (la même fonction que précédemment indique si le bouton a été appuyé ou non⁵). Que ce soit pour modifier les heures ou les minutes, le processus est semblable. On lit la valeur donnée par le potentiomètre (le potentiomètre donne une valeur comprise entre 0 et 1023 par un produit en croix on modifie cette valeur pour avoir une donnée entre 0 et 23 pour les heures et entre 0 et 59 pour les minutes), on stocke cette valeur dans une variable que l'on affiche sur les galvanomètres comme précédemment. A chaque fin de boucle, on vérifie si l'utilisateur a appuyé sur le bouton pour passer du réglage des heures au réglage des minutes, ou sur le bouton pour repasser en mode affichage de l'heure, si c'est le cas, la boucle s'arrête pour aller à l'endroit voulu. À la fin du réglage de l'heure du réveil, toutes les données sont envoyées au site web.

⁴ Annexe Figure 5

⁵ Annexe Figure 4

C) Les difficultés :

Un problème majeur a été au niveau du WiFi shield. Ce dernier n'était pas compatible avec le module horloge que nous avions prévu à la base, pour donner l'heure à notre réveil (cela aurait permis d'utiliser le réveil même sans connexion internet). Cependant, cette incompatibilité nous a obligés à utiliser le WiFi pour récupérer l'heure. Ce problème a été compliqué à régler car j'ai essayé une multitude de structures différentes du code mais aucune ne fonctionnaient. Les professeurs ne connaissant pas le langage Arduino, il était difficile pour eux de nous venir en aide quand nous avions un problème. Un des problèmes récurrents, est un problème de syntaxe du code. Par exemple, j'oubliais les deux « = » pour tester une condition, j'ai passé plus de 8 heures à essayer de trouver la solution.



IV/ Bilan :

L'ArduiWake a été un projet complet. J'avais déjà fait de la programmation et de l'électronique, mais ce projet m'a appris à élaborer un cahier des charges, à prendre conscience de ce qui était réalisable à notre niveau et à respecter des délais. J'ai eu beaucoup de chance de travailler avec Éléonore qui a un très bon sens de l'organisation et de la gestion. Ce projet nous a permis d'apprendre à travailler en équipe, il fallait toujours prévenir l'autre lorsqu'on faisait un changement et prendre les décisions ensemble pour ne pas abîmer le programme de l'autre. Cela nous a également fait prendre conscience de l'efficacité d'un partage équitable des charges de travail lorsqu'on est sur un projet important.



V/ Perspectives :

L'avantage de l'ArduiWake est qu'il est conçu pour s'adapter aux besoins personnalisés de l'utilisateur. Étant un réveil connecté à internet, il est facile de lui intégrer de nouvelles fonctions suivant la tendance du moment. Plus concrètement nous pensions, en plus d'être une horloge et un réveil lumineux, lui faire remplir le rôle de lampe. C'est à dire ajouter une fonction « allumage immédiat des LEDs de l'ArduiWake » qui se matérialiserait par un bouton supplémentaire sur le site web. En amélioration plus évidente, notre réveil étant tourné vers la personnalisation, il serait intéressant de pourvoir modifier la mélodie du réveil, en téléchargeant des musiques depuis le site internet. Nous n'avons pas eu le temps de nous attarder sur le design mais dans la même optique de personnalisation, si le réveil était industrialisé, nous pourrions créer plusieurs coques de différentes couleurs et personnalisables. Au niveau technique, il serait pratique de lui ajouter la fonction « snooze » qui permet de refaire sonner le réveil quelques minutes après la première sonnerie.

Annexes :

Pour accéder au code complet : <http://urlz.fr/3vlk>

(Figure 1)

```

35 //*****
36 #define HEURES 8 // galvanometre des heures
37 #define MINUTES 9 // galvanometre des minutes
38 #define AM 24 //Led d'indication du matin
39 #define PM 25 //Led d'indication de l'apres midi
40 #define LEDREVEILACTIF 26 //Led d'indication de l'activattion du réveil
41 #define BTNHEUREMIN 28 // bouton de sélection des heures ou des minutes
42 #define BTNHORLOGEREVEIL 29 //bouton de sélection du mode du réveil (réglage ou affichage)
43 #define BTNREVEILACTIF 31 //bouton d'activation du réveil
44 #define NEOPIXEL 52 //guirlande de led
45 #define POTENTIOMETRE A15 //potentiometre pour le parametrage
46 #define BUZZER 42 //buzzer
47 //*****

```

(Figure 3)

```

//si l'heure est entre 0 et 11
if (heure < 12 ){
  //on utilise directement la valeur
  analogWrite(HEURES, 255/12*(heure)+decalageHoraire);
  digitalWrite(AM, HIGH);
  digitalWrite(PM, LOW);
}
else{
  //Sinon on est l'apres midi, on retranche 12
  analogWrite(HEURES, 255/12*(heure-12)+decalageHoraire);
  digitalWrite(AM, LOW);
  digitalWrite(PM, HIGH);
}
//on utilise la valeur pour les minutes directement
analogWrite(MINUTES, 255/60*(minute2));
delay (150);

