



Travaux Pratiques Encadrés

THÈME : TRANSPORTS ET TRANSFERTS – LE TRANSFERT
D'INFORMATION

DISCIPLINES : SVT - MATHÉMATIQUES

SUJET :
COMMENT TRANSMETTRE UN MESSAGE À DISTANCE AVEC DE LA
LUMIÈRE SANS GÊNER A L'ŒIL HUMAIN ?

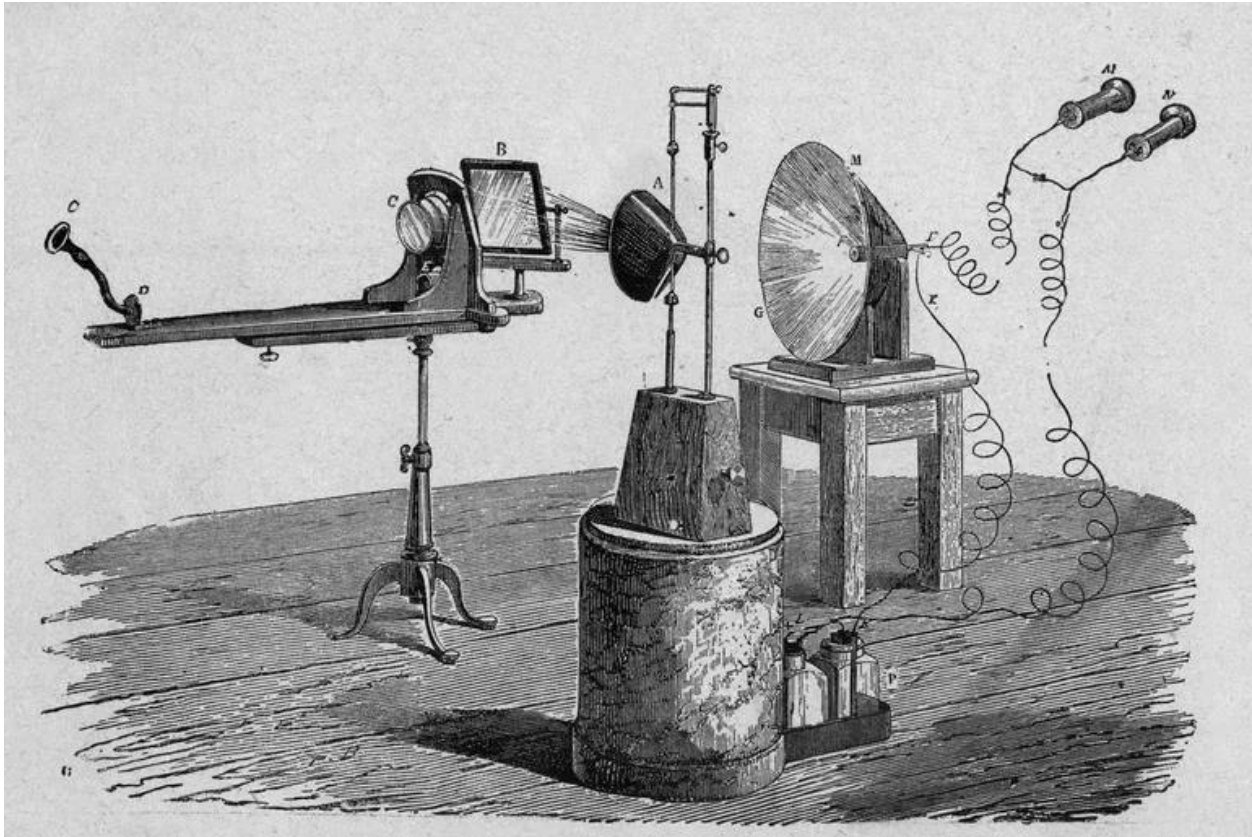
Groupe : Nguyen Phong LE et Nhat Minh DO, 1^{ère} S | Lycée Français Alexandre Yersin – Hanoï,
Vietnam | Année scolaire 2018 – 2019

SOMMAIRE

I.	Introduction -----	1
II.	La persistance rétinienne et la lumière -----	5
	1. La structure et le fonctionnement de l'œil humain -----	5
	2. La persistance rétinienne -----	7
	3. Expérience initiale : À quelle fréquence de clignotement d'une DEL l'œil humain voit-il une lumière continue ? -----	8
	4. Le seuil critique de fusionnement du papillotement -----	9
	5. Conclusion -----	14
III.	Réalisation d'appareils de transmission et de réception par voie lumineuse -----	15
	1. Présentation du projet -----	15
	2. Étude : Fonctionnement d'une télécommande infrarouge -----	15
	3. Conversion d'un message en code binaire et inversement -----	17
	4. La plateforme Arduino -----	18
	5. Composants et circuits -----	19
	6. Algorithme et programmation -----	23
	7. Tests et analyse des résultats -----	30
IV.	Conclusion -----	35
V.	Annexes -----	37

Introduction

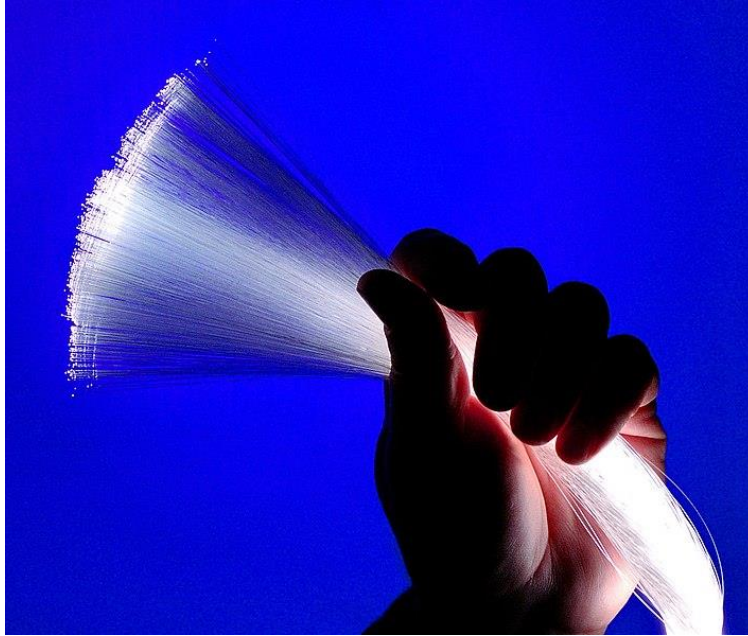
La communication par lumière visible n'est pas un concept nouveau. En fait, elle date depuis la fin du 19^e siècle, avant même l'avènement de la radio comme outil de communication à distance. En 1880, Alexander Graham Bell, aujourd'hui connu comme l'inventeur du téléphone, a transmis un message vocal par moyen des rayons lumineux avec le « photophone », une nouvelle invention que Bell considère sa plus innovante et grande réussite.



Dessin du « photophone » de Bell (Source : thoughtco.com)

Cependant, étant plus susceptible à des interférences qui nuisent au transport des messages que son cousin le téléphone qui utilise de l'électricité pour la transmission, le « photophone » est resté relativement obscur. Néanmoins, l'on peut dire que, d'une certaine manière, cette invention-ci de Bell a été véritablement révolutionnaire.

De nos jours, dans le domaine de transfert de données, la lumière visible est surtout utilisée dans les fibres optiques qui permettent la conduction des rayons lumineux par leur construction en verre ou en plastique. Ces fibres optiques bénéficient d'une vitesse de transmission plus rapide, une plus grande distance de transmission ainsi que plus de fiabilité que les câbles en cuivre.



Fibres optiques (Source : Wikipedia)

L'emploi de la lumière visible dans le transfert des données était donc limité aux câbles physiques. Mais, en 2011, le concept de la Li-Fi a été introduit au public pour la première fois par Harald Haas, un professeur allemand à l'Université d'Edinburgh, en Grande-Bretagne, qui a démontré son fonctionnement au TED Global Talk avec une lampe capable de diffuser une vidéo à un ordinateur qui la projette en temps réel.



Le professeur Harald Haas démontrant la Li-Fi avec une lampe (Source : TED)

La Li-Fi, acronyme de « Light Fidelity », est une méthode de transfert de données sans câble et à distance en utilisant de la lumière visible. Cette technologie se présente comme un successeur potentiel de la Wi-Fi : au lieu d'un modem transmettant des signaux radios, la Li-Fi emploie les sources lumineuses au sein d'une maison pour transmettre les données aux appareils

électroniques tels que les ordinateurs et les smartphones. Ces sources lumineuses vont devoir clignoter afin de transmettre les données.

Selon pureLiFi, l'entreprise pilote de cette technologie, la Li-Fi va permettre un plus grand débit de transfert, ainsi que moins d'interférence (une grande partie du spectre de radio étant déjà utilisée) et plus de sécurité (une autre personne ne pourra pas partager le même signal Li-Fi sans être dans la même pièce, la Li-Fi utilisant de la lumière visible).

L'implémentation de la Li-Fi pourra aussi se faire facilement, les infrastructures qui le permettent, les lampes DEL, étant déjà en place dans de nombreuses maisons modernes. Il suffit d'installer un circuit Li-Fi qui pourrait les contrôler.

Dans nos sources, il est indiqué que les DEL Li-Fi vont clignoter à une vitesse imperceptible à l'œil humain, donnant donc l'impression d'une lumière continue.

Nous nous sommes demandé donc : quel mécanisme de la vision humaine permet-elle d'ignorer les clignotements rapides d'une DEL ? Et pouvons-nous reproduire ce phénomène avec notre propre appareil de transmission par lumière visible ?

Ces questions peuvent être résumées par la problématique suivante : *Comment transmettre des données à distance par de la lumière visible sans gêner à l'œil ?*

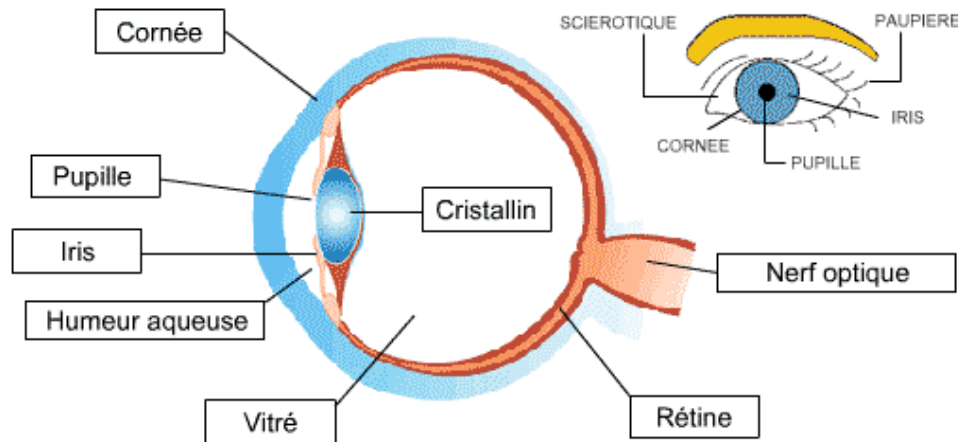
Afin de répondre à cette problématique, notre projet de TPE se fera en deux temps :

- Une étude du ou des mécanismes de l'œil et de la vision humaine qui permettent d'ignorer les clignotements rapides d'une DEL.
- La réalisation d'un appareil de transmission et d'un appareil de réception qui permettent la transmission des messages par voie lumineuse.

I. La persistance rétinienne et la lumière

1) La structure et le fonctionnement de l'œil humain

L'œil, un des organes fondamentaux de l'homme, responsable de la vision du monde, des couleurs, et des formes, est un organe complexe. Il est constitué de 4 parties principales, l'iris, le cristallin, la rétine et le nerf optique.



L'iris

L'iris correspond à la membrane circulaire de la face antérieure du globe oculaire. Il s'agit de la partie colorée de l'œil qui contrôle la quantité de lumière qui rentre dans le cristallin. Le cristallin est percé en son centre d'un orifice/cercle de diamètre varié, qui dépend de la luminosité, qui s'appelle la pupille. La lumière passant par la pupille rentre dans le cristallin.

Le cristallin

Le cristallin peut être comparé à une lentille convergente naturelle de l'œil qui concentre la lumière vers la rétine. Le cristallin possède une courbure qui peut être modifiée (grâce au muscle ciliaire), qui modifie sa vergence pour s'adapter à la distance entre l'objet et l'œil pour pouvoir former une image nette sur la rétine.

La rétine

La rétine est un organe indispensable de la vision. Il s'agit d'une membrane mince, d'épaisseur de 0,5mm, qui recouvre 75% de la face interne du globe oculaire. La partie centrale de la rétine est appelée la fovéa. La rétine est composée de plusieurs couches, dont la couche la plus interne est formée de cellules bipolaires, les cônes et les bâtonnets.

- Les cônes sont l'un des deux cellules photoréceptrices situées dans la rétine, qui transforment les signaux électromagnétiques de la lumière en signal nerveux permettant

la vision des couleurs, et notamment la vision diurne. Les cônes composent seulement 5% des cellules photoréceptrices de la rétine.

Les cônes sont situés principalement sur la fovéa, sur l'axe optique. Cette région est donc dotée d'une grande acuité visuelle. Chaque cône n'est relié qu'à une seule cellule bipolaire (On/Off, On – présence de lumière, le cône réagit, Off – absence de lumière) qui est elle-même liée qu'à une seule cellule ganglionnaire.

Il existe trois types de cônes différents. Les cônes (B), sensibles à des radiations basses, appelés cônes cyanolabes, sont sensibles à une longueur d'onde moyenne 437nm et permet la vision de la couleur bleue. Les cônes (V), sensibles à des radiations de moyennes longueurs d'onde, appelés cônes chlorolabes, sont sensibles à une longueur d'onde moyenne de 533nm et permet la vision de la couleur verte. Les cônes (R), sensibles aux radiations grandes, appelés cônes érythrolabes, sont sensibles à une longueur d'onde moyenne de 564nm et permet la vision de la couleur rouge. Ces trois cônes spécifiques envoient des signaux au cerveau, et la mise en commun des signaux provenant de ces 3 types de cônes permet donc la vision des couleurs.

- Les bâtonnets correspondent au deuxième type de photorécepteurs présent dans la rétine de l'œil. Il s'agit de cellules photosensibles qui permettent la vision nocturne. Les bâtonnets sont situés principalement dans la zone périphérique de la rétine. Grâce à un pigment nommé rhodopsine, le message est traduit en signal bioélectrique, un influx nerveux pris en compte par le cerveau. Cette vision, contrairement à celle que permettent les cônes, n'est qu'en noir et blanc, et des nuances infimes de gris dans la mesure où il n'existe qu'un seul type de bâtonnets. Les bâtonnets composent les 95% de cellules photoréceptrices restantes.

Les bâtonnets, contrairement aux cônes, sont réunis et connectés par groupes aux cellules bipolaires, qui elles-mêmes sont regroupés puis reliées aux cellules ganglionnaires. Les bâtonnets permettent donc une acuité visuelle très faible.

