

PROJET ARDUINO

Année scolaire 2018-2019

“PolyTool”

Etudiants : LENORMAND Damien ; RAKOTOMALALA
Lucas.

Enseignants : MASSON Pascal.

REMERCIEMENTS

Nous tenons à remercier tout particulièrement Monsieur Pascal Masson, professeur d'Electronique, pour son aide non négligeable tout au long du projet, mais aussi Monsieur Abderrahmane, et enfin Monsieur Marc Forner, manager du FabLab de Sophia Antipolis.

SOMMAIRE

<u>Introduction</u>	<i>p.1</i>
<u>I. La base de PolyTool</u>	<i>p.2</i>
a. Le module Bluetooth	
b. La boîte à encoches	
<u>II. DHT 11</u>	<i>p.3</i>
a. Présentation du module et fonctionnement	
b. Problèmes rencontrés	
<u>III. LOGO sensor</u>	<i>p.5</i>
a. Présentation du module et fonctionnement	
b. Problèmes rencontrés	
<u>IV. Pulse sensor</u>	<i>p.7</i>
a. Présentation du module et fonctionnement	
b. Problèmes rencontrés	
<u>V. Photorésistance</u>	<i>p.9</i>
a. Présentation du module et fonctionnement	
b. Problèmes rencontrés	
<u>VI. Les améliorations possibles</u>	<i>p.11</i>
<u>Conclusion</u>	<i>p.12</i>
<u>Bibliographie</u>	<i>p.13</i>

Introduction

Depuis la modernisation de la société, les Hommes cherchent toujours plus de moyens pour simplifier leurs vies. C'est la raison pour laquelle la technologie ne cesse d'évoluer. Ce projet a pour but de faciliter la vie des sportifs. C'est en tout cas son ambition. C'est ainsi qu'une problématique est née :

Comment offrir un outil peu cher et polyvalent, quel que soit le niveau de son utilisateur ?

La réponse qui s'est présentée naturellement est l'utilisation de module peu chers, performants, et donnant le plus d'informations possibles à son utilisateur. Arduino s'y prête spécifiquement bien puisque la communauté y est active, facilitant sa prise en main, pour les plus novices.

Ce projet a donc plusieurs objectifs :

- Offrir un outil délivrant des informations importantes aux sportifs durant leurs entraînements.
- Aider l'utilisateur à dépasser ses limites grâce à l'enregistrement des données de chaque entraînement.

Le fonctionnement du **PolyTool** est simple. Lorsque l'utilisateur sera en train de faire du sport, des informations seront récoltées grâce aux différents modules et vont permettre de lui donner des informations utiles tout au long de son trajet. Le projet aura pour but de :

- ✓ Donner la température ambiante.
- ✓ Donner la température corporelle de l'utilisateur.
- ✓ Allumer des LED en cas de très faible luminosité.
- ✓ Donner le rythme cardiaque de l'utilisateur.
- ✓ Donner la vitesse de l'utilisateur et le nombre de kilomètres parcourus.

La vitesse et le nombre de kilomètre parcourus est donnée pour les personnes étant sur un vélo car nous utiliserons un module qui détecte le passage d'un aimant.

Enfin, pour répondre à la problématique, ce rapport présentera dans un premier temps la base de ce projet, la boîte ainsi que le module Bluetooth. Ensuite les modules utilisés qui, réunis, donne le **PolyTool** seront présenté. Pour chaque module, une description du fonctionnement sera énoncé ainsi qu'un algorithme illustrant ce fonctionnement. Les problèmes rencontrés seront aussi dévoilés.

I. La base de PolyTool

a. Le module Bluetooth

Le module Bluetooth *HC-06* est utilisé pour envoyer les informations de tous les modules décrit précédemment sur le téléphone portable de l'utilisateur.

C'est donc un module essentiel pour le bon fonctionnement de **PolyTool**, puisque sans celui-ci, les données ne pourraient pas être sauvegardées.

Il n'y a pas eu de problèmes notables concernant ce module puisqu'il a été travaillé à de multiples reprises lors des cours d'électronique.

b. La boîte à encoches

Il s'agit de la boîte qui finira par accueillir la plaque et la carte Arduino du projet. Elle a été réalisée sur [ce site](#) et son impression a été faite au FabLab. Ses dimensions sont : profondeur : 81mm, hauteur : 71mm, largeur : 91mm, largeur des encoches : 15mm, épaisseur du matériau : 5mm.