```

(Figure 2)

```

137 //=====
138 // verifier la presence du shield ainsi que s'il est fonctionel
139 if (WiFi.status() == WL_NO_SHIELD) {
140   Serial.println("WiFi shield non present");
141   // ne pas continuer
142   while (true);
143 }
144 //=====

```

(Figure 4)

```

355 boolean changementValeurBouton (boolean x, int bouton) {
356   //test des conditions
357   if (digitalRead(bouton) == 0)//si bouton appuyé (donc le pin indique 0 car il est en mode INPUT_PULLUP)
358   {
359     if (x==1){
360       x=0;
361     }
362     else{
363       x=1;
364     }
365   }
366   delay (75);
367   return x;
368 }

```

(Figure 5)

```
227 //fonction qui s'occupe du réglage de l'heure du réveil en deux temps (heures puis minutes)
228 void reglageHeureMinuteDuReveil(){
229     //On entre dans la fonction changementValeurBouton () pour dire si on règle l'heure ou les minutes
230     switchHeureMinute = changementValeurBouton (switchHeureMinute, BTNHEUREMIN);
231     //si switchHeureMinute est à 1 (en mode heure)
232     if (switchHeureMinute == false){
233         while (switchHeureMinute == false && modeHorloge == false){
234             //récupération de la valeur du potentiomètre
235             // réglage des Heures:
236             heureDuReveil = analogRead (POTENTIOMETRE)/(1023/23);
237             //on règle une heure le matin
238             if (heureDuReveil < 12 ){
239                 analogWrite(HEURES, 255/12*(heureDuReveil));
240                 digitalWrite(AM, HIGH);
241                 digitalWrite(PM, LOW);
242             }
243             //on règle une heure l'après midi
244             else{
245                 analogWrite(HEURES, 255/12*(heureDuReveil-12));
246                 digitalWrite(AM, LOW);
247                 digitalWrite(PM, HIGH);
248             }
249             //on regarde si l'utilisateur a changé de mode (heure/minute)
250             switchHeureMinute = changementValeurBouton (switchHeureMinute, BTNHEUREMIN);
251             modeHorloge = changementValeurBouton (modeHorloge, BTNHORLOGEREVEIL);
252             //on reste dans cette boucle tant que l'utilisateur n'a pas appuyé sur le bouton pour changer de mode
253             delay (75);
254         }
255     }
256     //sinon est en mode minute
257     else{
258         while (switchHeureMinute == true && modeHorloge == false){
259             //récupération de la valeur du potentiomètre
260             //réglage des Minutes:
261             minuteDuReveil = analogRead (POTENTIOMETRE)/17.338983051;
262             analogWrite(MINUTES, 255/60*(minuteDuReveil));
263             //on regarde si l'utilisateur a changé de mode (heure/minute)
264             switchHeureMinute = changementValeurBouton (switchHeureMinute, BTNHEUREMIN);
265             modeHorloge = changementValeurBouton (modeHorloge, BTNHORLOGEREVEIL);
266             //on reste dans cette boucle tant que l'utilisateur n'a pas appuyé sur le bouton pour changer de mode
267             delay (75);
268         }
269     }
270 }
```

(Figure 6)

```
272 //Fonction qui sert à réveiller l'utilisateur progressivement durant 20 min
273 void reveilerUtilisateur(int heure, int minute) {
274     //on change le format de l'heure du reveil en une seule variable
275     heureDuReveilEnMinute = minute+60*heure;
276     //on change le format de l'heure actuelle en une seule variable
277     heureActuelleEnMinute = heureActuelle*60+minuteActuelle;
278     // nombre de minutes restantes avant l'heure du reveil
279     minuteAvantHeureDuReveil = heureDuReveilEnMinute-heureActuelleEnMinute;
280     if (minuteAvantHeureDuReveil > -1 && minuteAvantHeureDuReveil < 20){
281         leReveilASonne = 1;
282         int c=19-minuteAvantHeureDuReveil;
283         if (heurePrecedente != heureActuelleEnMinute){
284             //on allume la guirlande progressivement
285             for(int i = 0; i<=59; i++){
286                 strip.setPixelColor(i, (intensite[c]*rouge)/255, (intensite[c]*vert)/255, (intensite[c]*bleu)/255);
287             }
288             strip.show();
289         }
290     }
291     Serial.println(minuteAvantHeureDuReveil);
292     if (minuteAvantHeureDuReveil <= 0 && minuteAvantHeureDuReveil > -10 && leReveilASonne == 1) {
293         tone(BUZZER, 4000);
294         delay(100);
295         noTone(BUZZER);
296         delay (100);
297     }
298     heurePrecedente = heureActuelleEnMinute ;
299 }
300
301 //Fonction qui sert à stopper le réveil
302 void stopReveil (){
303     noTone(BUZZER);
304     for(int i = 0; i<=59; i++){
305         strip.setPixelColor(i, 0, 0, 0);
306     }
307     strip.show();
308     leReveilASonne = 0;
309 }
```

(Figure 7)