

**PROGRAMMATION PAR ADUINO
D'UN ANCIEN PANNEAU
D'AFFICHAGE SNCF**



Figure 1 : Image du panneau d'affichage

Etudiants :

Clément PAULY

Wassim YAAKOUBI

Lingyu LIU

Weihaio ZHOU

Enseignant-responsable du projet :

Clément KELLER

Cette page est laissée intentionnellement vierge.

Date de remise du rapport : 19/06/2017

Référence du projet : STPI/P6/2017 – 14

Intitulé du projet : **PROGRAMMATION PAR ADUINO D'UN ANCIEN PANNEAU D’AFFICHAGE SNCF**

Type de projet : **Expérimental, conception, simulation.**

Objectifs du projet (10 lignes maxi) :

Le but du projet est de réaliser un montage qui permettrait de piloter un ou plusieurs modules d'un panneau d'affichage SNCF pour y afficher des lettres.

En effet, à partir des modules, nous devons former des lettres et donc des mots. A partir de ces mots, on affichera sur le panneau l'ensemble des informations relatives aux voyages pour aider les voyageurs dans les gares SNCF.

Mots-clefs du projet (4 maxi) : **Arduino MEGA, panneau d'affichage , SNCF**

TABLE DES MATIERES

Sommaire

1. Introduction.....	6
2. Méthodologie / Organisation du travail	7
3. Travail réalisé et résultats	8
3.1. Travail préliminaire	8
3.1.1. Analyse du système.....	8
3.1.2. Recherche d'une solution : les composants.....	10
3.2. Conception	12
3.2.1. Schéma du montage et code informatique	12
3.2.2. Problèmes et phases de tests	17
3.3. Résultats et solution finale.....	17
3.3.1. Résultats attendus	17
3.3.2. Résultats obtenus	18
3.3.3. Résultat théorique.....	18
3.3.4. Autres pistes	18
4. Conclusion et perspective.....	19
5. Bibliographie.....	20
6. Table des illustrations	20
7. Annexes.....	21
7.1. Programme théorique final.....	21

NOTATIONS, ACRONYMES

- **Arduino, Arduino Méga** : Microcontrôleur programmable (le model « Méga » est le plus puissant).
- **CLK, Clock** : Signal horloge permettant la synchronisation de la carte Arduino avec le MAX7219.
- **Démultiplexeur, Driver LED** : Composant électronique servant à piloter une matrice LED.
- **DIN, DOUT** : Signal portant les informations des lignes et colonnes.
- **Fritzing** : Logiciel d'édition de schéma électronique.
- **LedMatrix** : librairie de fonction (langage C) servant à piloter le MAX.
- **LOAD** : Signal permettant de sélectionner un MAX.
- **Shield Web** : Extension Arduino permettant de générer un serveur web pour cette dernière.
- **Transistors Darlington** : Composants électroniques jouant le rôle d'interrupteur laissant passer le courant ou non en fonction du signe de l'intensité.
- **Amplificateurs Opérationnels** : Composant électronique servant à amplifier une tension donnée, un signal entrant.

1. INTRODUCTION

Dans le cadre de projets de P6 en semestre 4 STPI2, Mr Keller nous a donné un ancien panneau d'affichage SNCF. Tout au début, les seuls composants qu'on avait en disposition étaient un module du panneau, une platine d'expérimentation et des câbles. Mr Keller nous a aussi fourni un schéma électrique pour réaliser le montage de l'afficheur. Donc l'objectif primordial était de trouver des composants dont on a besoin dans ce projet. C'est ce qu'on a essayé de faire au début de ce semestre.

Le but principal de ce projet est de réaliser le schéma électrique et d'écrire un programme s'appuyant sur la Carte Arduino pour faire fonctionner l'afficheur en produisant une lettre ou un chiffre sur chaque module du panneau indépendamment.

Ce projet se divise en deux parties : partie électronique et partie informatique. C'est à dire, la réalisation du montage et du programme Arduino. On a quatre personnes dans ce projet, donc on sépare en deux binômes afin de réaliser ces deux parties du projet. À la fin de chaque séance, après qu'on ait travaillé chacun sur sa partie séparément, on les combine et on voit si ça fonctionne bien.

Pour bien réaliser ce projet, au début nous analysons le schéma électronique donné par Mr Keller, après avoir passé du temps à comprendre le fonctionnement du montage, on a déduit le bilan des composants nécessaires à la réalisation du projet. Ensuite, avec l'aide de Mr Keller, nous avons passé la commande des composants sur internet. Après, nous nous sommes séparés en deux binômes, un pour étudier le montage électronique et l'autre pour écrire le code permettant de piloter la Carte Arduino. Ensuite nous combinons les résultats trouvés et en discutons.