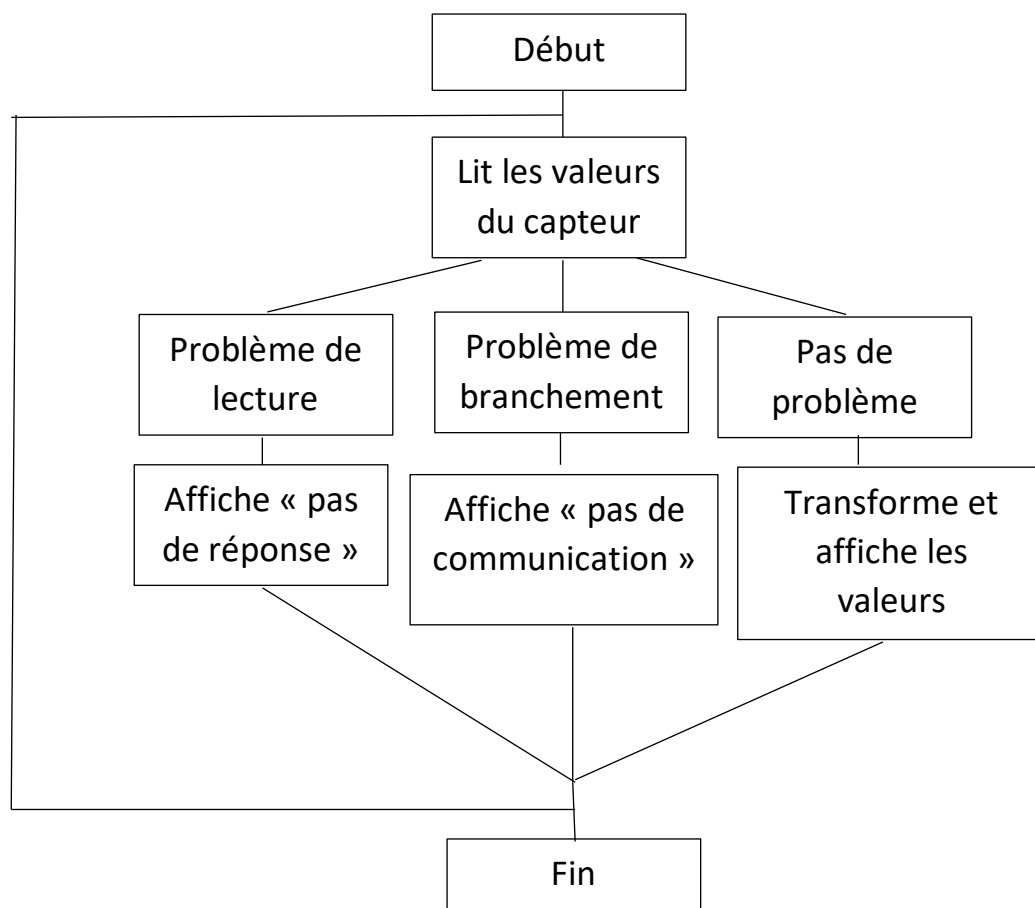
Les dimensions sont également disponibles sur le [GitHub du projet](#), ainsi que le fichier modifié, où les trous dans les surfaces sont mis à jour.

II. DHT11

a. Présentation du module et fonctionnement

Le DHT11 est un module qui permet de capter la température ambiante, ainsi que l'humidité de l'air. Il est disponible à l'achat [ici](#).

Il est utilisé dans **PolyTool** pour donner à l'utilisateur des informations sur l'environnement qui l'entoure.



Dans un premier temps, la carte Arduino va mettre la ligne de donnée dans un état LOW. Pendant un certain temps, le capteur va faire une mesure de la température et de l'humidité.