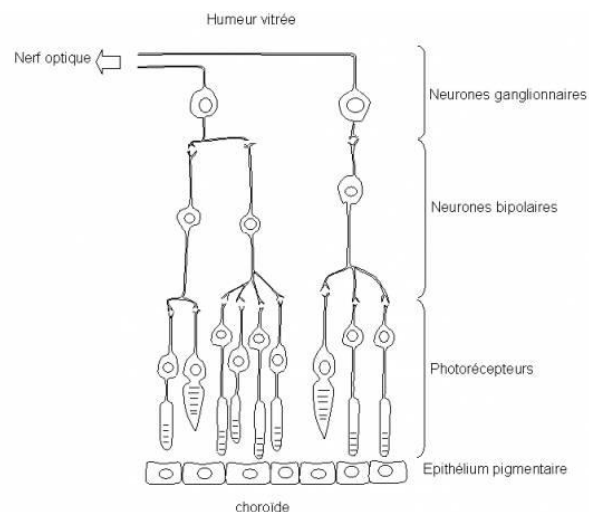


Schéma de la connexion des cônes et bâtonnet au nerf optique

Le nerf optique

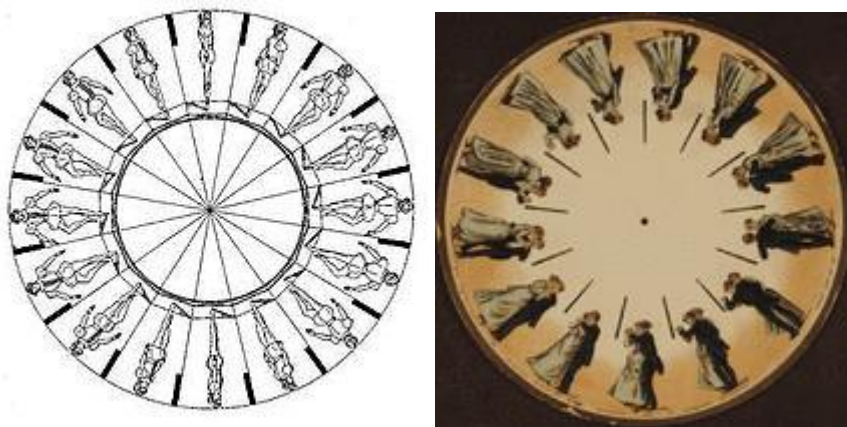
Le nerf optique, réunissant les cellules ganglionnaires, apporte les messages provoqués par les cellules photoréceptrices au cerveau, qui traitera cette information. Il existe 5 aires dans le lobe occipital du cerveau qui sont responsables de traiter cette information, l'aire V1 qui reçoit cette information du nerf optique et le disperse aux autres aires, comme l'aire V3 qui est responsable de la vision des formes, l'aire V4 des couleurs, et l'aire V5 du mouvement et de la profondeur. L'information, après être traitée par les différentes aires, est réunie par l'aire V2 qui formera l'image.

2) La persistance rétinienne

La persistance rétinienne est un phénomène, une illusion optique, qui a lieu lorsque quand l'image d'un objet perçue ne cesse d'être perçue après que la lumière provenant de celui-ci cesse d'entrer dans l'œil. La persistance rétinienne est dû au fait que l'œil et le cerveau ne perçoivent que 10 à 12 images complètement séparées par seconde.

Plusieurs stimulus visuels discontinus sont donc perçus comme continus lorsque ceux-ci atteignent une fréquence suffisante.

Par exemple, un objet utilisant ce phénomène est le phénakistiscope. Il s'agit d'un disque inventé en 1833 par Plateau. Le terme « phénakistiscope » vient du grec « Phenakistiscos » qui signifie la déception. Il s'agit alors d'une illusion optique qui exploite le mécanisme de la persistance rétinienne. Le disque est composé d'une séquence de dix à douze dessins alignés autour du bord de celui-ci. Lorsque le disque est tourné à une vitesse suffisante l'œil perçoit qu'une seule image qui semble être en mouvement constant.



Dessin de disque Phénakistiscope

3) Expérience initiale : À quelle fréquence de clignotement d'une DEL l'œil humain voit-il une lumière continue ?

Nous avons établi précédemment que l'œil humain, lorsqu'il est confronté à plusieurs images en succession l'une après l'autre de manière rapide, peut apercevoir un mouvement continu. Mais, afin que le concept de la Li-Fi puisse fonctionner, il faut impérativement que nous percevons en regardant une DEL transmettant des données non une lumière clignotante ou qui brille à intensités variées, mais une lumière stable et continue. Afin de déterminer cela, nous allons réaliser une expérience.

Hypothèse : Face à une lumière clignotante (qui s'allume et s'éteint) à une fréquence assez rapide, nous allons percevoir une lumière continue.

Expérience :

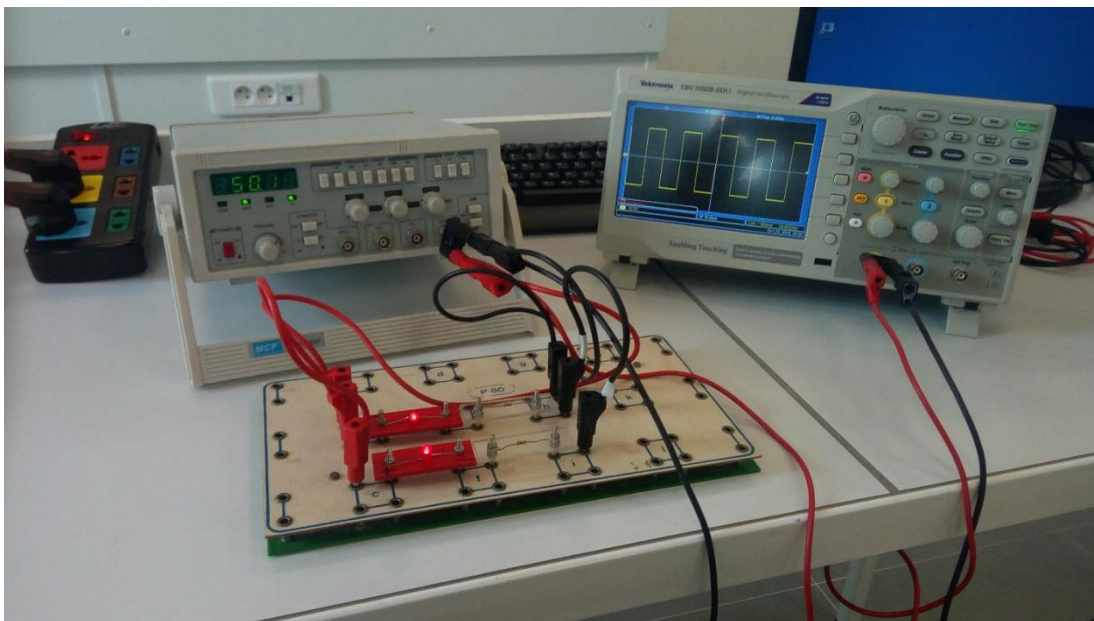
On cherche à déterminer précisément à quelle fréquence minimale de clignotement la lumière d'une DEL sera perçue comme continue.

Protocol expérimental :

1. Réaliser un circuit électrique permettant une DEL de clignoter à des vitesses variées
2. Observer la DEL clignotante tout en faisant varier la vitesse
3. Noter les observations

Montage électrique

Afin de faire clignoter la DEL, nous pouvons utiliser un générateur à basse fréquence. Celui-ci pourra générer des pulses électriques à intervalles réguliers, ce qui constitue un phénomène périodique que nous observerons avec l'aide d'un oscilloscope.



Montage pour clignotement de deux DEL avec un générateur à basse fréquence. Le phénomène périodique généré est montré sur l'oscilloscope.

Observations :

Nous constatons que lorsque la fréquence de clignotement atteint 45 à 50 Hz (autour de 50 clignotements par seconde) la lumière des DEL est perçue comme continue et ne semble pas gêner à l'œil qui le regarde pour une longue durée. Néanmoins, le clignotement est visible lorsque l'expérience est filmée avec la caméra d'un smartphone. En dessous de 45, la lumière semble instable avec une intensité lumineuse qui semble varier et encore plus bas, nous pouvons distinguer clairement les DEL qui s'allument et s'éteignent.

Conclusion :

D'après les observations à la suite de l'expérience, nous pouvons conclure qu'il est certainement possible de faire clignoter une DEL à une fréquence élevée de manière que l'œil humain perçoive une lumière continue. Cette fréquence semble être environ 50 Hz (voir image).

4) Le seuil critique de fusionnement du papillotement

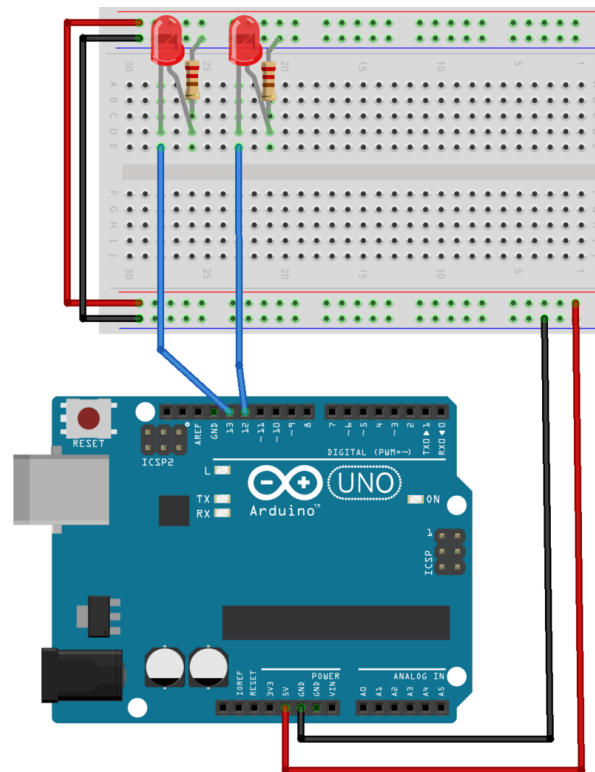
Le seuil critique de fusionnement du papillotement, ou la fréquence critique de fusion, est un phénomène associé à la persistance rétinienne. Il a été observé dans l'expérience précédente : c'est le seuil de la vitesse de clignotement d'une source lumineuse, au-delà duquel une lumière continue est aperçue par l'œil.

Le seuil critique de fusionnement peut dépendre de plusieurs facteurs en dehors de la fréquence de clignotement de la source lumineuses, dont deux desquels vont être démontrés expérimentalement (par manque de temps, les autres facteurs vont être cités, mais pas testés). Cette fois-ci, l'oscilloscope et le générateur à basse fréquence ne seront pas utilisés pour les expériences, mais un circuit comportant une carte Arduino (voir partie 2.3. La plateforme Arduino) qui peut être programmée pour faire clignoter des DEL.

Matériel :

- 2 DEL 5mm, une rouge et une verte
- Une carte Arduino Uno
- Des résistances à valeurs différentes

Montage :



fritzing

Schéma du montage électrique utilisé (la deuxième DEL sera remplacée par une de couleur verte) (Source : learn.adafruit.com)

Code :

```
#define DEL_ROUGE 13
#define DEL_VERT 12
#define FREQ 45 // La fréquence en Hz sera mise ici

void setup() {
  pinMode(DEL_ROUGE, OUTPUT);
  pinMode(DEL_VERT, OUTPUT);
}

void loop() {
  digitalWrite(DEL_ROUGE, HIGH);
  digitalWrite(DEL_VERT, HIGH);
  delay(1000 / FREQ / 2);
  digitalWrite(DEL_ROUGE, LOW);
  digitalWrite(DEL_VERT, LOW);
  delay(1000 / FREQ / 2);
}
```

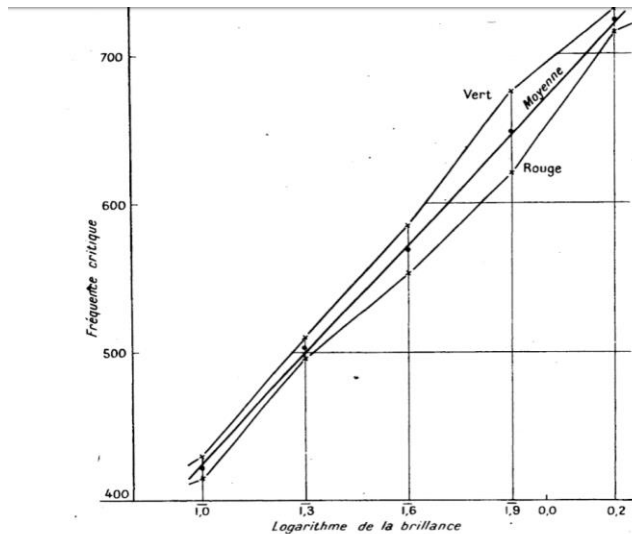
Remarque : Afin d'assurer une cohérence relative au niveau des résultats, la même personne fera les observations en une seule séance. Les résultats auront été plus fiables si plusieurs

observateurs ont pu être introduits, mais par manque de temps, nous n'avons pu faire qu'avec une seule personne.

Facteurs pouvant influencer le seuil critique de fusionnement du papillotement :

1) La longueur d'onde ou les longueurs d'onde de la lumière

Des recherches ont établi que la ou les couleurs de la lumière ont une influence sur le seuil critique de fusionnement du papillotement. Cela peut être dû à la différence entre les différents types de cônes visuels qui perçoivent le bleu, le vert et le rouge individuellement.



Graphique présentant la fréquence critique des lumières monochromatiques en fonction de la brillance (luminosité) et de la couleur. (Source : Étude des fréquences critiques de fusion pour des stimulations chromatiques intermittentes à brillance constante, par Galifret Yvres et Piéron Henri dans L'année psychologique, 1944)

Nous remarquons dans le graphique ci-dessus que la lumière verte nécessite globalement une fréquence plus élevée que la lumière rouge pour atteindre le seuil critique. La même source note aussi une augmentation globale de la fréquence critique avec de la lumière bleue. Nous pouvons donc assumer que la machine finale, qui utilise une lumière blanche générée par une DEL et composée des trois couleurs, doit clignoter au moins à la fréquence critique pour une lumière bleue ou plus rapide.

Vérification expérimentale :

À l'aide du circuit décrit précédemment, nous avons pu faire clignoter les DEL (accompagnées chacune d'une résistance de 470 Ohms) tout en variant la fréquence avec la variable **FREQ**.

À environ 42 Hz, nous n'apercevons plus de changement au niveau de la DEL rouge, mais une légère instabilité chez la DEL verte. Cette instabilité ne peut qu'être vu que de très proche en

fixant tout droit la DEL verte. À 46 Hz, cette instabilité disparaît complètement et les deux DEL semblent émettre une lumière stable.

Nous pouvons donc conclure que la couleur / la longueur d'onde de la lumière a bien un effet sur le seuil critique de fusionnement de papillotement.

2) L'amplitude de la variation, donc la différence entre la luminosité maximale et la luminosité minimale.

Le seuil critique de fusion de papillotement peut aussi varier en fonction de la différence entre la luminosité maximale et la luminosité minimale.

Vérification expérimentale :

Puisque que nous avons affaire à un circuit qui ne peut qu'éteindre ou allumer la DEL (voir la partie 2.4. Composants et circuits), nous allons nous intéresser à la luminosité maximale. Afin de faire celle-ci, nous pouvons faire varier la valeur des résistances connectées aux DEL.

Observations :

<u>Valeur de la résistance (en Ohms)</u>	<u>Fréquence critique de fusion (en Hz) chez la DEL verte</u>	<u>Fréquence critique de fusion (en Hz) chez la DEL rouge</u>
470	46	42
680	46	42
1000	46	42
2200	42	42
4400	41	41
10000	39	39

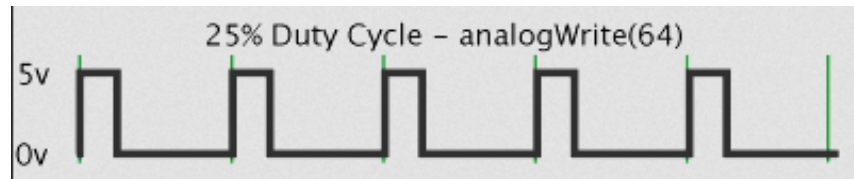
D'après les observations ci-dessus, il semble que la variation de la résistance (et donc de la luminosité de la DEL) a pu changer la fréquence critique de fusion. Nous pouvons observer une baisse globale mais pas du tout égale sur les 3 dernières valeurs testés, avec la diminution la plus dramatique chez la DEL verte.