Enfin nous présenterons le résultat final, les solutions et modèles envisageables pour faire fonctionner l'afficheur.

2. METHODOLOGIE / ORGANISATION DU TRAVAIL

Lors de la première séance, Mr Keller nous a expliqué que ce même projet a été élaboré par des élèves STPI2 l'année dernière et que finalement ça n'avait abouti à rien. On a vite assimilé la complexité de la tâche et on s'est mis un but depuis le début c'est de faire mieux que les anciens élèves.

Dans un premier temps, on a essayé de comprendre le projet dans sa globalité : composants utilisés, mécaniques constatées... Ensuite une longue phase de documentation s'étant vite imposée. En cherchant sur différents sites Internet, on s'est renseigné sur le fonctionnement global de la carte Arduino : son régime de courant, sa tension d'entrée, de sortie et ses branchements...

Une phase de sélection et de conception d'une solution a été nécessaire ; n'étant pas un spécialiste en électronique, Mr Keller a fait appel à un collègue électronicien pour concevoir un dessin du montage à réaliser. Une fois ce schéma électronique fourni, on a cherché à le comprendre globalement puis on devait faire l'inventaire des composants à utiliser pour ensuite passer la commande des pièces grâce aux références indiquées sur le schéma.

La répartition des tâches n'a pas été aussi compliquée vu notre petit effectif. Ce petit groupe de 4 personnes nous a permis de travailler en binôme et ainsi de confronter nos idées.

Une fois les différents composants arrivés, Clément et Wassim avons commencé à réaliser le montage final tandis que Weihao et Lingyu se sont concentrés sur la partie codage pour assurer le fonctionnement de la carte Arduino Méga.

Nous avons travaillé essentiellement durant les séances de projet (à raison d'une heure et demie par semaine) car le matériel électronique n'était pas disponible pour un travail à la maison. Ayant scindé le groupe en 2, il était important que nous communiquions tout de même : nous nous tenions au courant des objectifs au début de chaque séance et voyions ensemble l'avancée du projet à chaque fin de séance, ce qui nous a permis de travailler en coordination et de ne pas perdre de temps lorsque les 2 parties du projet se sont réunies pour l'étape finale.

3. TRAVAIL REALISE ET RESULTATS

3.1. Travail préliminaire

3.1.1. Analyse du système

Lors de notre première séance de P6, Mr Keller nous a présenté le panneau d'affichage SNCF. Celui-ci est en effet composé de 12 modules : on peut produire une lettre ou un chiffre sur chaque module indépendamment ; la combinaison de ces modules permet d'afficher un mot, une phrase ou une série de chiffres. On s'est vite persuadé que la carte Arduino Méga nous permettait de piloter chacun de ses modules à l'aide d'un programme informatique.



Figure 2 : Intérieur du panneau d'affichage

Nous sommes rapidement arrivés à la conclusion que l'étude du fonctionnement d'un seul module permettrait d'élaborer un modèle pour faire fonctionner un groupe de modules. Chaque module est composé de 35 pastilles (7 Lignes * 5 colonnes) à deux côtés : un noir et un jaune. Différentes broches sont connectées à ces pastilles : une partie des broches commande la rotation des pastilles d'une colonne, tandis que l'autre partie commande celle d'une ligne.

En alimentant une broche de la partie ligne et une broche de la partie colonne, on commande ainsi la rotation d'une pastille en particulier. En inversant le branchement, on peut au contraire revenir à l'état de base, c'est-à-dire celui où les pastilles affichent un côté noir. Pour se faire, il faut mettre le pôle plus sur la broche reset et ensuite pointer sur les différentes colonnes à faire passer au noir.

Les broches sont en fait reliées à des bobines : une perturbation du champ électrique entraîne donc une rotation d'une pastille en particulier.

De plus, nous avons déterminé à l'aide d'un multimètre que la rotation d'une pastille ne se faisait que sous une tension minimum de 10V et une intensité de 300mA.

Pour ce fait, on a utilisé un générateur de tension continue capable de délivrer une tension de 12V et on a programmé la carte Arduino de façon qu'on ait une tension de 5V à sa sortie. Ensuite, nous devons trouver un moyen de contrôler chaque pastille indépendamment les unes des autres afin de pouvoir afficher la lettre désirée.