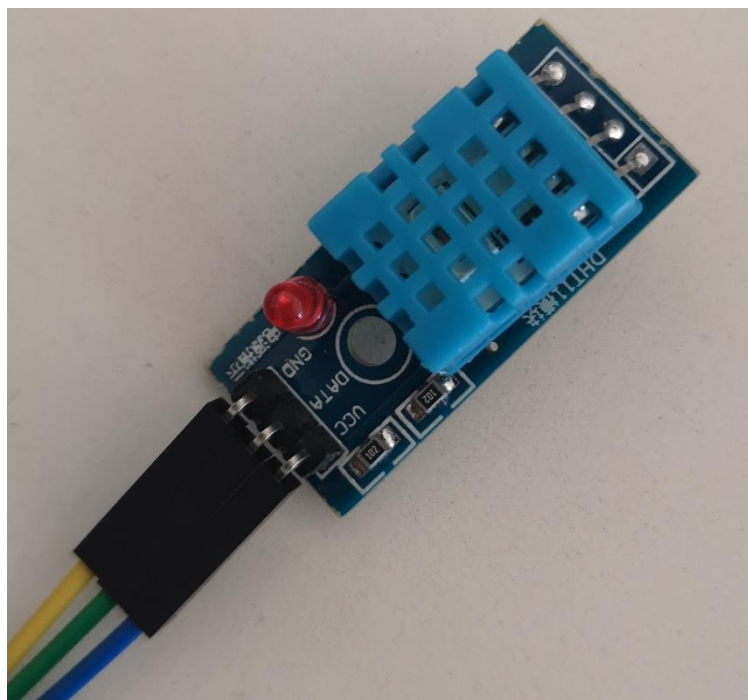
Ces valeurs sont envoyées sous la forme de 5 octets et la ligne de donnée passe à l'état HIGH. Les 2 premiers octets contiennent les mesures de l'humidité, les 2 suivants celles de la

température et le dernier contient des valeurs permettant de vérifier que les valeurs récupérées précédemment sont correct.

C'est ensuite au programme Arduino de transformer les octets de température en degrés Celsius et ceux pour l'humidité en pourcentages. Ces 2 valeurs sont ensuite envoyées sur le téléphone portable.

b. Problèmes rencontrés

Le principal problème lié à ce module a été la compréhension des programmes et du module en lui-même. Ce module utilise des bits/octets pour les données de la température et de l'humidité. Des dysfonctionnements étaient engendrés par la mauvaise utilisation de ces octets.

