**Rapport de Séance 6**

Durant cette séance, j’ai continué à travailler sur le programme, ainsi que sur le design de notre application ***MIT App Inventor***.

1. **Mise en place du mode « veilleuse » :**

Comme expliquer dans le deuxième rapport, notre « veilleuse » fonctionnera grâce au principe d’ombres ou de pochoirs.

En effet, nous créeront deux supports afin de voir lequel des 2 fonctionnera le mieux.

Les deux seront circulaires, courbés et fabriqués dans un matériau opaques, la seule différence sera que :

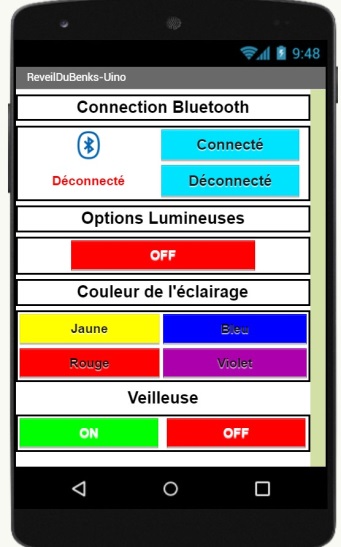
* Le premier, sera pleins avec des trous en forment d’étoiles (pochoir)
* Le second quant à lui, sera « vide » avec des étoiles pleines (ombre)

Sous ce support, nous placeront une « plaque » en bois, recouverte de bandes LED en guise de source lumineuse.

Ce mode, comme tous les autres modes lumineux sera commandé via notre application.J’ai donc ajouté une seconde bande LED au montage de base, ainsi que plusieurs blocs de codes, tant sur celui de l’application, que sur celui de la carte.

Voici ce que j’ai obtenu :

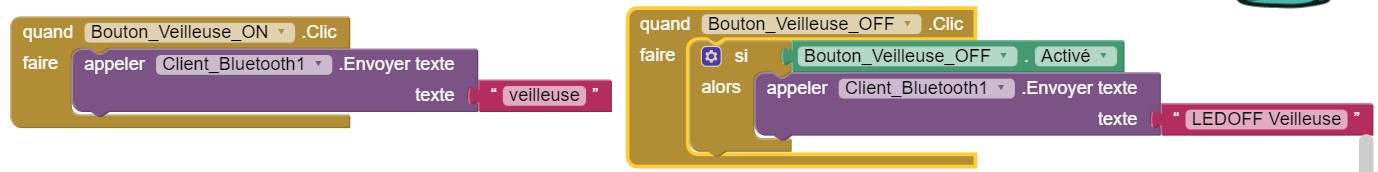
* **Sur *App Inventor*:**



Comme vous pouvez le constater, des boutons *ON* et *OFF* ont été ajoutés, afin d’activer ou non ce mode.

Grâce à l’ajout du second bouton *OFF*, nous pouvons allumer et éteindre, simultanément ou non, les deux modes lumineux de notre réveil.

Ainsi, ils fonctionnent indépendamment l’un de l’autre, ce qui rend leur utilisation encore plus simple, et adaptable à chaque utilisateur !



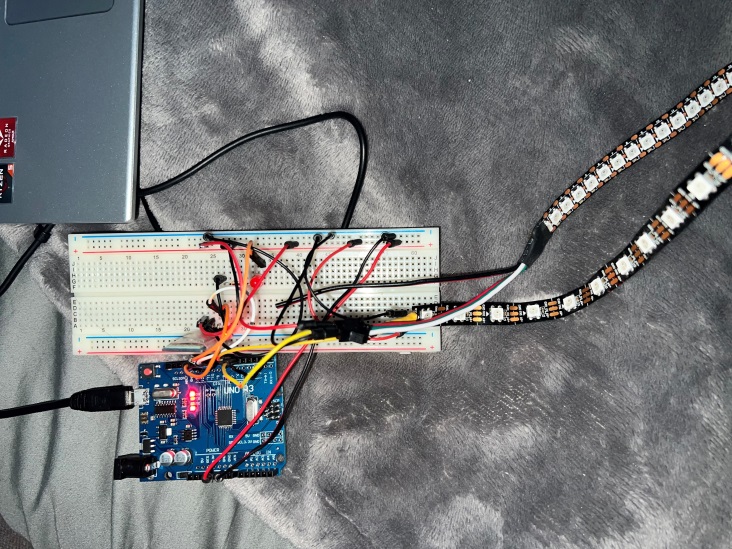
* **Sur le code Arduino :**

Il m’a simplement suffit de rajouter deux blocs « *if* ».

Le premier permet d’allumer la bande LED quand on clique sur le bouton *ON* (la carte reçoit « *veilleuse* »). Le second, permet d’éteindre la bande LED une fois le bouton *OFF* actionné (la carte reçoit « *LEDOFF Veilleuse* »).

Voir le code sur ***Gitub*** nommé : « *Application\_chevet\_\_ veilleuse* »

* **Sur le Montage :**



Vous pouvez consulter la démonstration du fonctionnement de notre application en cliquant sur le lien suivant :

<https://youtube.com/shorts/ACLFyJWFKGg?feature=share>

1. **Mise en place du mode « réveil » sur l’application**:

De base, nous voulions connecter notre réveil à un calendrier, afin que l’alarme se programme automatiquement en fonction de nos rendez-vous.

La date de rendue de projet s’approchant grandement, j’ai décidé de me focaliser dans un premier temps sur un réglage manuel de l’alarme, sur l’application.

Ainsi, si je n’ai pas le temps de m’occuper du calendrier, nous auront quand même un réveil fonctionnel.

Pour cela, je me suis aidé du site référencer dans la partie « ***Sources*** ».

Vous pouvez voir en cliquant sur le lien ci-dessous, une visualisation de ce que j’ai fait pour le moment : <https://youtube.com/shorts/I0NLdrZuKrY?feature=share>

Comme vous pouvez le voir, le module pour choisir l’heure est mis en place.

Il manque donc plus qu’à réussir à l’envoyer à la carte *Arduino* via Bluetooth.

1. **Sources :**

[Utilisation de TimePicker et DatePicker pour entrer des informations d’heure et de date | Coldstreams Learn 2 Code avec MIT App Inventor](Aide%20pour%20le%20réglage%20de%20l'heure)