Nous pouvons donc conclure que oui, la luminosité a bien un effet sur le seuil critique de fusionnement du papillotement, et celui-ci semble descendre pour les luminosités plus faibles.

Remarque : La loi de Talbot-Plateau

Lorsque le seuil critique de fusionnement du papillotement est atteint, intervient la loi de Talbot-Plateau. D'après cette loi, une lumière clignotante en phénomène périodique, une fois que son clignotement dépasse la fréquence critique de fusion, aura sa luminosité aperçue par un observateur comme étant sa luminosité moyenne durant un cycle.

Cette loi est appliquée par un processus appelé modulation de largeur d'impulsion, intégré dans des appareils comme un écran d'ordinateur ou un téléphone portable. La luminosité d'un écran peut varier selon la fréquence du clignotement des DEL intégré dans celui-ci. Par exemple, lorsque nous réglons la luminosité à 25%, sur une période T , la lumière s'allume pour $t_1 = \frac{T}{4}$ (soit 25% du temps) et s'éteint pour $t_2 = \frac{3T}{4}$ (soit 75% du temps) pour créer l'illusion que l'écran est moins éclairé.



3) La luminosité ambiante.

Les cônes permettent une grande acuité visuelle, mais ne possède pas le même niveau de sensibilité à la lumière que les bâtonnets. Cette grande acuité visuelle se traduit aussi par une fréquence critique de fusion plus élevée. Donc, dans un milieu bien éclairé, l'œil est plus sensible aux clignotements que dans les milieux peu éclairés où la vision avec bâtonnets est employée. Les bâtonnets, qui permettent seulement une faible acuité visuelle, ont un seuil de fusionnement du papillotement moins élevé. Dans les conditions de nuit, ceci peut descendre jusqu'à seulement 4 clignotements par seconde.

4) La position de la lumière sur le champ de vision.

Encore une fois, la différence entre les cônes et les bâtonnets joue un rôle déterminant. Sur la rétine de l'œil humain, les cônes sont présents en plus grande densité au fovéa (centre du champ de vision) qu'aux régions périphériques, qui comportent surtout des bâtonnets plus dispersés dans leur distribution. Ainsi, une lumière clignotante nécessite une plus grande fréquence de clignotement lorsque nous la regardons directement en face ou/et de près pour atteindre le seuil critique de fusion que lorsqu'elle est regardée du coin des yeux.

Bien sûr, dans le contexte de la Li-Fi, il est nécessaire de pouvoir regarder directement la lumière du transmetteur Li-Fi en face sans distinguer aucun changement dans la luminosité.

5) La condition psychologique, donc l'âge et la fatigue.

Des recherches ont montré qu'à la suite des travaux nécessitant l'utilisation intense des yeux comme l'étude des documents ou l'utilisation des écrans électroniques, il est possible d'observer des baisses dans la fréquence critique de fusion allant jusqu'à 3 Hz.

Étant donné tous ces facteurs, ainsi que la variation de personne en personne, le seuil critique de fusionnement de papillotement se montre être une statistique plutôt qu'un nombre concret.

Et, comme toute statistique, certains cas virent dans l'extrême. Par exemple, une recherche a montré qu'il est possible qu'un individu, lors des saccades oculaires (mouvements rapides de l'œil entre deux positions stables, pour cibler rapidement un objet/une image), peut percevoir des modifications de luminosité d'une DEL à des vitesses de clignotements en excès de 2 KHz (environ 2000 fois par seconde).

En somme, le résultat de l'expérience de la partie 1.2 ne doit pas être pris comme conclusion finale, mais plutôt comme une ligne directrice pour trouver la fréquence nécessaire lors de la transmission, qui peut être plus élevée.

5) Conclusion

Sans doute, la viabilité de la Li-Fi est en partie grâce aux propriétés de persistance rétinienne et de fréquence critique de fusion de notre perception visuelle.

Cependant, la grande variabilité du seuil critique de fusionnement du papillotement pose un problème : il faudra des fréquences de clignotement extrêmement élevées pour pouvoir éliminer le risque de trouble chez les utilisateurs potentiels.

De plus, jusqu'ici les expériences que nous avons effectuées ne prennent en compte que des clignotements réguliers, avec la DEL allumée et éteinte pour des intervalles réguliers formant des phénomènes périodiques. Or, la transmission de données par un système Li-Fi est loin d'être un phénomène périodique, celui-ci devant encoder et envoyer des informations en modifiant la durée et les moments de clignotements des DEL.

Il faudra donc :

- Un système d'encodage qui permettra de garder une luminosité constante (loi Talbot-Plateau).
- Une vitesse de transmission, ou du moins une fréquence de clignotement suffisamment rapide, dépassant la fréquence critique de fusion de l'œil humain de manière à éviter la gêne à le plus grand nombre de personnes possible, peu importe leur âge, condition physique, et environnement.

Dans notre modélisation d'un système Li-Fi, nous allons essayer d'adresser ces problèmes si possible.

II. Réalisation d'appareils de transmission et de réception par voie lumineuse

1) Présentation du projet :

Afin de démontrer le fonctionnement de la Li-Fi, nous avons décidé de réaliser un transmetteur et un récepteur ayant pour objectifs :

- De transmettre un message en utilisant de la lumière visible. Le transmetteur doit être capable de transmettre :
 - Les 26 lettres de l'alphabet (sans accents) + espace
 - Les chiffres de 0 à 9
 - D'autres caractères présents sur le clavier comme les ponctuations ou « ! », « @ », « # », « \$ », « % » ...
- Si possible, de faire cela en ne pas nuisant à l'œil humain.

La réalisation de ces appareils présentera plusieurs étapes :

- Trouver une plateforme qui permettra la programmation d'un microcontrôleur.
- Trouver un système de codage qui va permettre de convertir le message à envoyer en une forme qui serait communicable par voie lumineuse, notamment en code binaire.
- Trouver les composants électriques nécessaires et réaliser les circuits.
- Écrire les algorithmes pour le transmetteur et le récepteur, puis les programmer.
- Tester et déboguer si nécessaire.

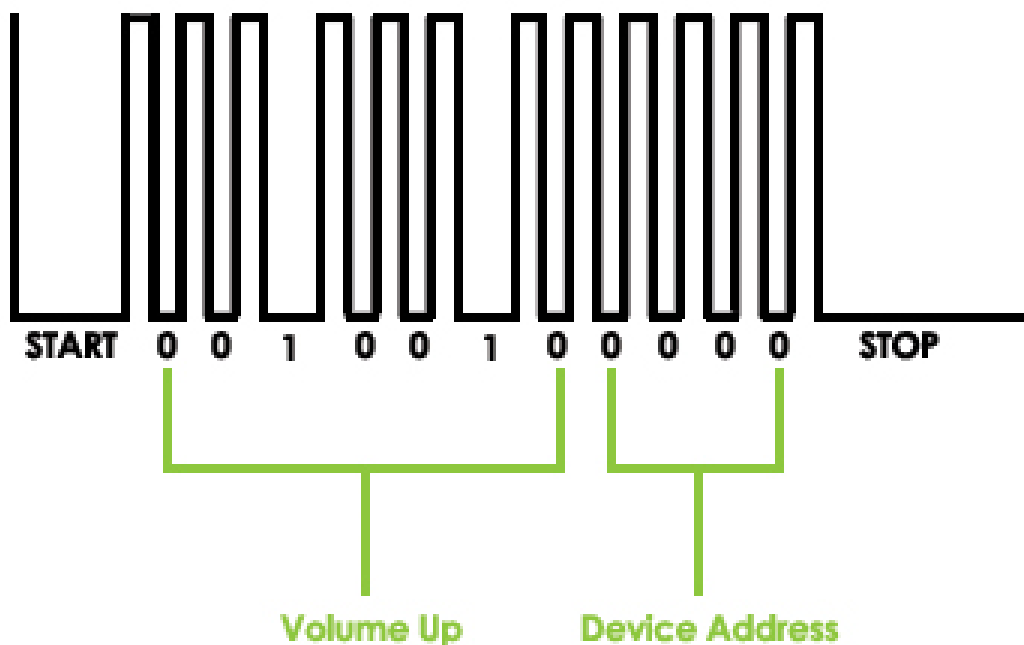
2) Étude : Fonctionnement d'une télécommande infrarouge

Avant de commencer, étudions le fonctionnement d'un objet utilisant de la lumière (infrarouge, qui n'est pas visible à l'œil humain) pour transmettre des données que nous utilisons tous les jours : la télécommande.

Lorsque l'utilisateur appuie sur un bouton, celui-ci touche le contact se trouvant en dessous, et ferme le circuit. Le circuit intégré détecte ce changement et envoie un message en code binaire au DEL qui se trouve au bout de la télécommande et ce DEL clignote en fonction du code reçu. C'est comme du code morse, sauf avec des « 0 » et des « 1 » au lieu des points et tirets. Chaque fonction de la télécommande a un code unique qui lui est associé, que la télévision, munie d'un récepteur infrarouge, va pouvoir reconnaître pour l'exécuter. Aussi, afin d'assurer que la télévision exécute bien la commande qui lui est destinée plutôt qu'à un autre appareil qui utilise

aussi de la lumière infrarouge, un code d'identification unique à elle est aussi attaché au début ou à la fin du message transmis.

Prenons l'exemple d'une télécommande de la marque Sony, chaque appareil pouvant avoir un système de codage différent :



©2005 HowStuffWorks

Schéma présentant le mode de transmission d'une télécommande. (Source : HowStuffWorks)

Ici, le code binaire est pour la fonction "augmenter le volume" et a pour valeur "001001". La télévision pour laquelle il est destiné a pour adresse unique "0000" qui la distingue des autres appareils. La télécommande Sony utilise une méthode de transmission qui consiste à faire varier les durées entre impulsions infrarouges pour représenter des "1" et des "0". D'après le schéma ci-dessus, une plus longue pause représente "1", et une plus courte représente "0".

Comme mentionné avant, une télécommande infrarouge utilise, bien sûr, des rayons infrarouges, contournant donc le problème de nuisibilité à l'œil humain (ces rayons étant en dehors du spectre visible). Un autre avantage est la possibilité d'installer des filtres au niveau du récepteur afin de limiter celui-ci à la réception d'une seule longueur d'onde, réduisant ainsi les risques d'interférence.

3) Conversion d'un message en code binaire et inversement

Comme vu dans l'étude concernant les télécommandes de télévision, nous avons vu que les DEL infrarouges permettent la transmission des codes binaires (en 0 et 1). Nous pouvons assumer que les DEL à lumière visible seront soumises aux mêmes contraintes. Ainsi, afin de transmettre un message contenant des lettres de A à Z (sans accents) (+ espace) et des chiffres de 0 à 9, celle-ci devra être convertie en code binaire pour pouvoir être transmise par le transmetteur puis reconvertie en son état initial du côté du récepteur.

Cela pourra se faire en deux temps :

1. Conversion du message en code ASCII : Acronyme de « American Standard Code for Information Interchange », ASCII est un système d'encodage de caractères pour les communications électroniques et est utilisé dans de nombreux appareils électroniques pour stocker et traiter des textes. Dans ce système, chaque caractère est représenté par un nombre en base 10 (colonne Dec) :

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	Start of Header	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	Start of Text	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	End of Text	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	End of Transmission	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	Enquiry	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	Acknowledgment	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	Bell	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	Backspace	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	Horizontal Tab	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	Line feed	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	Vertical Tab	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	Form feed	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	Carriage return	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	Shift Out	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	Shift In	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	Data Link Escape	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	Device Control 1	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	Device Control 2	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	Device Control 3	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	Device Control 4	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	Negative Ack.	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	Synchronous idle	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	End of Trans. Block	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	Cancel	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	End of Medium	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	Substitute	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	Escape	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	File Separator	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	Group Separator	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	Record Separator	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	Unit Separator	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		Del

asciichars.com

Les lettres de A à Z sont représentées par des nombres de 65 à 90 (lettres majuscules) et de 97 à 122 (lettres minuscules), tandis que les chiffres de 0 à 9 le sont par des nombres de 48 à 57. Il est aussi possible de représenter avec le système ASCII les caractères tels que « @ », « # », etc., ainsi que des ponctuations.

2. Conversion du code ASCII (nombre en base 10) en code binaire :

Pour transformer un nombre en base 10 en code binaire, on utilise la formule suivante : nous pouvons le diviser plusieurs fois par 2, prenant à chaque fois le reste obtenu (qui sera forcément 0 ou 1) et arranger les chiffres obtenus dans cet ordre de droite à gauche. Prenons comme exemple le nombre 66 (lettre « B » en code ASCII) :

$$66 = 33 \times 2 + 0$$

$$33 = 16 \times 2 + 1$$

$$16 = 8 \times 2 + 0$$

$$8 = 4 \times 2 + 0$$

$$4 = 2 \times 2 + 0$$

$$2 = 1 \times 2 + 0$$

$$1 = 0 \times 2 + 1$$

Code obtenu : 1000010.

Finalement, du côté du récepteur, il faudra décoder le code binaire :

Soit un code binaire. Considérons ce code comme une série de valeurs x ayant pour nombre de valeurs n . Pour la convertir en base 10, le premier chiffre du code est multiplié par 2^{n-1} puis le deuxième chiffre par 2^{n-2} et ainsi de suite jusqu'à ce que le dernier chiffre du code est multiplié par 2^0 . Ensuite nous faisons la somme de tous ces produits pour obtenir le nombre en base 10. Ce processus peut être décrit par la formule suivante, avec d le nombre en base 10 :

$$d = \sum_{i=1}^n (x_i \times 2^{n-i})$$

Prenons comme exemple le code 1010011. On constate le code est composé de 7 chiffres :

$$1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 83$$

On obtient le nombre 83 qui, en code ASCII, correspond à la lettre S.

4) La plateforme Arduino

Une fois l'idée et les contraintes des appareils en place, nous avons tourné immédiatement vers la plateforme Arduino pour le circuit. Les microcontrôleurs Arduino sont connus pour leur simplicité par rapport aux autres comme AVR ou PIC et sont donc plus abordables pour les débutants, ayant plusieurs avantages :

- Les microcontrôleurs Arduino ont déjà un "bootloader" installé, éliminant le besoin d'un programmeur, et les cartes telles que l'Arduino Uno ont déjà un circuit en place pour assurer le fonctionnement correct du microcontrôleur, accélérant le procès de prototypage.



Une carte Arduino Uno

- Le langage de programmation utilisé par la plateforme Arduino est une version simplifiée du langage C, facilitant la programmation en éliminant les aspects plus compliqués présents dans les microcontrôleurs AVR ou PIC comme la programmation des registres ou de l'horloge interne.
- Son usage répandu dans les domaines d'éducation (Arduino est utilisé comme plateforme électronique pour enseigner la robotique dans certaines écoles), dans les projets professionnels ou encore chez les amateurs de l'électronique donne lieu à des ressources documentaires abondantes auxquelles il est possible de faire référence, ce qui facilitera la réalisation du circuit et la programmation.

Alternativement, il existe la plateforme PICAXE qui elle-aussi cible les étudiants en présentant un moyen accessible pour apprendre la robotique. Néanmoins, à cause de sa disponibilité limitée au Vietnam, les cartes Arduino, qui, en revanche, peuvent être facilement obtenus ici, restent notre choix pour ce projet.