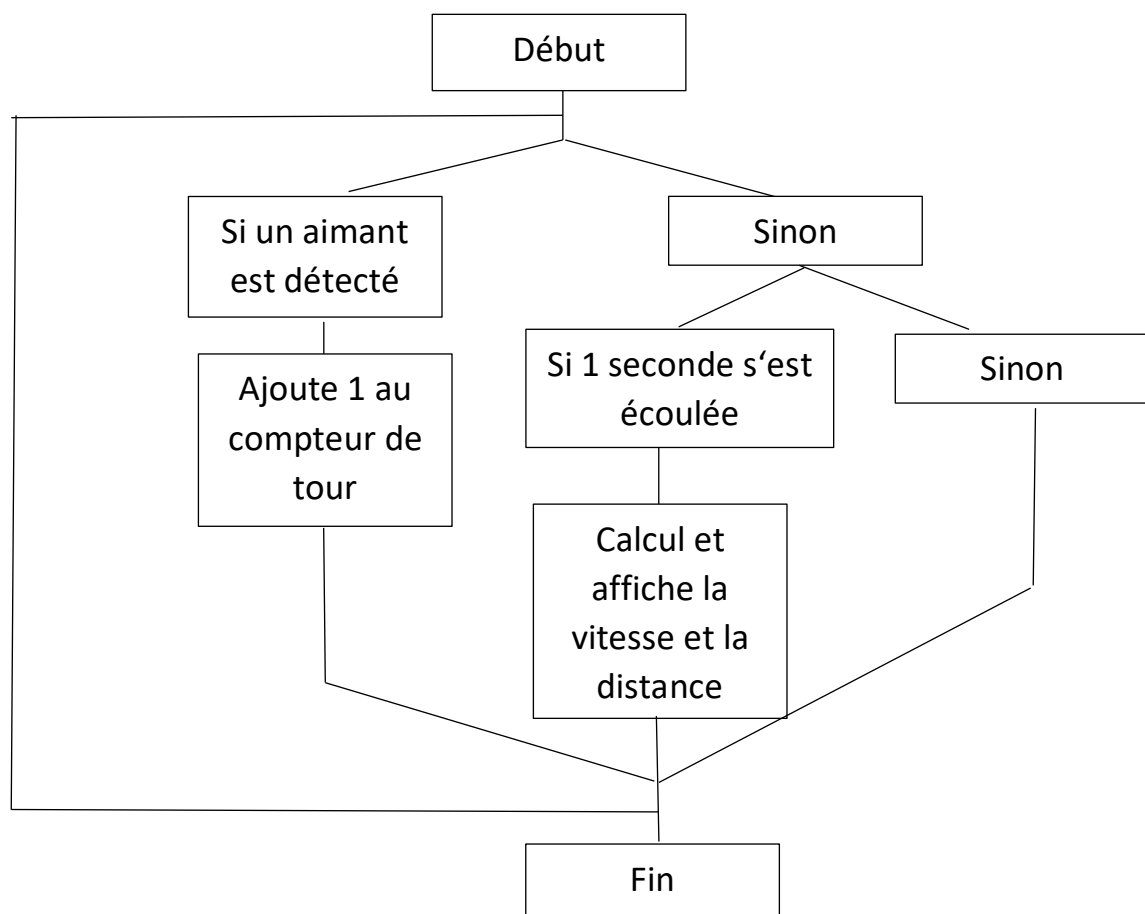


III. LOGO sensor

a. Présentation du module et fonctionnement

Le *LOGO sensor* est un module qui permet de récupérer la vitesse instantanée de l'utilisateur ainsi que la distance qu'il a parcourue. Il est disponible à l'achat [ici](#).

Il est utilisé dans **PolyTool** pour donner à l'utilisateur des informations sur ses performances.



Ce module est composé d'un détecteur infrarouge et il utilise l'effet Hall. Cela veut dire qu'il détecte les champs magnétiques d'une pièce métallique (un aimant par exemple) passant à côté. L'utilisation d'un aimant est donc nécessaire pour faire fonctionner ce module (aimant qui peut être fixé sur la roue d'un vélo par exemple). Lorsque cet aimant passe devant le capteur, le module va passer à l'état *HIGH* et va ajouter 1 à une variable utilisée comme compteur de tour.

Toutes les secondes, le programme Arduino va transformer ce nombre de tour en vitesse et distance. Ces 2 résultats sont ensuite envoyés sur le téléphone portable par le biais du module Bluetooth.

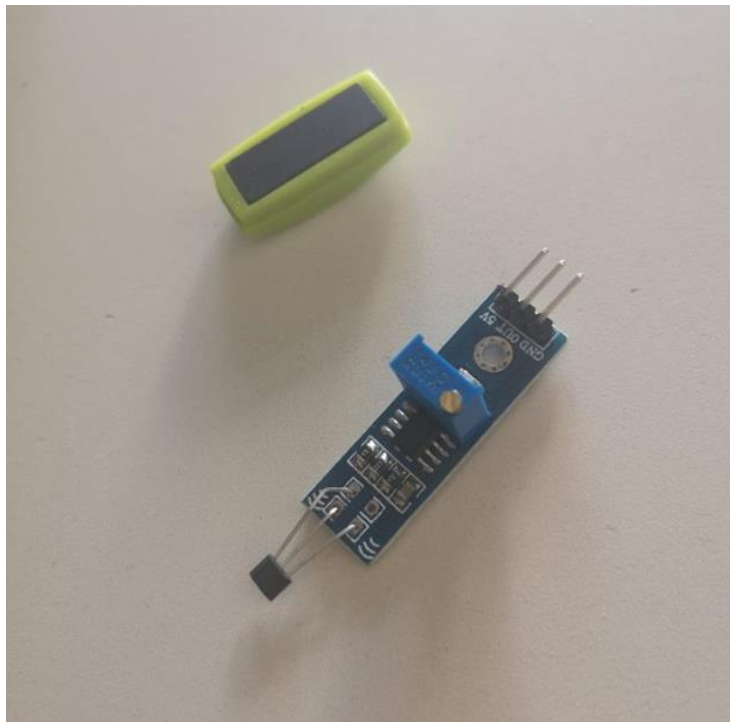
b. Problèmes rencontrés

Ce module a été le plus gros problème de ce projet. En effet, la détection, par le biais d'une interruption, du changement d'état, était assez capricieuse. Le projet utilise le changement d'état pour compter le nombre de tour, et ainsi le nombre de kilomètres parcourus par l'utilisateur. Or, si la détection est mauvaise, ce qui en résulte est nécessairement faussé.

Ainsi, le projet a commencé par utiliser un premier capteur à effet Hall. Ce dernier avait le problème cité plus haut dans ce rapport.