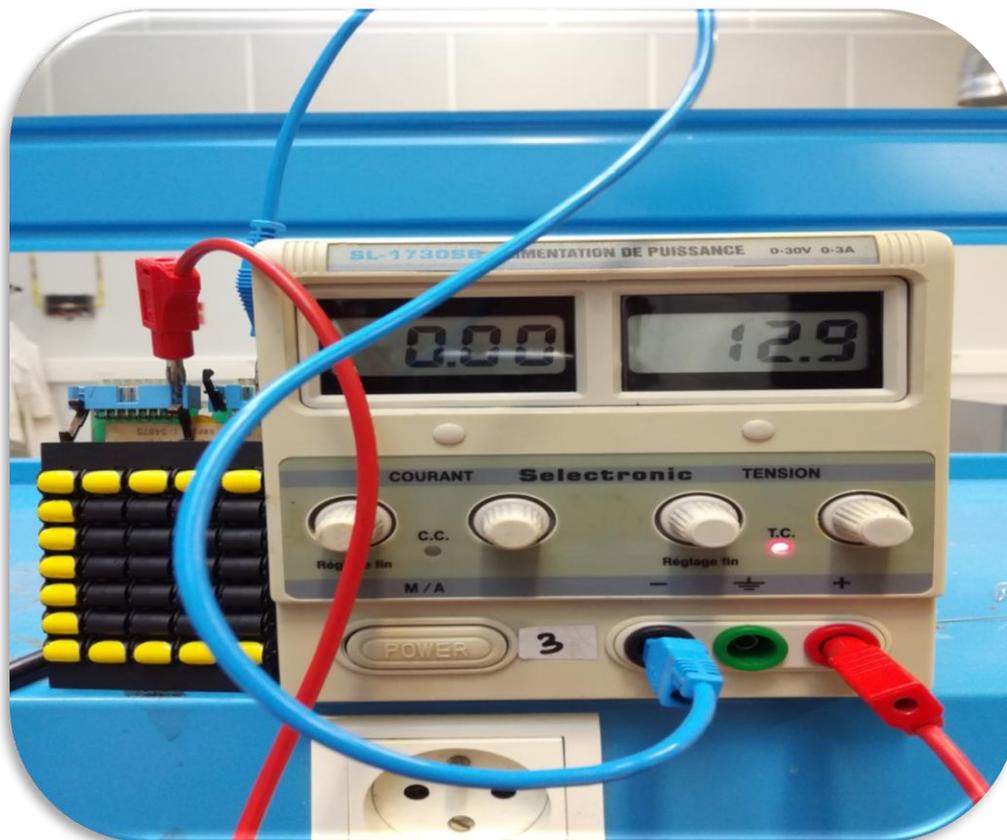


Figure 3 : Test de fonctionnement des modules

3.1.2. Recherche d'une solution : les composants



Figure 4 : Exemple d'affichage possible

Tout d'abord, on s'est beaucoup aidé des recherches effectuées par les anciens élèves STPI2 l'année dernière sur le panneau d'affichage. En effet, l'afficheur étudié était un modèle assez original, il date de 1994 et a été retrouvé dans une ancienne gare espagnole. De plus l'utilisation de ces afficheurs a été limitée, surtout avec l'apparition des nouveaux modèles plus performants et innovants à palettes, LED ou plus récemment les afficheurs légers capables de diffuser des informations visuelles et sonores pour les personnes malentendantes et malvoyantes.

L'étude du projet s'est vite basée sur l'utilisation de la carte *Arduino Méga 2560* dotée d'un microcontrôleur *ATMéga2560o* disposant d'une mémoire flash de 256 KB. Cette carte, accessible à prix abordables, dispose de 54 broches numériques d'entrées/sorties permettant ainsi de piloter plusieurs pastilles en même temps. Pour pouvoir utiliser la carte Arduino, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB et de naviguer sur un logiciel Arduino conçu pour ce besoin.