5) Composants et circuits

Pour notre prototype, nous allons utiliser la carte Arduino Uno, qui est la plus basique dans la famille Arduino. Cependant, dans les appareils finals, nous souhaitons réduire la taille physique du circuit en employant deux circuits imprimés de taille 5 x 7 cm, un pour le transmetteur et un pour le récepteur. Pour faire cela, il est nécessaire de ne pas utiliser une carte Arduino Uno en elle-même mais la reconstituer sur les circuits imprimés.

Le circuit suivant fonctionne comme une carte Arduino Uno, avec les composants essentiels à son fonctionnement :

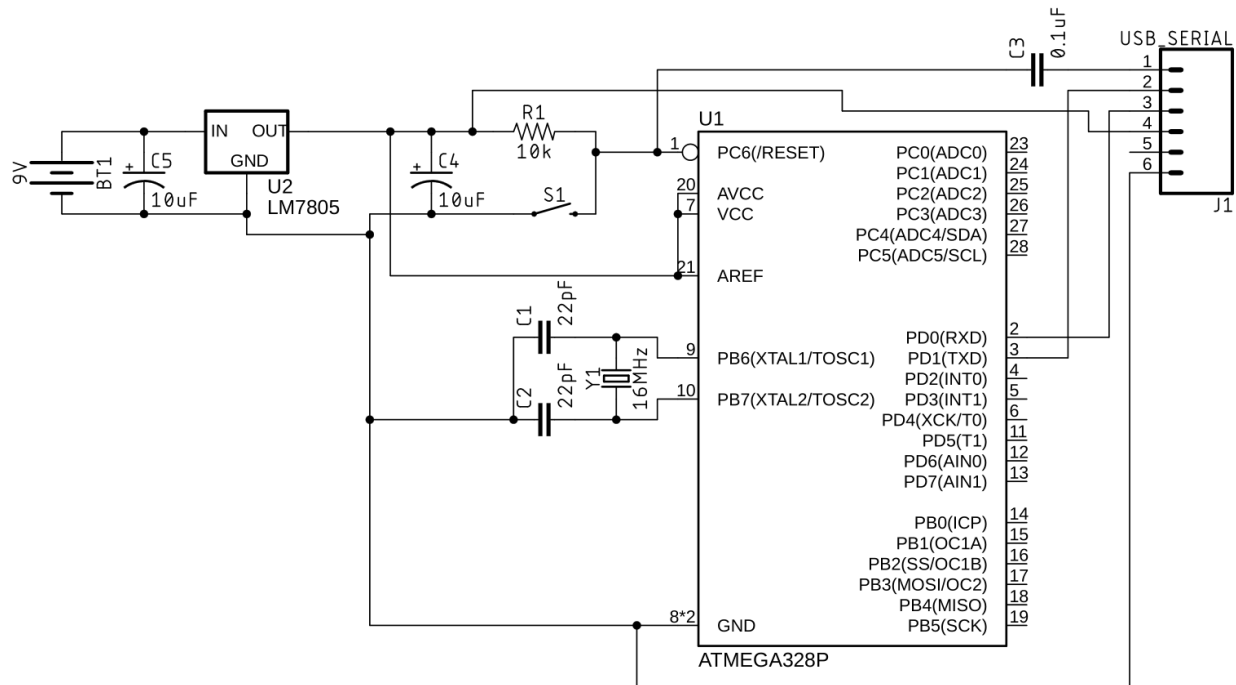


Schéma électrique simplifié d'une carte Arduino Uno (réalisé avec le logiciel EAGLE)

Il comporte :

- Un microcontrôleur ATmega328P muni du bootloader Arduino (programme préinstallé qui permet de programmer le microcontrôleur)
- Un oscillateur à quartz de fréquence 16 MHz
- 2 condensateurs céramiques de capacitance 22 pF
- Un régulateur de tension linéaire LM7805
- 2 condensateurs électrolytiques de capacitance 10 µF
- 1 bouton
- Un condensateur céramique de capacitance 0,1 µF
- Une résistance de 10 K
- Un module d'interface USB (qui permettra de programmer et de communiquer avec le microcontrôleur). Le module utilisé a pour numérotation de ports suivant :
 - 1) DTR (connecté au port RESET du microcontrôleur)
 - 2) RXD (entrée d'information, connecté au port TXD du microcontrôleur)
 - 3) TXD (sortie d'information, connecté au port RXD du microcontrôleur)
 - 4) VCC (5 V)

5) CTS

6) GND (0 V)

Le bouton S1 sert comme bouton de redémarrage pour le microcontrôleur, qui peut être appuyé en cas d'erreur du logiciel.

Une pile de 9 V est utilisée pour représenter la source d'électricité / le générateur. Le microcontrôleur fonctionnant à 5 V, les composants U2, C4 et C5 forment ensemble un circuit régulateur qui assurent un courant stable de tension 5 V, quel que soit la source utilisée à condition que sa tension est entre 7 et 25 V.

Maintenant, il suffit de combiner ce circuit avec ceux du transmetteur et du récepteur :

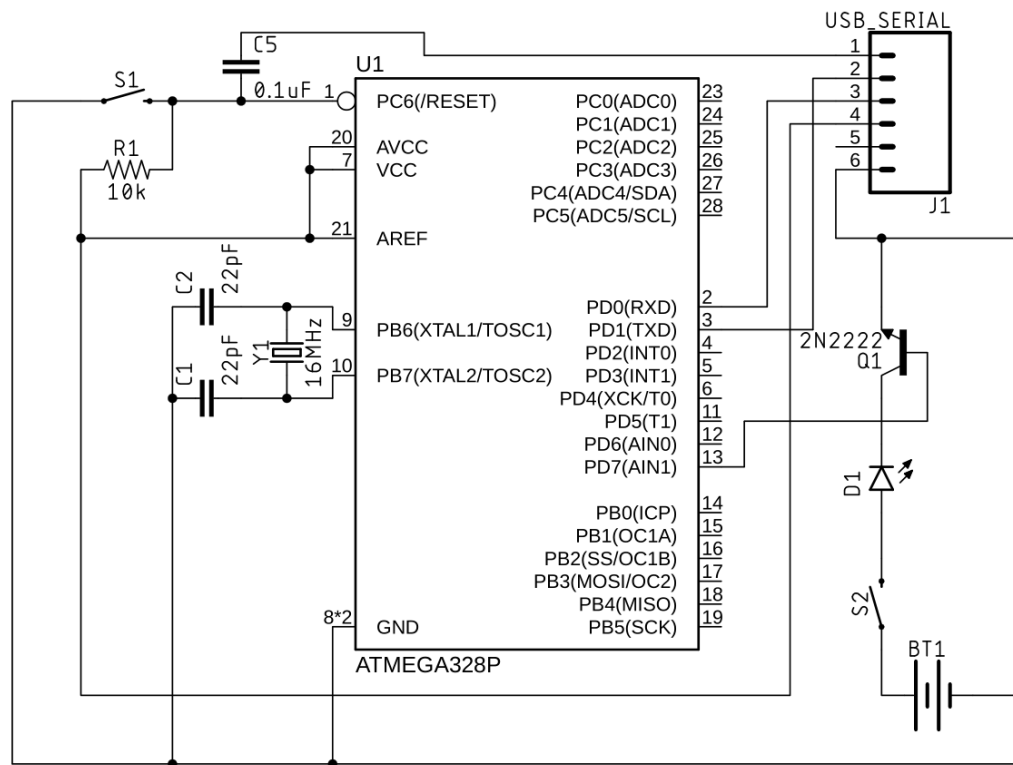


Schéma électrique du transmetteur

Nous avons supprimé le circuit de régulation de tension, le transmetteur devant être tout le temps connecté à l'ordinateur afin de recevoir les messages venant de celui-ci et donc peut s'alimenter uniquement du courant venant du port USB. Cependant, une source d'électricité secondaire, constitué de 3 piles AA, donc 4,5 V, est aussi utilisée pour alimenter la DEL D1 (qui transmettra les messages au récepteur) qui aura une puissance de 3W, pour maximiser la luminosité. Le microcontrôleur pourra faire clignoter la DEL par moyen du transistor Q1, connecté au port 13 (Port Digital Arduino 7).



Le circuit du récepteur (schéma électrique à la page précédente), quant à lui, semble compliqué mais les principes de son fonctionnement et les rôles des composants sont assez simples :

- Le panneau solaire joue ici le rôle de capteur de lumière. Il est connecté au port 28 (Port Analogique Arduino 5). Les ports analogues du microcontrôleur permettent de mesurer la tension du composant qui lui est connecté.
- De même, le potentiomètre VR2 est connecté au port 23 (Port Analogique Arduino 0). Il servira à la calibration du panneau solaire (plus de détail dans la partie suivante).
- Le bouton S2, connecté au port 16 (Port Digital Arduino 10), servira à mettre le récepteur en mode attente (prêt à recevoir un message) ou en mode calibration.
- La DEL D1, connecté au port 19 (Port Digital Arduino 13) sert ici comme outil de débogage : elle s'allume lorsque la tension générée par le panneau solaire est suffisamment importante.
- Finalement, l'écran LCD, avec une résolution de 16 x 2 (2 lignes de 16 carreaux, chaque carreau pouvant afficher un caractère), nécessite plusieurs ports. Il est connecté au ports 5, 6, 11, 12, 13 et 14 (Ports Digitaux Arduino 3, 4, 5, 6, 7 et 8 respectivement). Le potentiomètre VR1 sert à régler le contraste de l'écran.

6) Algorithme et programmation

1) Codage binaire

Comme mentionné précédemment, la transmission d'un message se fait en plusieurs étapes. Le premier est, bien sûr le codage en code binaire. La conversion de caractère en code ASCII est très simple, il s'agit juste de mettre le nombre correspondant. Le procès de conversion de nombre en base 10 en code binaire, quant à lui, a été expliqué dans la partie 3.3. Conversion d'un message en code binaire et inversement. Il reste donc juste à le convertir en algorithme.

```
Tableau : T (liste de chiffres binaires)
Variables : c (caractère à convertir), a, i
Début_algo
| Saisir c
| a ← code ASCII de c
| i ← 0
| Tant que a > 0 faire
| | T[i] ← reste de a // 2
| | a ← a // 2
| | i ← i + 1
| Fin_tantque
| Afficher T
Fin_algo
```

Avec l'algorithme ci-dessus, T sera la liste des chiffres qui compose le code binaire du caractère c. Pour un mot ou une phrase, il suffit de les séparer en caractères individuels puis les soumettre à cet algorithme par moyen d'une boucle.

De même que la conversion base 10-binaire, le processus inverse a aussi été expliqué précédemment. Son algorithme :

Tableau : T (liste de chiffres binaires)
 Variables : n, c
 Début_algo
 Saisir T
 $c \leftarrow 0$
 $n \leftarrow \text{longueur de T}$
 Pour i allant de 1 à n faire
 $c \leftarrow c + T[i] * 2^{N-1}$
 Fin_pour
 $c \leftarrow \text{caractère correspondant au code ASCII } c$
 Fin_algo

2) Transmission du code binaire

Notre idée initiale pour la transmission de données est simple : lorsque la DEL du transmetteur s'éteint, c'est un « 0 », et lorsqu'elle s'allume, c'est un « 1 ». Chaque chiffre binaire est appelé un bit. La DEL va rester allumée ou éteinte pour une durée précise avant de passer au prochain bit. Afin de permettre au récepteur de distinguer entre différents caractères, chacun de ceux-ci seront envoyés en série de 8 chiffres binaires (les codes binaires avec moins de 8 chiffres seront complétés avec des « 0 »), ce qui permet d'exprimer les nombres en base 10 allant de 0 à 255. 256 nombres sont largement suffisants pour exprimer tous les caractères dans le système ASCII standard, qui ne comporte que 127.

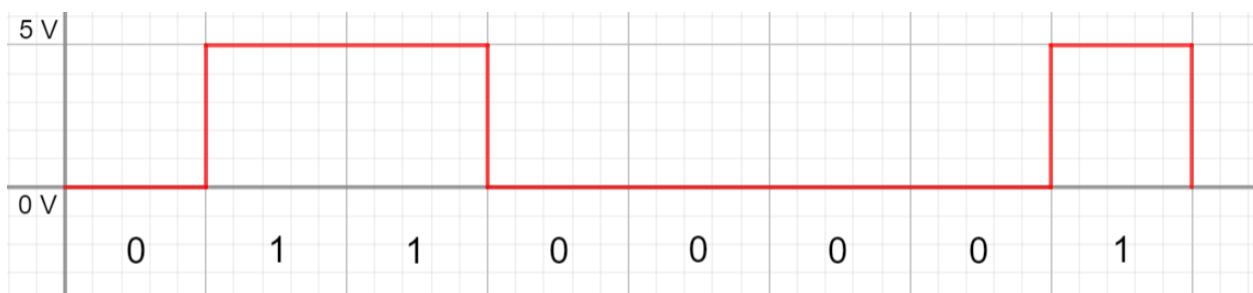


Schéma présentant la transmission du caractère « a » (code ASCII 97, code binaire 1100001). Un zéro a été ajouté au début pour atteindre 8 chiffres. Un grand carré représente une intervalle T (en secondes), la durée de l'envoi d'un bit.

Cette méthode, la plus simple à implémenter, pose néanmoins un problème au niveau de la nuisance à l'œil humain : la durée totale du rayonnement de la DEL n'est pas toujours égale au temps durant lequel elle reste éteinte. Donc, même si la fréquence de clignotement est

suffisante, il est toujours possible d'apercevoir des changements de luminosité (loi de Talbot-Plateau).

Afin d'avoir toujours la DEL qui soit éteinte et allumée pour les mêmes durées durant l'envoi du message, nous allons utiliser le code Manchester, nommé après l'université de Manchester dans lequel il a été développé. Le code Manchester consiste d'un changement d'état au milieu du bit.

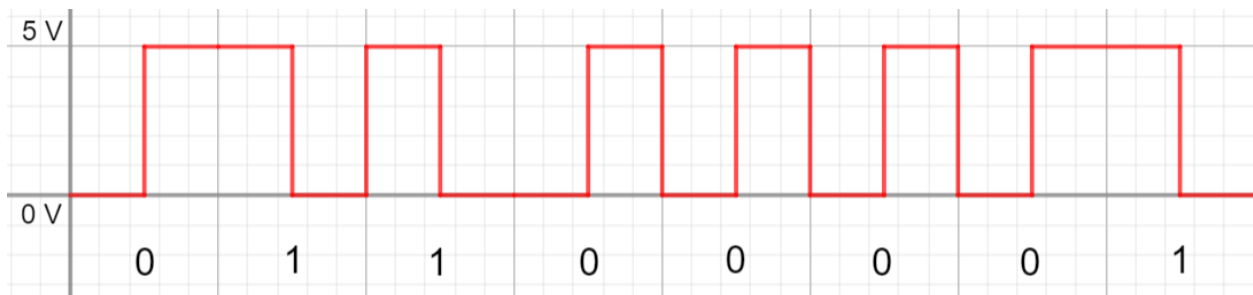
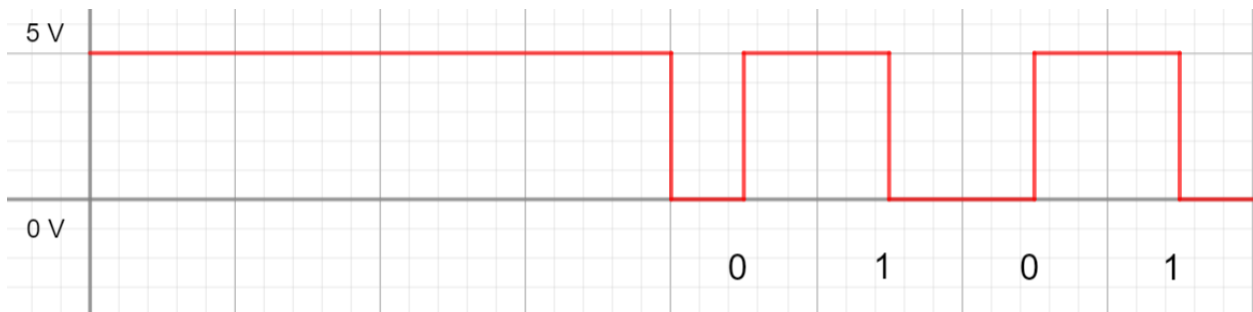


Schéma présentant la transmission du caractère « a » en code Manchester. Pour un « 0 », la DEL s'éteint pour $T/2$ et s'allume pour $T/2$ dans cet ordre, et pour un « 1 » inversement.