Après l'obtention d'un deuxième capteur à effet Hall, ce problème était en partie résolu, mais jusqu'à ce jour, ce module reste capricieux.

Donc pour pallier ce problème, le changement de capteur a été une bonne idée. Le programme lié à ce module a également été modifié. De ajustements ont dû être fait pour permettre de faire des calculs davantage juste, afin de se rapprocher de l'objectif d'avoir un vrai compteur kilométrique.

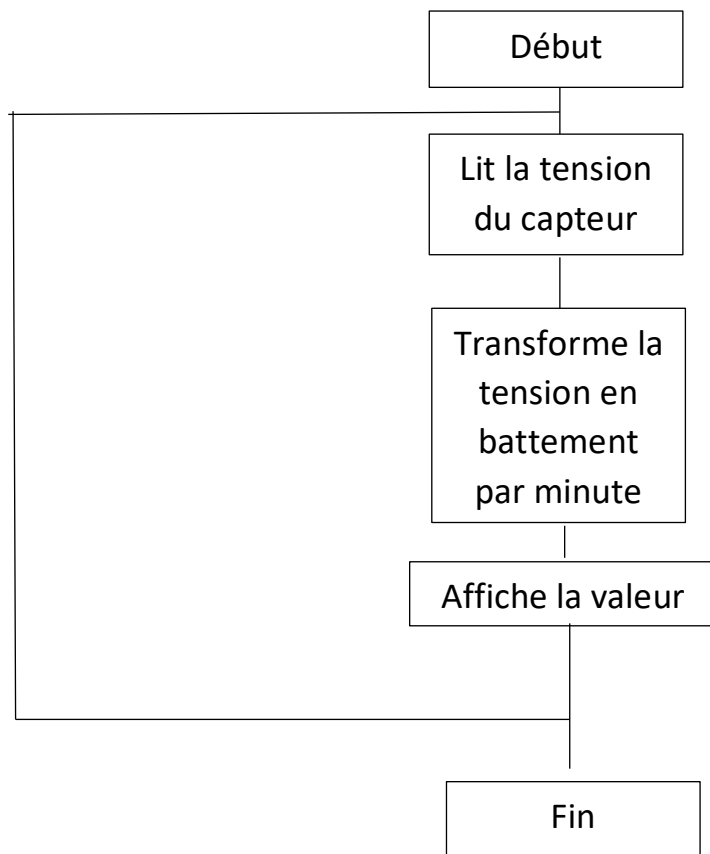


IV. Pulse sensor

a. Présentation du module et fonctionnement

Le *Pulse sensor* est un module qui permet de donner le rythme cardiaque de l'utilisateur. Il est disponible à l'achat [ici](#).

Il est utilisé dans **PolyTool** pour donner à l'utilisateur des informations sur sa santé, et notamment son nombre de battements par minute.



Ce capteur utilise une bibliothèque Arduino précise. Le *Pulse sensor* doit être placé sur le bout du pouce. L'utilisateur ne doit pas trop forcer lorsqu'il le tient sinon tout le sang va descendre dans le doigt et le capteur ne détectera rien. Et si l'utilisateur ne le tient pas assez fort, le capteur va avoir des perturbations liées aux conditions extérieures.

Une mesure est effectuée lorsque la valeur du signal analogique dépasse une valeur seuil. Cette valeur permet de faire changer la sensibilité du capteur. Si ce seuil est élevé, la sensibilité sera très faible, et inversement.

Une fois la mesure prise, la valeur sera affichée sur le téléphone, toujours grâce au module *Bluetooth*.

b. Problèmes rencontrés

Ce module a également été un problème de taille lors du projet. En effet, après la mise en place des différentes parties du Pulse sensor, notamment pour éviter de fausser les résultats lors de la prise de pouls, la détection des battements par minute (bpm) était assez hasardeuse, quand bien même le capteur ait été mis aux meilleurs endroits pour prélever le pouls d'un individu.

Pour pallier ce problème de prise de mesures imprécises, un autre module aurait dû remplacer celui utilisé dans cette version de **PolyTool**.

Malheureusement, les délais de livraisons n'ont pas permis de faire le changement avant la fin des projets.