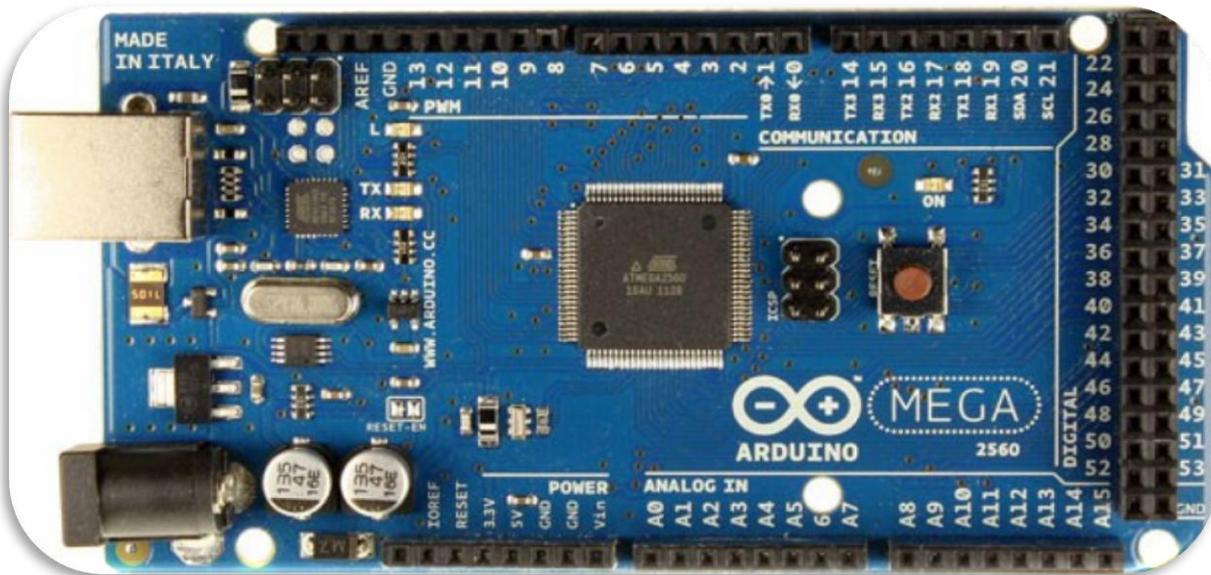


Figure 5 Carte Arduino Méga 2560

Il y a une infinité d'utilisations de la carte Arduino, on peut citer quelques applications possibles réalisées grâce à cette fameuse carte :

- Contrôler des appareils domestiques
- Donner une « intelligence » à un robot
- Réaliser des jeux de lumière
- Permettre à un ordinateur de communiquer avec une carte électronique et un capteur
- Télécommander un appareil mobile (Modélisme)
- Etc.

Pédagogiquement, Arduino a aussi pas mal d'atouts. En effet, ses créateurs ont d'abord pensé ce projet pour qu'il soit facile d'accès. Il permet ainsi une très bonne approche de nombreux domaines et ainsi d'apprendre plein de choses assez simplement.

Voici quelques exemples d'utilisation possible :

- Simuler le fonctionnement des portes logiques
- Mettre en œuvre et faciliter l'utilisation d'un réseau informatique
- Se servir d'Arduino pour créer des maquettes animées montrant le fonctionnement des collisions entre les plaques de la croûte terrestre, par exemple
- Donner un exemple concret d'utilisation des matrices avec un clavier alphanumérique 16 touches ou plus.
- Être la base pour des élèves ayant un TPE à faire pour le bac

3.2. Conception

3.2.1. Schéma du montage et code informatique

Les composants utilisés pour le fonctionnement du système sont :

Nom du composant	Rôle	Quantités
Carte Arduino MEGA	Gérer la transmission de l'information et alimentation 5V	1
Générateur	Alimentation 12V	1
Transistor NPN	Laisse passer le courant de la base à l'émetteur et bloque dans l'autre.	12
Transistor PNP	Laisse passer le courant de l'émetteur à la base et bloque dans l'autre.	7
Transistor NPN Darlington	Laisse passer le courant de la base à l'émetteur et bloque dans l'autre.	6
Transistor PNP Darlington	Laisse passer le courant de l'émetteur à la base et bloque dans l'autre.	5
Amplificateur opérationnel inverseur	Permet d'obtenir en sortie l'opposé de la tension d'entrée. ($U_s = -U_e$)	5
Résistance	Atténuer l'intensité du courant qui passe dans le circuit et protéger les transistors	1.2 KΩ : 20 220Ω : 5 200Ω : 5