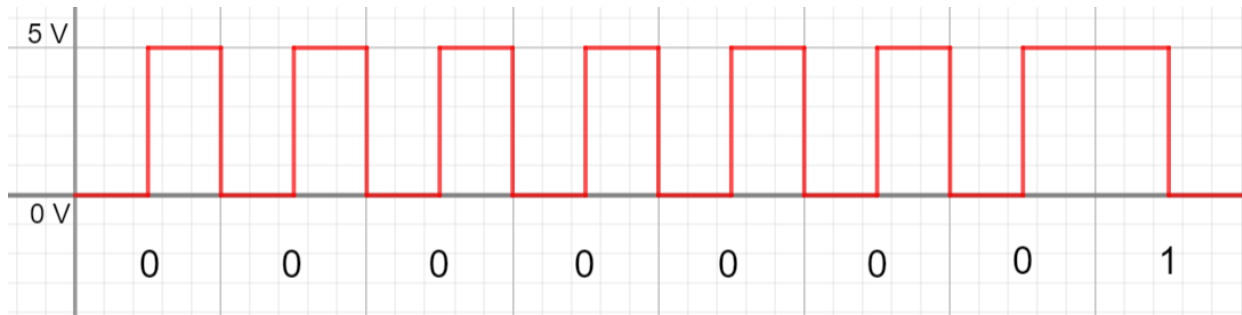
L'état au repos du transmetteur aura la DEL déjà allumée, afin de simuler les conditions réelles de la Li-Fi qui devra transmettre des données à partir d'une source de lumière active. Afin d'éviter le déclenchement de la lecture de manière non-intentionnelle (un objet opaque passant entre le transmetteur et le récepteur, déplacement du transmetteur ou du récepteur, changement de la luminosité ambiante...), une commande « commencer » doit être transmise au début afin que le récepteur reconnaisse qu'un message va venir. Ensuite, la longueur du message (nombre de caractères à lire) doit être envoyée pour que le récepteur sache quand arrêter la lecture. Puisqu'elle sera aussi envoyée en série de 8 chiffres binaires, la longueur maximale d'un message sera théoriquement de 255 caractères. Finalement, les caractères du message, convertis en codes binaires avec les algorithmes précédents, seront envoyés.

Notre commande « commencer » sera « 0101 ». Transmission de la lettre « a » avec ce système :

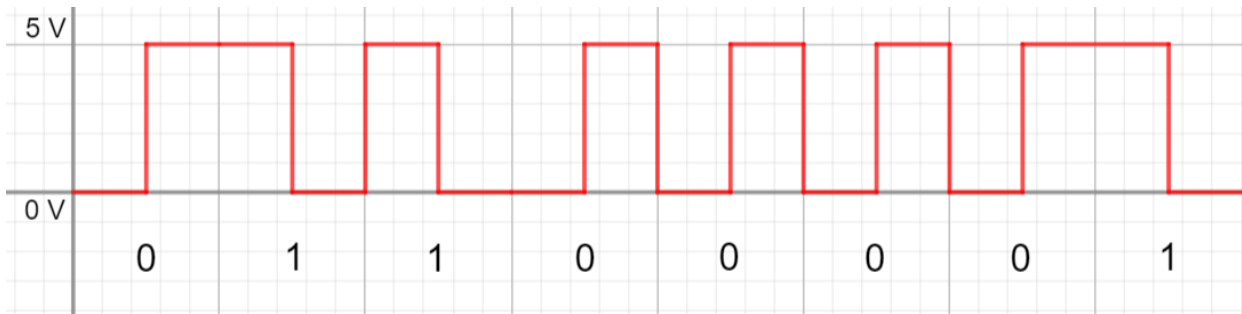
- 1) Envoi de la commande « commencer » :



- 2) Envoi de la longueur du message, 1 dans ce cas-ci :



3) Envoi du caractère « a » :

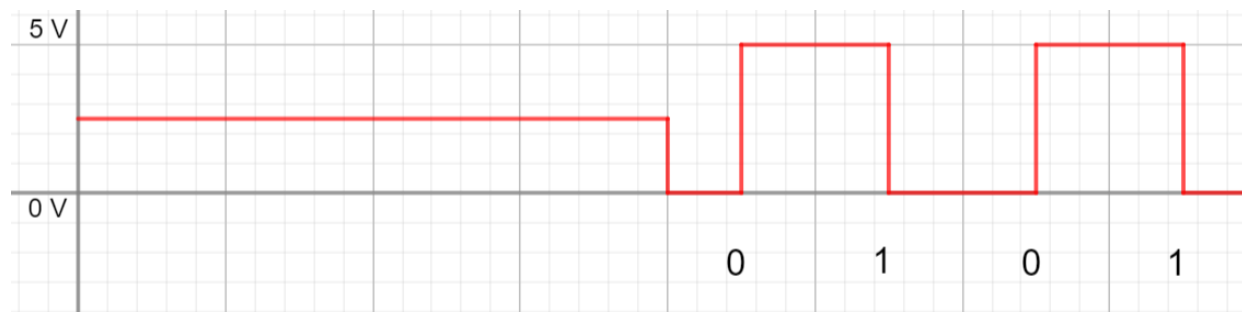


Entre chaque série de 8, il y aura une pause d'environ 5 millisecondes pour permettre au récepteur le temps d'enregistrer la série précédente avant de passer à la suite. Durant cette pause, le transmetteur devra faire clignoter la DEL à une fréquence de $1/T$ Hz.

Remarque :

Sur la durée T_{total} que prend le transmetteur pour transmettre le message, nous observons que la DEL est allumée pour $T_{total}/2$, ce qui se traduit théoriquement par une luminosité de 50% lors de la transmission si l'œil ne perçoit pas le clignotement (loi de Talbot-Plateau).

Donc, idéalement, la tension qui traverse la DEL en attente doit être environ 2,5 V pour qu'aucune différence ne soit percevable. Mais nous n'avons pas pu tester cette hypothèse, car le microcontrôleur utilisé ne peut pas faire sortir des tensions autre que 5 V. (Néanmoins, cela ne pose pas de problème dans un système Li-Fi qui doit transmettre des données en continu)



Algorithme pour la transmission d'un code binaire :

Tableau : B (série des chiffres du code binaire)

Variables : d (intervalle d'envoi d'un bit)

Début_algo

 Saisir B

$n \leftarrow$ longueur de B

 Saisir d

 Pour i allant de 1 à n faire

 Si $B[i] = 1$ alors faire

 Allumer la DEL

 Attendre d/2 secondes

 Éteindre la DEL

 Sinon faire

 Éteindre la DEL

 Attendre d/2 secondes

 Allumer la DEL

 Fin_si

 Attendre d/2 secondes

Fin_pour

Fin_algo

Algorithme pour la transmission d'un message (« Envoyer » fait référence à la fonction définie par l'algorithme ci-dessus) :

Remarque : Tout le processus de conversion doit être terminé avant le début de la transmission du message pour éviter tout délai qui peut nuire au minutage.

Tableaux : B, D (commande « commencer »), L, M

Variables : m (message à transmettre), n, i

Début_algo

$D \leftarrow [0, 1, 0, 1]$

 Saisir m

$n \leftarrow$ nombre de caractères dans m

$M \leftarrow$ liste de caractères dans m

$B \leftarrow [] * n$

 Pour i allant de 1 à n faire

$B[i] \leftarrow$ liste des chiffres du code binaire de $M[i]$

 Fin_pour

$L \leftarrow$ code binaire de n

 Envoyer D

 Envoyer L

 Pour i allant de 1 à n faire

 Envoyer $B[i]$

 Fin_pour

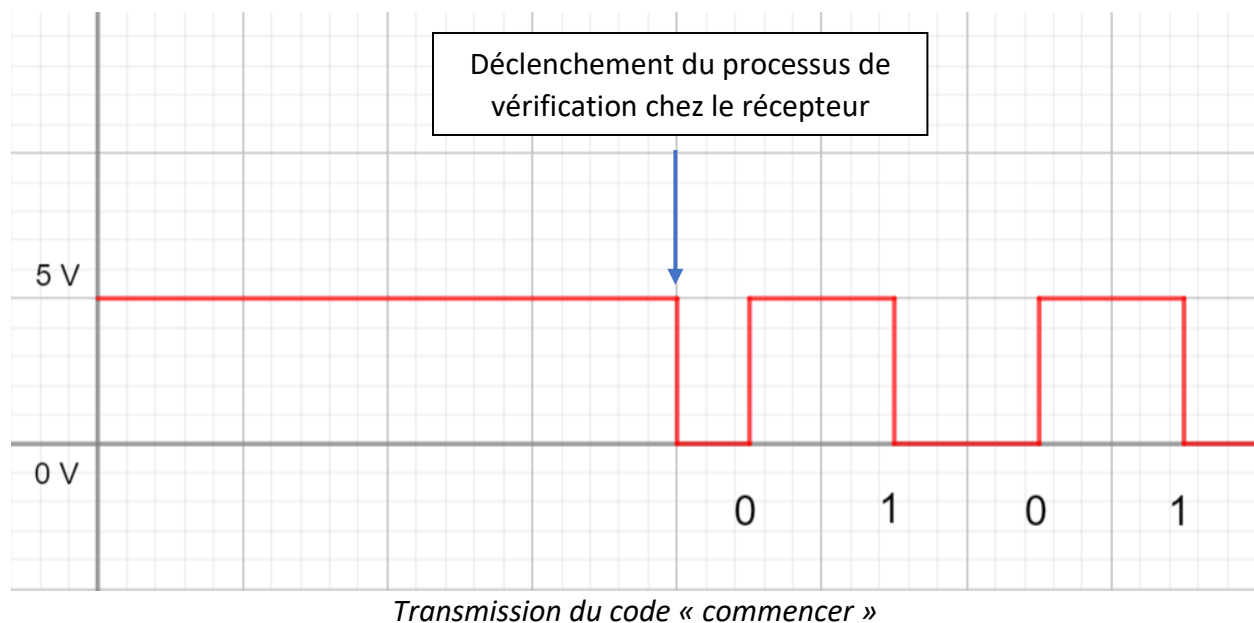
Fin_algo

4) Réception du code binaire

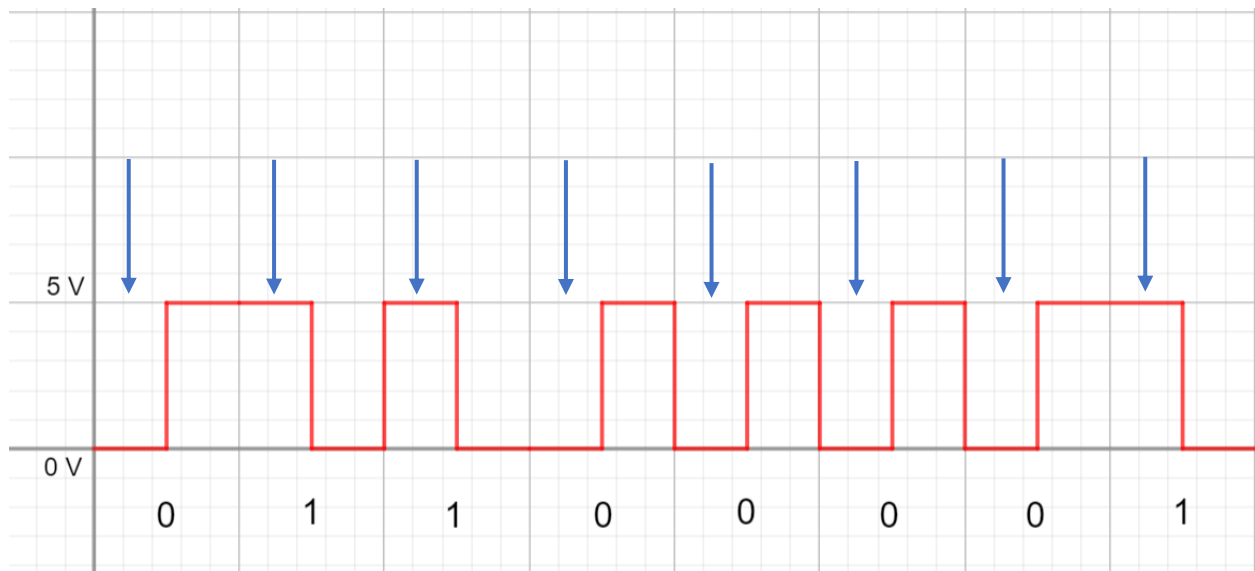
La réception du code binaire est plus délicate que la transmission. Il nécessite une gestion de temps correcte afin de pouvoir synchroniser avec le transmetteur et une prise en compte des erreurs de chronométrage possibles au niveau du microcontrôleur.

Le premier problème à adresser est celui de la lumière ambiante : dans une salle bien éclairée, il est possible que le panneau solaire génère un courant électrique sans que la DEL du transmetteur ne soit allumée. Pour éliminer ce risque, il est possible de faire de telle sorte que la tension générée par le panneau solaire doit dépasser un certain seuil pour être reconnu comme un signal « 1 ». C'est là que se sert le potentiomètre VR2 vu dans la partie 3.5. Composants et circuits : il peut être réglé par l'utilisateur afin de « calibrer » le seuil du panneau solaire à un niveau approprié pour la luminosité actuelle. Ainsi, à chaque lecture, le récepteur va devoir comparer la tension du panneau solaire à la valeur définie par le potentiomètre VR2.

Il faut maintenant mettre en place un mécanisme de vérification pour la commande « commencer ». Nous avons aussi établi précédemment que la DEL du transmetteur sera allumée lorsqu'il n'y a pas de message à envoyer. Donc, ce processus de vérification sera déclenché par une baisse de luminosité en-dessous du seuil défini par la calibration qui correspondra au premier « 0 » de la commande « commencer ». Le récepteur va ensuite tenter de « lire » la commande pendant $T \times 4$ secondes puis comparer le résultat obtenu à la commande stockée dans sa mémoire. Si le résultat est positif, la lecture va se déclencher. Sinon, il retournera en état d'attente. Cela permet au récepteur d'ignorer les perturbations temporaires.



Passons à la lecture. Idéalement, le récepteur doit pouvoir identifier le bit en lisant au début. Si la DEL est allumée, c'est un « 1 », sinon c'est un « 0 ». Mais, comme dit précédemment, les microcontrôleurs sont parfois susceptibles aux décalages temporaires qui peuvent fausser les lectures. Une solution est d'attendre $T/4$ secondes avant de lire pour être pile au milieu du demi-bit, permettant une marge d'erreur un peu avant ou après.