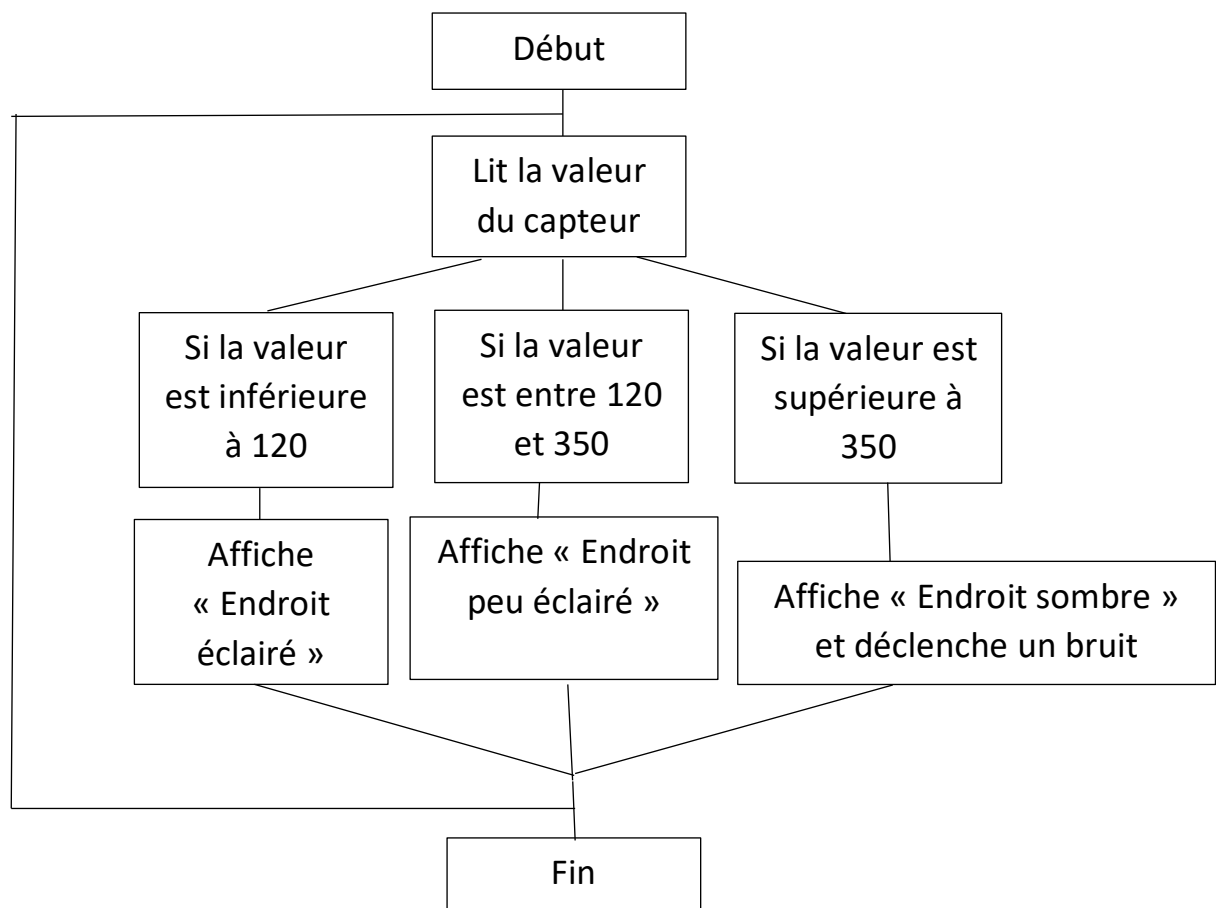


V. Photorésistance

a. Présentation du module et fonctionnement

La *photorésistance* est un module qui détecte la lumière de l'environnement. Il est disponible à l'achat [ici](#).

Il est utilisé dans **PolyTool** pour donner à l'utilisateur des informations sur les conditions extérieures.



La *photorésistance* est un module assez simple. C'est une résistance dont la valeur change en fonction de la lumière reçue.

Le programme va donc comparer les valeurs lues et affichera le message adéquat. Le module va envoyer des valeurs sous forme de tension à la carte Arduino. La carte Arduino va changer

ces valeurs sous la forme de valeurs analogiques comprises entre 0 et 1023. Si cette valeur est inférieure à 120, un message indiquant une bonne luminosité s'affichera. Si elle est comprise entre 120 et 350, un message annoncera une luminosité faible. Et si la valeur relevée est supérieure à 350, un message avertira l'utilisateur que la luminosité est faible, accompagné d'un bruit.

b. Problèmes rencontrés

Il n'y a pas vraiment eu de problèmes concernant ce module car il est très répandu. La documentation est assez claire. Cependant, le calibrage a été fait par tâtonnement, ce qui n'est pas très scientifique. Le calibrage évoqué n'est autre que celui utilisé pour définir si l'endroit est éclairé, peu éclairé ou sombre.