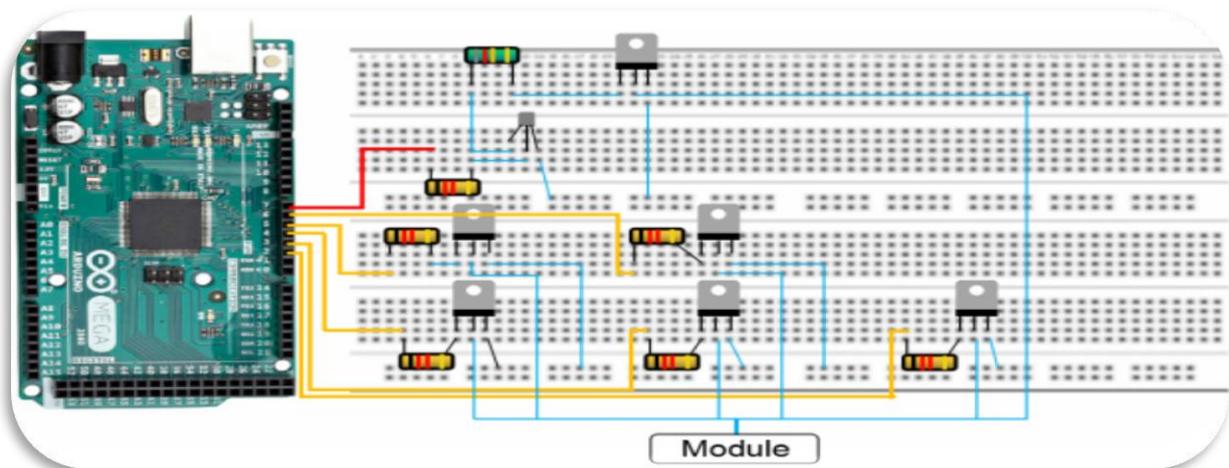


Figure 6 : Schéma du montage

Le schéma proposé par l'électronicien ne comportait que le câblage pour une ligne et une colonne ainsi que le RESET (Fonction permettant de faire passer les pastilles de la couleur jaune à la couleur initiale noire).

Le bilan du matériel est résumé dans le tableau ci-dessus. Pour une ligne, il faut 1 transistor NPN, 1 transistor PNP, 2 résistances, et ceci à multiplier par 7.

Pour une colonne, 1 transistor Darlington NPN, 1 transistor Darlington PNP, 1 transistor NPN, 3 résistances, un inverseur qu'on a ensuite remplacé par un amplificateur opérationnel inverseur et ceci à multiplier par 5.

Une étude approfondie de chaque composant isolé nous a parue indispensable pour comprendre le fonctionnement du montage.

Dans un premier lieu, on a étudié le principe et le comportement d'un transistor bipolaire :

- ✓ **Un transistor** est constitué de 2 jonctions PN (ou diodes) montées en sens inverse.
Selon le sens de montage de ces diodes on obtient 2 types de transistors :

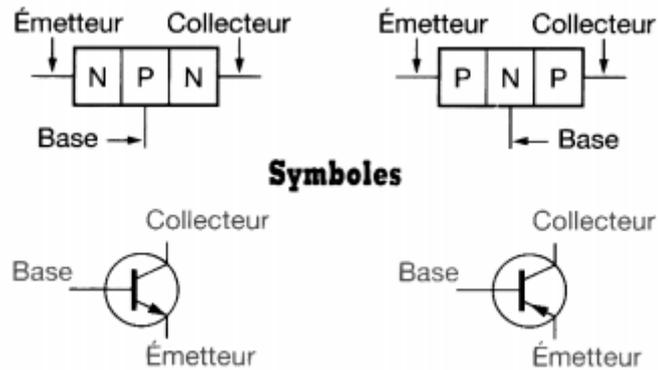


Figure 7 : Symbole d'un transistor bipolaire

<p>Un transistor comporte 3 connexions : L'émetteur (E) , La base (B) et le collecteur (C)</p>	
<p>Le transistor NPN</p>	<p>Le transistor PNP</p>
<p>La base, zone de type P est située entre deux zones de type N.</p>	<p>La base, zone de type N, est située entre deux zones de type P.</p>

- ✓ Un transistor Darlington est la combinaison de 2 transistors bipolaires de même type (tous les 2 NPN ou tous les 2 PNP)

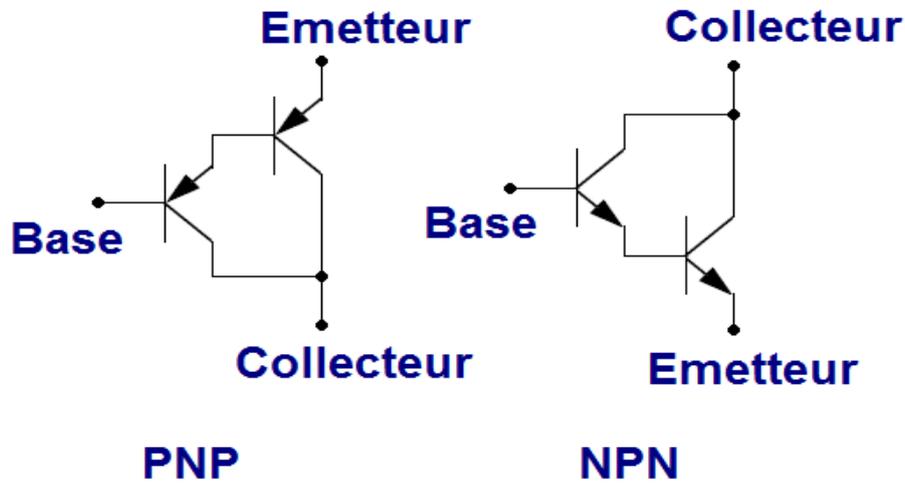


Figure 8 Transistor Darlington

Ensuite on a étudié le montage de l'amplificateur opérationnel inverseur :