Transmission du code 01100001. Les flèches marquent les moments de lecture par le récepteur.

Les clignotements fréquents de la DEL avec le code Manchester permet aussi la correction du « chronomètre » intérieur du microcontrôleur après chaque lecture en attendant le changement d'état de la DEL au milieu du bit puis remettre le chronomètre à zéro et attendre $\frac{T}{4} \times 3$ secondes avant la prochaine lecture. Cette correction est essentielle lors des transmissions à grande vitesse en permettant une réduction du taux d'erreurs dû au décalage temporaire dans la transmission.

Algorithme pour la lecture d'une série de 8 chiffres (à la suite d'une lecture précédente) :

Tableau : T

Variables : t (intervalle entre chaque bit), c (chronomètre, qui va augmenter automatiquement avec le temps), i, p, s

Début_algo

```

T ← []
Pour i allant de 1 à 8 faire
    Tant que c < t/4*3
        Attendre
    Fin_tantque
    p ← tension du potentiomètre
    s ← tension du panneau solaire
    Si p > s alors faire
        T[i] ← 0
        Tant que p > s faire
            Attendre
        Fin_tantque
        c ← 0
    Sinon faire
        T[i] ← 1
        Tant que p <= s faire

```

```

| | | | Attendre
| | | | Fin_tantque
| | | | c ← 0
| | | | Fin_si
| | | | Fin_pour
| | | | Fin_algo

```

Algorithme pour la lecture d'un message (« Lire » fait référence à l'algorithme décrit ci-dessus) :

```

Tableau : T, D, M
Variables : p, s, t (intervalle entre chaque bit), n
Début_algo
| D ← [0, 1, 0, 1]
| p ← tension du potentiomètre
| s ← tension du panneau solaire
| Si s < p alors faire
| | Attendre t/4
| | T ← lire série binaire
| | Si D = T alors faire
| | | T ← lire série binaire
| | | n ← T en base 10
| | | M ← [] * n
| | | Pour i allant de 1 à n faire
| | | | T ← lire série binaire
| | | | M[i] ← T en base 10
| | | Fin_pour
| | | Pour i allant de 1 à n faire
| | | | Afficher caractère du code ASCII M[i]
| | | Fin_pour
| | Fin_si
| Fin_si
Fin_algo

```

Remarque : les codes en C seront en annexe.

7) Tests et analyse des résultats

Remarque : Les images du prototype et des circuits finals sont en annexe.

Après la réalisation du prototype et sa vérification, nous avons soudé les circuits imprimés finals et les avons fixés sur des plaques de PVC.

Dans les circuits imprimés, il y a quelques différences par rapport au schéma de la partie 2.5. Composants et circuits pour des raisons pratiques (distance physique entre les ports, placement des composants...). Ces changements sont reflétés dans le code en annexe.

Changements chez le récepteur :

- Les ports connectés à l'écran LCD sont 2, 4, 5, 6, 7 et 8 au lieu de 3, 4, 5, 6, 7 et 8.
- Le panneau solaire est connecté au port A1.
- Le potentiomètre VR2 est connecté au port A5.
- Le bouton de changement d'état est connecté au port 9.
- Par manque d'espace, la DEL D1 n'est pas présente dans le circuit final. Cela n'affecte pas le fonctionnement du récepteur, cette DEL ne servant qu'un rôle de débogage.

Changements chez le transmetteur :

- Le circuit imprimé utilisé contient déjà des composants soudés d'un circuit précédent, d'où la présence de câbles de liaison (voir photo dans l'annexe).
- La DEL est connectée au port 7.
- Les batteries ne sont pas utilisées, à cause du mauvais fonctionnement du transistor (peut-être dû au processus de soudure).

Une fois les codes téléchargés sur les microcontrôleurs, nous avons pu transmettre avec succès des messages. Mais, la machine ne peut aller au-delà d'une vitesse de transmission de 200 bits par seconde (le récepteur donne des erreurs, impossibilité de lire le message reçu), soit, avec les délais décrits dans l'algorithme, un temps de 45 ms par caractère (délai de $T = 5$ ms entre chaque bit) avec une fréquence de clignotement moyenne de 400 Hz. Néanmoins, nous pouvons toujours distinguer les clignotements de la DEL du côté du transmetteur.

Remarque : les parties suivantes sont purement hypothétiques et les analyses et hypothèses n'ont pas été vérifiées de manière fiable expérimentalement.

Pourquoi détecte-t-on toujours des clignotements à cette vitesse ?

Nous pouvons supposer que cela est surtout dû à l'irrégularité du clignotement de la DEL : tandis que les tests réalisés dans la partie précédente montre un seuil critique de fusionnement du papillotement d'environ 45 à 50 Hz, cela est avec une DEL qui clignote de manière régulière. Or, la DEL de la machine, pour pouvoir transmettre des messages, doit faire varier ces clignotements. Bien que ceci donne une fréquence moyenne de 400 Hz, les irrégularités semblent hausser de manière importante la fréquence critique de fusion.

De plus, la DEL utilisée émet une lumière blanche d'une luminosité maximale supérieure aux DEL utilisées dans les expériences précédentes, facteurs qui, d'après la partie 1.4. Le seuil critique de

fusionnement du papillotement, peuvent faire augmenter la fréquence nécessaire pour ne pas nuire à l'œil humain.

Avec le transmetteur, à une vitesse de transmission de $T = 2$ ms, soit une fréquence de clignotement de 1000 Hz, nous ne détectons plus de changement de luminosité, mais le récepteur est incapable de lire le message correctement à cette vitesse.

Pourquoi ne peut-on pas aller au-delà de 200 bits par seconde ?

D'après les tests, le récepteur donne toujours des erreurs au-delà de cette vitesse, même avec une hausse de seulement 50 bits par seconde.

La taille du circuit imprimé (5 x 7 cm) indique que les délais de transmission potentiellement causé par les fils électriques sont négligeables, tandis que le transmetteur fonctionne déjà de manière optimale, la conversion du message à envoyer se faisant avant la transmission plutôt que durant. La DEL du transmetteur est aussi peu probable de causer des problèmes, étant donné que les DEL commerciales peuvent être, d'après le professeur Harald Haas (voir partie Introduction), utilisé pour la Li-Fi.

Reste donc le récepteur. D'après le site de référence Arduino, la fonction *analogInput* que nous avons utilisée pour la lecture de la charge du panneau solaire a un délai de 100 μ s (0,1 ms), mais cela seul n'explique pas la limite de 200 b/s.

Nous avons donc émis l'hypothèse que le panneau solaire est responsable du délai qui rend impossible l'augmentation de la vitesse de transmission. Or, il est impossible de vérifier cela de manière entièrement conclusive, puisque le panneau solaire est déjà fixé sur la machine, ce qui ne permet pas de vérifier son fonctionnement avec un oscilloscope.

Donc, pour tester le temps de réponse du récepteur en général, nous avons utilisé les codes suivants :

```
#include <elapsedMillis.h>

const int led = 7;

void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  delay(10000);
  digitalWrite(led, HIGH);
  delayMicroseconds(100);
  digitalWrite(led, LOW);
}
```

Le code ci-dessus est pour le transmetteur : toutes les 10 secondes, la DEL va s'allumer pour 100 μ s (le temps minimal de lecture avec *analogInput* d'après le site de référence Arduino).

```

#include <elapsedMillis.h>
#include <LiquidCrystal.h>

LiquidCrystal lcd(2, 4, 5, 6, 7, 8);

const int sensor = A1;
elapsedMicros chrono;
int temps;

void setup() {
  pinMode(sensor, INPUT);
  lcd.write("Hello");
}

void loop() {
  if(analogRead(sensor) > 5){
    chrono = 0;
    while(analogRead(sensor) > 0){}
    temps = chrono;
    lcd.clear();
    lcd.print(temps);
  }
}

```

Ci-dessus est le code du récepteur : il démarre le chronomètre lorsque que la lumière venant de la DEL est détectée et affiche le temps d'éclairage perçu.

Avec un seuil de luminosité de zéro, nous avons eu un temps d'éclairage de 2,4 ms. Or, nous savons que la DEL est seulement allumée pour 100 μ s. Il y a eu donc un délai de 2,3 ms. Si le panneau solaire, dont la fonction principale est de produire de l'électricité à partir des rayons lumineux, a pu générer un courant électrique durant les 100 μ s, le délai serait probablement dû au temps nécessaire au panneau solaire pour se décharger et revenir à 0 V.

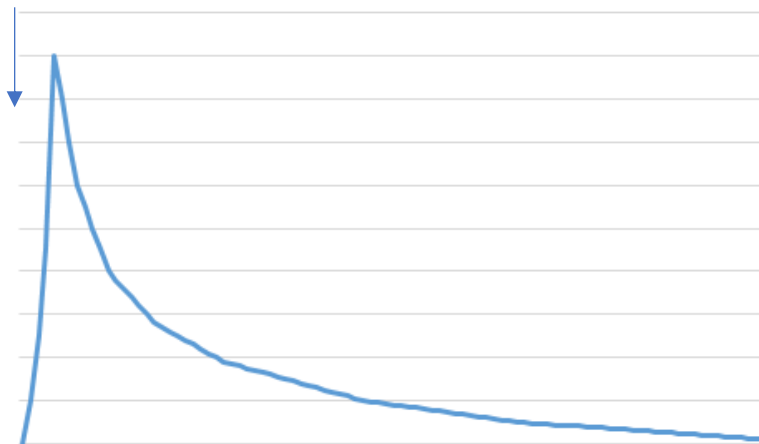


Schéma hypothétique du processus de charge et de décharge du panneau solaire (axe des ordonnées : charge électrique du panneau solaire ; axe des abscisses : temps). La flèche indique le moment d'allumage de la DEL.

Pour tester cela, nous avons augmenté le seuil au-delà de laquelle le panneau solaire « détecte » de la lumière venant de la DEL. En répétant l'expérience précédente, nous avons obtenu cette fois-ci un temps de réponse minimal de seulement 1 ms.

Nous pouvons donc assumer que le composant limitant chez le récepteur est le panneau solaire. Néanmoins, cette assumption ne peut se faire à 100 % de certitude, étant donné qu'elle a été réalisée avec un récepteur déjà monté (d'autres facteurs pouvant très bien intervenir) et non un panneau solaire détaché avec un oscilloscope.

Que faire pour surmonter cet obstacle ?

Il y a quelques voies possibles pour potentiellement améliorer le récepteur/transmetteur :

- Des composants plus performants, donc notamment le panneau solaire, pour minimiser les délais dans la lecture et dépasser la barrière de 5 ms, ou encore un microcontrôleur plus puissant qui permettra de minimiser les décalages de chronométrage.
- Un algorithme meilleur, qui permettrait de rapprocher la vitesse de transmission au temps de réponse minimal du récepteur. Un tel algorithme pourrait utiliser, par exemple, un système de codage et/ou de transmission plus efficace.

Alternativement, en prenant en compte les facteurs ayant une influence sur la fréquence critique de fusion de la DEL, nous pouvons :

- Diminuer l'intensité maximale de la DEL par moyen d'une résistance ou d'un potentiomètre, au détriment de la distance maximale de transmission.
- Réaliser la transmission dans une pièce sombre.

Cependant, ces dernières propositions sont seulement des astuces et ne constituent pas de réelles solutions.

Conclusion

Rappelons la problématique établie au début : *Comment transmettre un message à distance avec de la lumière visible sans gêner à l'œil humain ?*

Au cours de ce projet, nous avons pu établir les étapes nécessaires à une telle transmission :

1. Conversion du message en une forme communicable par une DEL, donc en code binaire. La conversion s'est faite par moyen du code ASCII qui attribue un nombre à chaque caractère du message.
2. Programmation de la DEL afin de la rendre capable de transmettre le message codé. La DEL doit clignoter pour envoyer le message en code binaire. Ce clignotement doit se faire à une fréquence au-delà du seuil critique de fusionnement du papillotement, à laquelle l'œil humain va percevoir une lumière continue. La fréquence du clignotement de la DEL doit être la plus élevée possible, étant donné les divers facteurs et les différences entre individus qui peuvent influencer ce seuil.
3. Réception du message par le récepteur. Dans cette étape, la synchronisation entre transmetteur et récepteur est indispensable et il faut prendre en compte les erreurs de chronométrage et les délais possibles pour assurer le transfert du message.
4. Décodage du message en une forme lisible.

Cependant, dans les appareils réalisés, nous n'avons pas pu assurer en même temps un clignotement de la DEL invisible à l'œil humain et une transmission correcte des messages. Nous avons pu identifier le temps de réponse du récepteur comme l'élément limitant dans cette entreprise.

Bien que nous n'ayons pas réussi à reproduire une transmission parfaite avec notre modélisation, nous pouvons conclure que ce serait parfaitement possible avec des composants plus performants, notamment chez le récepteur et le panneau solaire utilisé comme capteur.

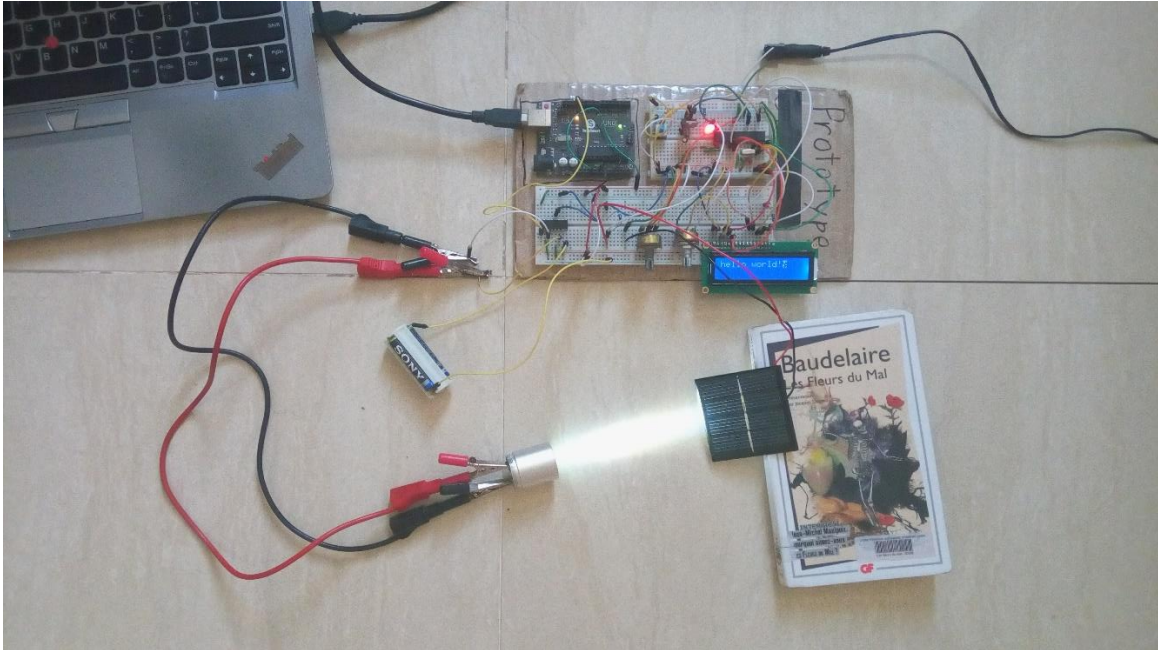
Bien sûr, la méthode de transmission que nous avons utilisée n'est pas la seule possible. Par exemple, de manière de la télécommande étudiée dans la partie 1.2. Étude : Fonctionnement d'une télécommande infrarouge, la transmission du code binaire peut se faire en variant la durée d'éclairage de la DEL pour différencier les bits. Il existe donc peut-être une méthode de transmission plus efficace de laquelle nous n'avons pas pensé qui permettrait de remplir les deux objectifs mentionnés avec les mêmes composants.