VI. Les améliorations possibles

Il y a plusieurs points qui peuvent être amélioré pour rendre cet outil davantage performant.

Tout d'abord, il y a la boîte. En effet, il serait préférable de présenter une boîte plus ergonomique, moins encombrante pour satisfaire les utilisateurs. D'autant plus que la boîte (à encoches) en bois est relativement lourde, du fait de l'utilisation de planches ayant 5mm d'épaisseur.

Ensuite, l'ajout des modules prévus initialement est une amélioration évidente au projet tel qu'il est actuellement. Parmi les modules prévus initialement, il y a le capteur de température corporelle (*LM335Z*).

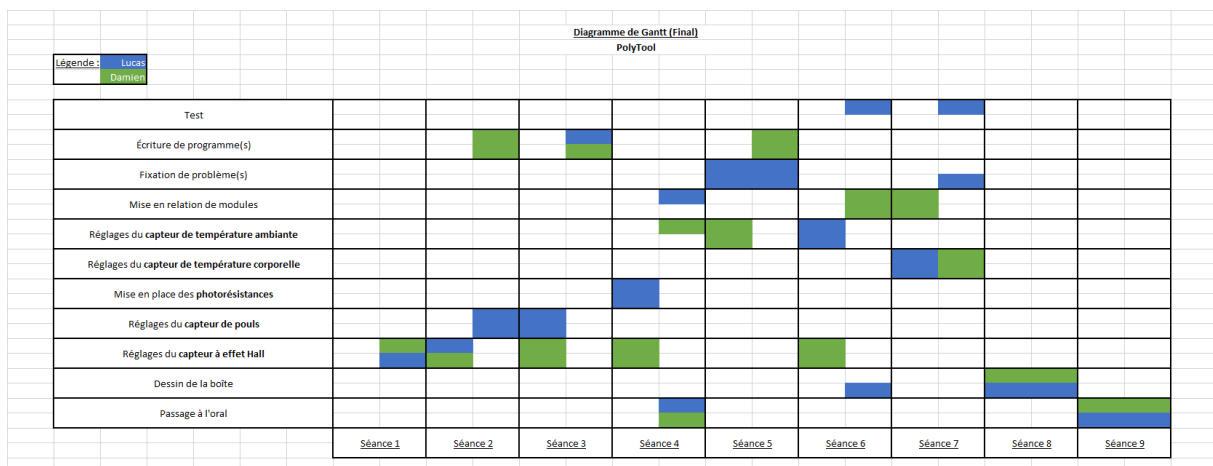
Mais également mieux intégrer les modules sur la boîte. En effet, la révision des positionnements de chaque module sur les faces visibles permettra d'harmoniser l'outil qu'est **PolyTool**.

Conclusion

Finalement, plusieurs modules composent ce projet **PolyTool**. Cependant, certains modules n'ont pas pu être intégré à la version finale proposée, dont notamment le capteur de température corporelle (*LM335Z*). La version finale ne possède également pas les LED qui devaient s'allumer en cas de faible luminosité.

Malgré tout, les autres modules qu'embarque **PolyTool** marchent pour la plupart, c'est le cas de la *photorésistance*, du *DHT11*, ou encore du module *Bluetooth*. Les 2 modules restants (*Pulse sensor* ; *LOGO sensor*) présentent toujours des problèmes, pouvant être résolu avec en changeant de module pour un plus sensible (pour le *Pulse sensor*), ou encore en comprenant mieux l'effet Hall (pour le *LOGO sensor*).

Finalement, le diagramme de Gantt du projet ressemble à cela :



Toutes autres informations (comme les programmes utilisés) se trouvent sur le [GitHub du projet](#).

Bibliographie

- [Carnet du Maker](#)
- [Les Capteurs à Effet Hall](#)
- [Letmeknow - Mise en place de Pulse sensor](#)
- [Wikipédia](#)
- [DatasheetCatalog](#)