- ✓ C'est un amplificateur électronique qui amplifie une différence de potentiel électrique présente à ses entrées. Il permet facilement d'implémenter facilement les opérations mathématiques de base comme l'addition, la soustraction, l'intégration, la dérivation et d'autres.

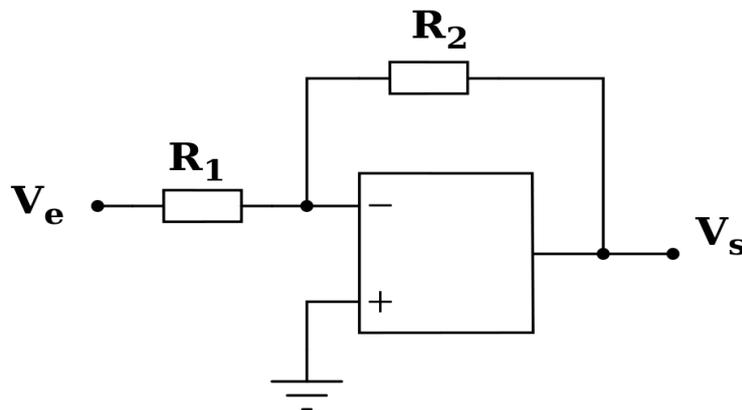


Figure 9 : Montage inverseur d'un amplificateur opérationnel

Equation de l'amplificateur opérationnel inverseur :

$$V_s = -V_e \times \left(\frac{R_2}{R_1}\right)$$

Dans notre cas, l'utilisation de 2 résistances R1 et R2 de même valeur (R1=R2) nous permettrait d'avoir un coefficient multiplicatif de -1 de façon que ça réponde à nos attentes pour inverser un potentiel d'entrée $V_e=5V$ en un potentiel de sortie $V_s=-5V$.

Mis à part le montage électrique, on devait étudier la phase de programmation qui est, elle aussi, aussi complexe.

Les contrôleurs Arduino sont programmables à l'aide de différents langages de programmation, comme le Java ou le C. Nous avons choisi ici d'utiliser le langage, plus accessible.

Tout d'abord pour tester notre programme, on avait pour but de faire tourner une seule ligne de pixels parmi les 7 lignes composant le module. Par conséquent, nous utilisons les broches 2-6 de la carte Arduino pour piloter les 5 colonnes et la broche 7 pour piloter la ligne.

Une tension minimale de 5V est nécessaire pour alimenter chaque broche, pour ce fait il a fallu compiler un code qui va avec la carte Arduino. Le code `pinMode(Pin, OUTPUT)` configure la broche spécifiée pour se comporter en entrée ou en sortie. Ici, pour alimenter les broches, on prendra la sortie(OUTPUT). Dans chacune broche, on prendra le code `digitalWrite(Pin, HIGH)` qui peut transmettre une valeur HIGH ou LOW à une broche numérique. C'est-à-dire, si la broche a été configurée comme OUTPUT avec `pinMode()`, sa tension sera réglée sur la valeur correspondante: 5V avec l'instruction HIGH, 0V (la masse) avec LOW.

Une chose vraiment importante que nous ne pouvons ignorer, c'est que nous devrions toujours mettre un delay (temps en millisecondes) entre `digitalWrite(Pin, HIGH)` et `digitalWrite(Pin, LOW)` sinon, la carte Arduino n'aura pas le temps de fournir la tension à la broche. Si cela se produit, la carte Arduino semblera ne rien faire après avoir reçu la commande.

À la fin, une autre chose que nous avons trouvée intéressante, c'est que le programme Arduino fonctionnera comme une boucle et ne s'arrêtera jamais. Ainsi, nous devons trouver un moyen d'éviter cela car nous ne voulons pas que ces pixels tournent infiniment. Nous avons ajouté une boucle `while (1)` au début et `return 0` à la fin du programme qui permettrait de faire tourner le programme une seule fois.