Dans le cas d'une poursuite de ce projet, une étude de ces possibilités serait intéressante.

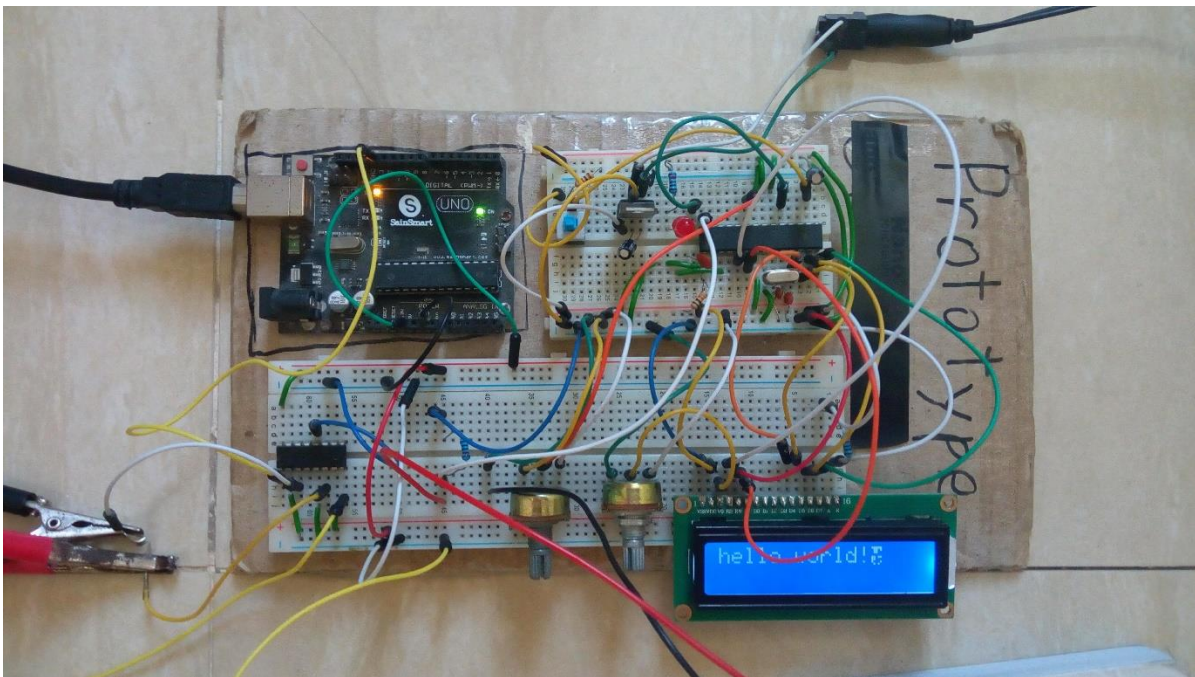
ANNEXE

Photos

Prototype :

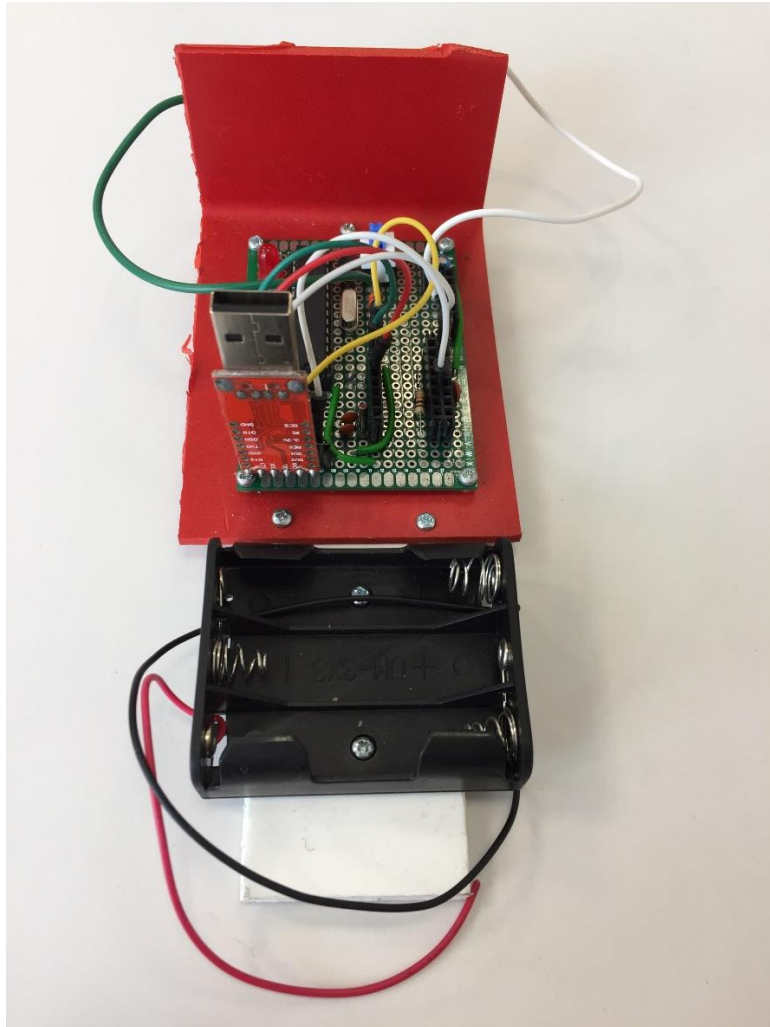


Vue d'ensemble

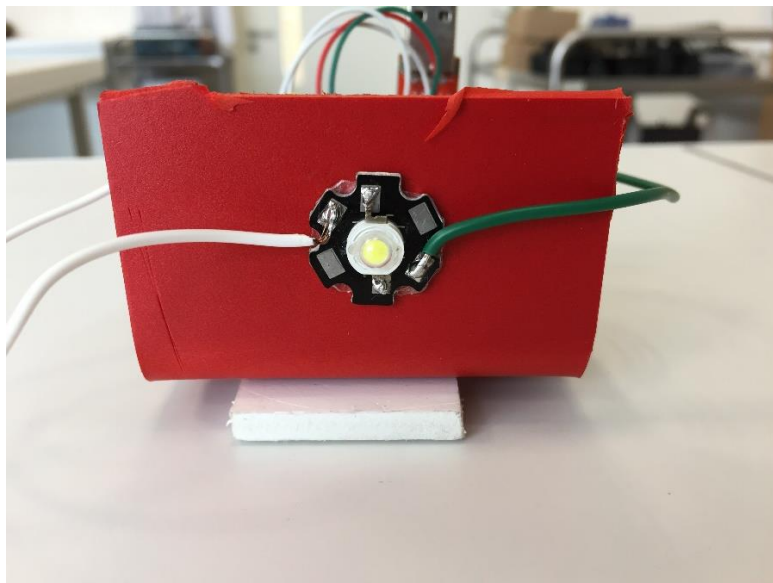


Cartes de prototypage

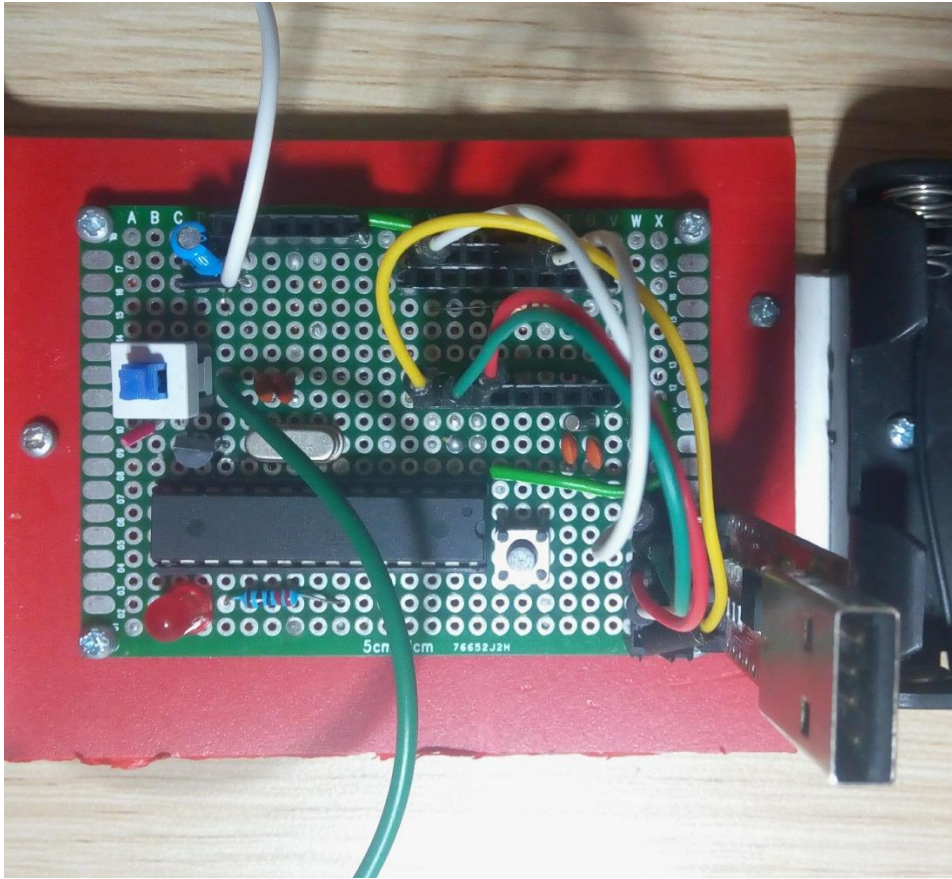
Transmetteur :



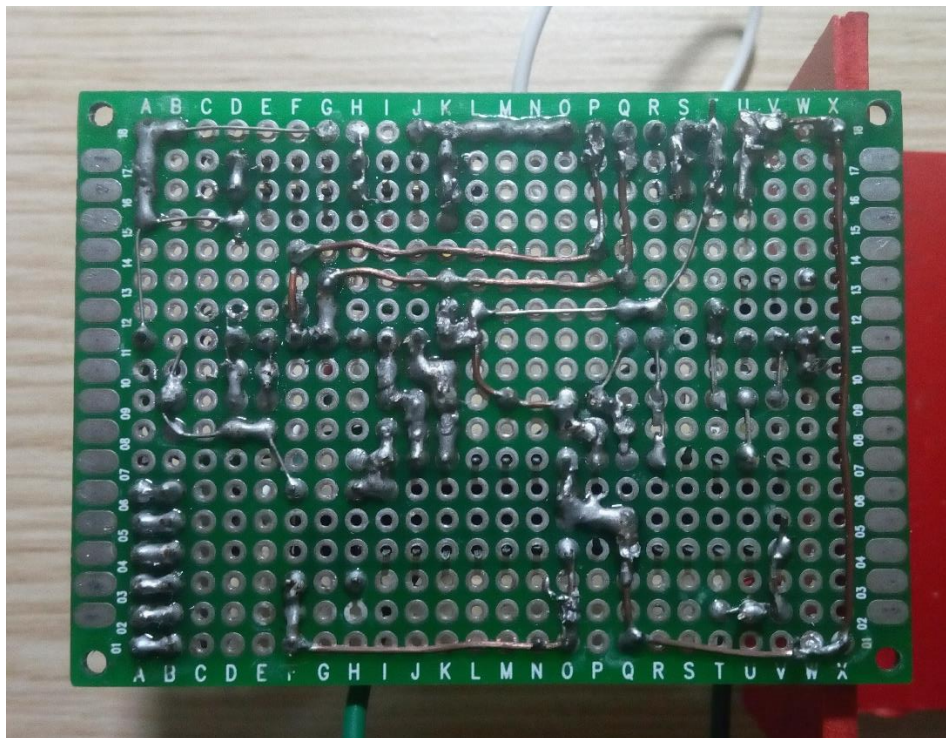
De derrière



De face

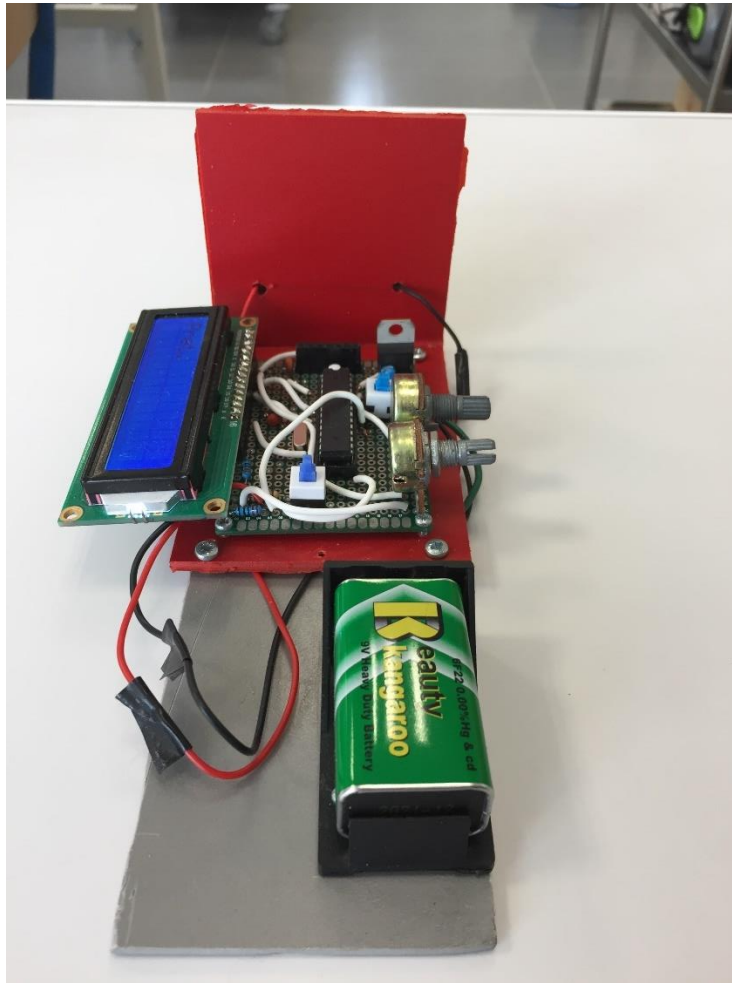


Circuit imprimé, recto

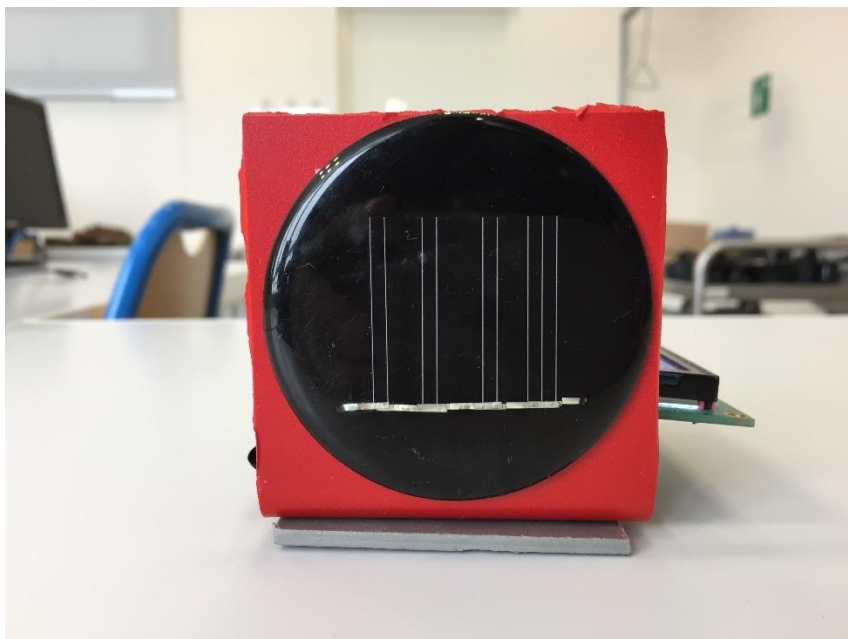


Circuit imprimé, verso

Récepteur :