3.2.2. Problèmes et phases de tests

Nous avons d'abord réalisé le montage avec une seule sortie ligne et 5 sorties colonne, la ligne étant reliée à la masse. Ainsi, nous avons branché directement les 2 fils de sortie sur les broches du module, cela nous permettait de choisir quelle pastille allait tourner. Cette étape de test était nécessaire pour vérifier que le montage réalisé était fonctionnel sur une seule pastille avant que nous l'étendions à plusieurs.

Nous avons souvent rencontré des problèmes sur ce test : cela provenait parfois d'erreurs d'inattention (fil mal branché sur alimentation ou sur la platine d'essai par exemple) vite résolues, mais plus souvent de problèmes avec les composants : par exemple les transistors Darlington ne semblaient pas adaptés à la platine d'essai, leurs embouts étaient trop larges pour rentrer sans forcer dans les trous, il nous arrivait qu'il y ait des faux-contacts. Plusieurs fois, un petit transistor a grillé ce qui produisait un trou dans le circuit et engendrait des coupures de courant. Il fallait alors mesurer les potentiels avant et après chaque composant à l'aide d'un multimètre afin d'identifier la source du problème. Ces quelques soucis techniques nous ont fait perdre beaucoup de temps et étaient assez récurrents malheureusement.

Ensuite, nous avons placé les 5 transistors Darlington NPN TIP142 pour créer les 5 sorties du circuit pour commander les colonnes et nous avons remplacé l'alimentation 5V délivrée par un générateur par la carte Arduino. Elle était codée de sorte à délivrer une tension colonne par colonne pour que les pixels situés sur la même ligne se retournent tour à tour, de la droite vers la gauche.

Malheureusement, le transistor a de nouveau grillé et nous avons cherché à revoir le montage pour le remplacer par un plus adapté mais nous avons manqué de temps.

3.3. Résultats et solution finale

3.3.1. Résultats attendus

Après la présentation du projet, nous avons bien entendu comme objectif de pouvoir piloter dans un premier temps un module puis de reproduire notre montage pour en contrôler plusieurs. Nous voulions afficher une lettre sur un module en faisant tourner ses pixels en fonction du code que nous écrivions pour programmer l'Arduino. L'idée était ensuite de réinitialiser ce module (reset), c'est-à-dire de repasser tous les pixels du côté noir puis d'afficher une seconde lettre. Nous aurions aimé étendre cela à plusieurs modules pour afficher des mots entiers les uns à la suite des autres pour former une phrase.

Nous espérions avoir le temps de réaliser un panneau d'affichage fonctionnant selon nos envies et ainsi de faire une démonstration lors de la soutenance.

3.3.2. Résultats obtenus

Lors de nos dernières séances, nous avons réussi à faire tourner les pixels d'une même ligne tour à tour grâce au signal que nous envoyions avec l'Arduino. Nous branchions une ligne et les 5 colonnes sur l'Arduino qui envoyait donc un courant de 5V tour à tour dans les colonnes en respectant les consignes données par le code informatique.

Malheureusement nous avons eu une fois de plus un problème avec un composant : le transistor PNP permettant d'alimenter la colonne était mal calibré. L'intensité qui le traversait était trop grande et il a grillé. Nous avons donc essayé de le remplacer par un Darlington TIP 147 mais son comportement était différent. Nous avons tenté d'augmenter la résistance en amont de ce transistor pour qu'il arrête de se comporter uniquement comme un interrupteur fermé. Nous n'avons pas eu le temps d'ajuster la résistance pour qu'il se comporte de la bonne manière et que le montage puisse fonctionner sans risque d'abîmer les composants.

Notre montage était donc fonctionnel, tout comme le code informatique, mais nous aurions eu besoin d'un transistor PNP résistant à une plus grande intensité pour que l'ensemble fonctionne.

3.3.3. Résultat théorique

Avec un peu plus de temps nous aurions pu également faire le montage pour contrôler chaque ligne. Ceci nous aurait permis de contrôler l'intégralité des pixels du module directement à partir de la carte Arduino et ainsi de choisir lesquels faire tourner pour écrire des lettres. Cependant, nous n'avons pas pu faire le montage pour la ligne car l'inverseur commandé n'avait pas le comportement que nous espérions : il n'arrivait pas à transformer un potentiel d'entrée positif en un potentiel de sortie négatif. Nous avons donc pensé à le remplacer par un amplificateur opérationnel inverseur afin de fonctionner comme nous l'aurions souhaité, nous aurions pu contrôler l'intégralité des pixels un à un que ce soit pour les passer du côté jaune ou bien les réinitialiser côté noir.