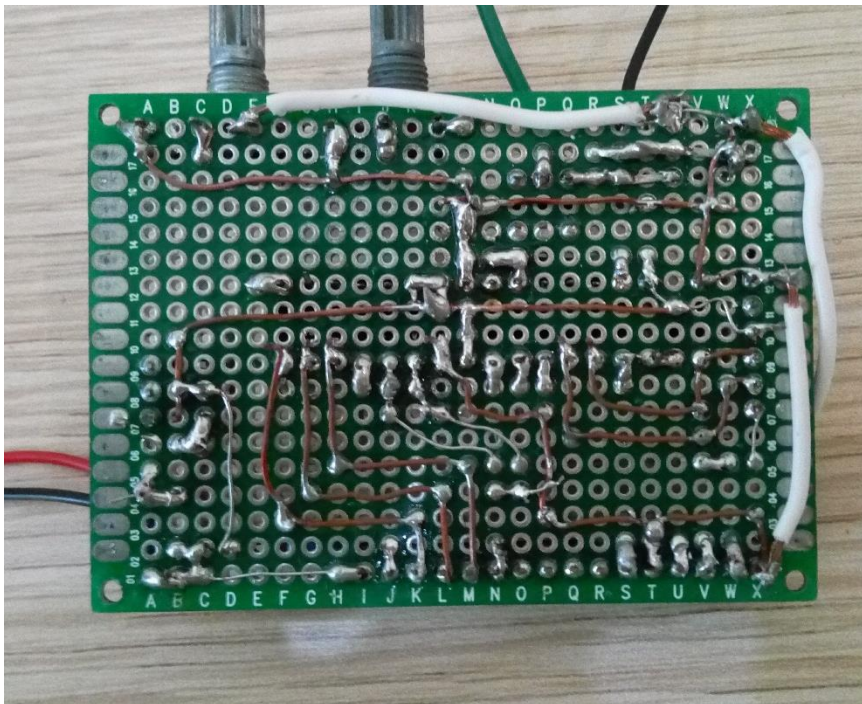
De derrière



De face



Circuit imprimé avec l'écran, recto



Circuit imprimé, verso

Code

Transmetteur :

```
/*
Code pour transmetteur Li-Fi
Par DO Nhat Minh et LE Nguyen Phong
Projet TPE 2018 - 2019
*/

#include <elapsedMillis.h>

#define INTERVAL 5000 // Temps T (intervalle entre chaque bit) en µs

const int led = 7; // Port de la DEL

const int attente = INTERVAL / 2;
elapsedMicros chrono; // Définition du chronomètre

String message = "";
int val[8] = {0, 0, 0, 0, 0, 0, 0, 0}; // Stockage temporaire de la série
binaire à envoyer

const int startLen = 4;
int start[startLen] = {0, 1, 0, 1}; // Commande start

void setup() {
    pinMode(led, OUTPUT);
    digitalWrite(led, HIGH);
    Serial.begin(9600);
}

void loop() {
    if(Serial.available() > 0){
        message = Serial.readString();
        int longueur = message.length();
        longueur--;
        if(longueur > 255){ // Ne pas envoyer si le message excède 255
caractères
            Serial.println("Message trop long");
        }
    }
}
```

```

    }
    else{
        int msgBin[longueur][8];
        for(int i = 0; i < longueur; i++){ // Conversion du message en code
binaire
            deccon(message.charAt(i) + 0);
            for(int a = 0; a < 8; a++){
                msgBin[i][a] = val[a];
            }
        }
        deccon(longueur);
        /* Code pour débogage, affichage des séries binaires sur ordinateur
        for(int i = 0; i < startLen; i++){Serial.print(start[i]);};
        Serial.println(" ");
        for(int i = 0; i < 8; i++){Serial.print(val[i]);};
        Serial.println(" ");
        for(int i = 0; i < longueur; i++){
            for(int a = 0; a < 8; a++){Serial.print(msgBin[i][a]);};
            Serial.println(" ");
        }
        */
        chrono = 0; // Remettre le chronomètre à zéro, pour commencer la
transmission
        sendChr(start, startLen); // Envoyer la commande start
        sendChr(val, 8); // Envoyer la longueur du message
        for(int i = 0; i < longueur; i++){
            sendChr(msgBin[i], 8); // Envoyer chaque caractère
        }
        Serial.println("Envoyé."); // Notifier l'utilisateur une fois le
message envoyé
        while(chrono < attente){}
        chrono = 0;
        digitalWrite(led, HIGH);
    }
}

void deccon(int x){ // Conversion en code binaire
    for(int i = 7; i > - 1; i--){

```

```

    val[i] = x%2;
    x = x/2;
}
}

void sendChr(int msg[], int len){ // Envoi de la série binaire val
    for(int i = 0; i < len; i++){

        while(chrono < attente){};
        chrono = 0;

        if(msg[i] == 0){
            digitalWrite(led, LOW);
            while(chrono < attente){};
            chrono = 0;
            digitalWrite(led, HIGH);
        }
        else if(msg[i] == 1){
            digitalWrite(led, HIGH);
            while(chrono < attente){};
            chrono = 0;
            digitalWrite(led, LOW);
        }
    }
    while(chrono < attente){};
    chrono = 0;
    digitalWrite(led, HIGH);
    delayMicroseconds(attente);
    digitalWrite(led, LOW);
    chrono = 0;
}

```

Récepteur :

```

/*
Code pour récepteur Li-Fi
Par DO Nhat Minh et LE Nguyen Phong

```



```

Projet TPE 2018 - 2019
*/

#include <elapsedMillis.h>
#include <LiquidCrystal.h>

#define INTERVAL 5000 // Temps T (intervalle entre chaque bit) en µs
#define TIME_OUT 2500 // La durée time-out pour vérification

const int attente = INTERVAL / 4 * 3;

LiquidCrystal lcd(2, 4, 5, 6, 7, 8); // Définir l'écran LCD

const int sensor = A1; // Port du panneau solaire
const int pot = A5; // Port du potentiomètre VR2

// Variables pour le mode calibration
const int cal = 9;
int calButton = 2;
int potVal;
int calPotVal;
int sensorVal;
int calSensorVal;

const int startLen = 4;
int start[startLen] = {0, 1, 0, 1}; // Commande "start"
int val[8] = {0, 0, 0, 0, 0, 0, 0, 0}; // Tableau pour stockage temporaire
des bits lors d'une lecture

elapsedMicros chrono; // Définir le chronomètre

bool error = false; // Variable permettant de détecter les erreurs de
lecture

void setup() {
    lcd.begin(16, 2);
    pinMode(sensor, INPUT);
    pinMode(pot, INPUT);
    pinMode(cal, INPUT);

```

```

}

void loop() {
    if(digitalRead(cal) == LOW){ // mode attente du message
        if(calButton != 0){
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Pret.");
            calButton = 0;
        }
        bool startRec = false;
        sensorVal = analogRead(sensor);
        if(sensorVal <= potVal){
            chrono = 0;
            delayMicroseconds(TIME_OUT);
            chrono = 0;
            val[0] = 0;
            if(readChar(1, 4) == true){ // Vérification du commande start
                for(int i = 0; i < 4; i++){
                    if(val[i] != start[i]){
                        break;
                    }
                    if(i == 3){
                        startRec = true;
                        lcd.clear();
                        lcd.setCursor(0, 0);
                        lcd.print("Lecture...");
                    }
                }
            }

            if(startRec == true){ // Si commande start valide, commencer la
lecture

                if(readChar(0, 8) == true){ // Lire la longueur du message
                    int msgLen = decon(8);

                    lcd.setCursor(0, 1);
                    lcd.print(msgLen);
                }
            }
        }
    }
}

```

```

int msgBin[msgLen][8];

// Lecture du message
for(int i = 0; i < msgLen; i++){
    if(readChar(0, 8) == true){
        for(int a = 0; a < 8; a++){
            msgBin[i][a] = val[a];
        }
    }
    else{
        error = true;
        break;
    }
}

if(error == false){

    // Affichage du message sur l'écran LCD

    char msg[msgLen];
    for(int i = 0; i < msgLen; i++){
        for(int a = 0; a < 8; a++){
            val[a] = msgBin[i][a];
        }
        msg[i] = deccon(8);
    }
    lcd.clear();
    for(int i = 0; i < msgLen; i++){
        lcd.write(msg[i]);
        if(i == 15){
            lcd.setCursor(0, 1);
        }
    }
    startRec = false;
}
else{
    error = true;

```

```

    }
}

// En cas d'erreur, afficher "Erreur !" sur l'écran LCD

if(error == true){
    lcd.clear();
    lcd.print("Erreur !");
    error == false;
}
}

else{ // Mode calibration qui permet le réglage du seuil de tension du
panneau solaire
    if(calButton != 1){
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Pot: ");
        lcd.print(analogRead(pot));
        lcd.setCursor(0, 1);
        lcd.print("Sensor: ");
        lcd.print(analogRead(sensor));
        calButton = 1;
    }
    potVal = analogRead(pot);
    sensorVal = analogRead(sensor);
    if(chrono > 500000){
        if(potVal != calPotVal){
            calPotVal = potVal;
            lcd.setCursor(5, 0);
            lcd.print("          ");
            lcd.setCursor(5, 0);
            lcd.print(potVal);
        }
        if(sensorVal != calSensorVal){
            calSensorVal = sensorVal;
            lcd.setCursor(8, 1);
            lcd.print("          ");

```

```

        lcd.setCursor(8, 1);
        lcd.print(sensorVal);
    }
    chrono = 0;
}
}
}

bool readChar(int deb, int len){ // Lecture d'un caractère (série de 8 bits)

    bool notError = true;

    if(deb == 0){ // Attendre la fin de la pause si au début de la série
        while(chrono < INTERVAL){}
        chrono = 0;
    }

    for(int i = deb; i < len; i++){

        while(chrono < attente){}; // Attendre le moment de lecture
        chrono = 0;

        if(analogRead(sensor) <= potVal){

            val[i] = 0;

            while(analogRead(sensor) <= potVal && chrono < TIME_OUT){}
            if(chrono < TIME_OUT){
                chrono = 0; // Remise du chronomètre à zéro; correction des
délages temporels
            }
            else{
                notError = false;
                break;
            }
        }
        else{

            val[i] = 1;

```

```

        while(analogRead(sensor) > potVal && chrono < TIME_OUT){}
        if(chrono < TIME_OUT){
            chrono = 0; // Même chose
        }
        else{
            notError = false;
            break;
        }
    }
}
return notError; // Retourner l'état d'erreur
}

int deccon(int len){ // Conversion de la série binaire val en nombre
décimal
    int num = 0;
    for(int i = 0; i < 8; i++){
        num += val[7 - i] * bit(i); // La fonction bit(n) retourne 2 puissance
n
    }
    return num;
}

```

Bibliographie et sitographie

I. Introduction

Sites consultés :

- Harald Haas introduit et démontre la Li-Fi (vidéo) :
https://www.ted.com/talks/harald_haas_a_breakthrough_new_kind_of_wireless_inter_net
- Article sur le photophone : <https://www.thoughtco.com/alexander-graham-bells-photophone-1992318>
- Informations sur Alexander Graham Bell :
https://en.wikipedia.org/wiki/Alexander_Graham_Bell

- Introduction à la Li-Fi : <https://www.linkedin.com/pulse/what-lifi-technology-how-does-work-beginners-guide-akash>

II. La persistance rétinienne et la lumière

Sites consultés :

- La persistance rétinienne dans l'animation : <https://www.futurelearn.com/courses/explore-animation/0/steps/12222>
- Informations sur les cônes : [https://fr.wikipedia.org/wiki/C%C3%B4ne_\(photor%C3%A9cepteur\)](https://fr.wikipedia.org/wiki/C%C3%B4ne_(photor%C3%A9cepteur))
- Informations sur les bâtonnets : <https://fr.wikipedia.org/wiki/B%C3%A2tonnet>
- Fonctionnement de la rétine : <https://www.britannica.com/science/human-eye/The-work-of-the-retina#ref64944>
- La loi Ferry-Porter : <http://www.oxfordreference.com/view/10.1093/oi/authority.20110803095815649>
- Modulation de largeur d'impulsion : <https://www.arduino.cc/en/tutorial/PWM>

Documents consultés :

- F. Cail, S. Salsi. La fatigue visuelle.. [Rapport de recherche] Notes scientifiques et techniques de l'INRS NS 92, Institut National de Recherche et de Sécurité (INRS). 1992, 58 p., ill., bibliogr. <https://hal-lara.archives-ouvertes.fr/hal-01420177/document>
- Galifret Yves, Piéron Henri. Étude des fréquences critiques de fusion pour des stimulations chromatiques intermittentes à brillance constante. Dans : L'année psychologique. 1944 vol. 45-46. pp. 1-15
- Roberts, J., & Wilkins, A. (2013). Flicker can be perceived during saccades at frequencies in excess of 1 kHz. *Lighting Research & Technology*, 45(1), 124-132. <https://doi.org/10.1177/1477153512436367>
- Lehman B, Wilkins AJ. "Designing to Mitigate Effects of Flicker in LED Lighting: Reducing risks to health and safety". *Power Electronics Magazine, IEEE*. **2014** (09): 18–26. [doi:10.1109/MPPEL.2014.2330442](https://doi.org/10.1109/MPPEL.2014.2330442).

Des informations ont aussi été prélevées des cours de SVT vus en classe.

III. Réalisation d'appareils de transmission et de réception par voie lumineuse

Sites consultés :

- Fonctionnement d'une télécommande : <https://electronics.howstuffworks.com/remote-control2.htm>
- Code ASCII : <https://en.wikipedia.org/wiki/ASCII>
- Conversion décimal-binaire : <https://blog.angularindepth.com/the-simple-math-behind-decimal-binary-conversion-algorithms-d30c967c9724>
- Introduction sur Arduino : <https://www.arduino.cc/en/Guide/Introduction>
- Tutoriel AVR (pour la comparaison avec la plateforme Arduino) : <http://www.ladyada.net/learn/avr/fuses.html>
- Code Manchester : https://en.wikipedia.org/wiki/Manchester_code
- Tutoriel Arduino : <https://fr.flossmanuals.net/arduino/historique-du-projet-arduino/>
- Arduino sur platine de prototypage : <https://www.arduino.cc/en/main/standalone>
- Fiche technique LM7805 : <https://www.sparkfun.com/datasheets/Components/LM7805.pdf>
- Utilisation de EAGLE : <https://learn.sparkfun.com/tutorials/using-eagle-schematic/all>
- Utilisation d'un écran LCD : <https://www.arduino.cc/en/Reference/LiquidCrystal>
- Utilisation d'un transistor : <https://learn.adafruit.com/adafruit-arduino-lesson-13-dc-motors/transistors>
- Référence pour le langage de programmation d'Arduino : <https://www.arduino.cc/reference/en/>
- La librairie « ElapsedMillis » (utilisé pour le minutage) : <https://playground.arduino.cc/Code/ElapsedMillis>