3.3.4. Autres pistes

Le schéma du montage fourni semblait bon cependant nous avons constaté que le choix des composants n'était peut-être pas adapté à une alimentation 12V qui pouvait envoyer du courant jusqu'à 3A. Ainsi les transistors PNP n'étaient pas suffisants pour une intensité aussi forte et grillaient souvent. Il faudrait donc revoir le choix des composants pour les adapter à notre alimentation. Sinon la solution mise en place semblait être la bonne car la tension en sortie ainsi que l'intensité étaient suffisantes pour faire tourner les pixels.

4. CONCLUSION ET PERSPECTIVE

Ce projet de physique a permis de développer 2 compétences nécessaires au métier d'ingénieur : l'apprentissage autonome et le travail en groupe.

En effet, durant les premières séances, nous avons fait beaucoup de recherches, principalement sur internet, afin de nous documenter sur le fonctionnement des transistors, de l'inverseur ou de la carte Arduino Méga.

Nous avons ensuite commencé le montage, il nous a fallu comprendre comment fonctionnait une platine d'essai pour réaliser les branchements. Lors de la préparation du montage, nous avons dû nous poser les bonnes questions lorsque ça ne fonctionnait pas afin d'identifier la source des erreurs et de pallier aux différents problèmes rencontrés.

Nous avons aussi appris à représenter notre montage sur le logiciel Litzing pour avoir un schéma clair du montage et avons découvert le langage C pour programmer la carte Arduino.

Nous avons aussi dû nous assurer de garder une bonne communication à chaque séance afin de vérifier que la partie codage et montage du projet avançaient de façon coordonnée.

Nous avons pu constater que ce projet utilisait beaucoup de notions d'électronique que nous n'avons pas encore étudiées à l'INSA. La P3 semblait être une introduction à l'électronique et a parfois pu nous aider lors de ce projet.

Ainsi, ce projet nous a permis d'élargir nos connaissances en informatique et électronique et de développer nos compétences d'organisation, de recherche, de modélisation et réalisation d'un montage électronique.

5. BIBLIOGRAPHIE

- [1] Allegro Inc, “Datasheet of UDN2981 et UDN2982”
- [2] Maxim Ltd, “Datasheet of MAX7219 DS”
- [3] Eskimon et Olyte, « Arduino pour bien commencer en électronique et en programmation », OpenClassRoom, v2.5,2012.
- [4] lien internet : <http://tronixstuff.com/2013/10/11/tutorial-arduino-max7219-led-display-driver-ic> (valide à la date du 12/06/2016) : Site proposant un exemple d'utilisation du MAX7219.
- [5] <http://www.alldatasheet.fr/view.jsp?Searchword=2N2222>
- [6] <http://www.alldatasheet.fr/view.jsp?Searchword=2N2905>
- [7] Transistor Darlington PNP <http://www.alldatasheet.fr/view.jsp?Searchword=TIP147>
- [8] Transistor Darlington NPN <http://www.alldatasheet.fr/view.jsp?Searchword=TIP142>

6. TABLE DES ILLUSTRATIONS

Figure 1 : Image du panneau d'affichage.....	1
Figure 2 : Intérieur du panneau d'affichage.....	8
Figure 3 : Test de fonctionnement des modules.....	9
Figure 4 : Exemple d'affichage possible.....	10
Figure 5 : Carte Arduino.....	11
Figure 6 : Schéma du montage.....	13
Figure 7 : Symbole d'un transistor bipolaire.....	14
Figure 8 : Transistor Darlington.....	15
Figure 9 : Montage inverseur d'un amplificateur opérationnel.....	15

7. ANNEXES

7.1. Programme théorique final

Le programme est le suivant :

```
int L1 = 7 ;//Ligne1 pour broche 7
int C5 = 6;//Colonne 5 pour broche 6
int C4 = 5;
int C3 = 4;
int C2 = 3;
int C1 = 2;
void setup() {
// Mettez les broches comme sortie

pinMode(L1 , OUTPUT);
pinMode(C5 , OUTPUT);
pinMode(C4 , OUTPUT);
pinMode(C3 , OUTPUT);
pinMode(C2 , OUTPUT);
pinMode(C1 , OUTPUT);
}

void loop() {
//Tourner les pixels.

while(1){ // Terminer la boucle
digitalWrite(L1 , HIGH);
```

```
delay(500);  
digitalWrite(C1 , HIGH);  
delay(500);  
digitalWrite(C2 , HIGH);  
delay(500);  
digitalWrite(C3 , HIGH);  
delay(500);  
digitalWrite(C4 , HIGH);  
delay(500);  
digitalWrite(C5 , HIGH);  
delay(500);  
digitalWrite(L1 , LOW);  
delay(500);  
digitalWrite(C1 , LOW);  
delay(500);  
digitalWrite(C2 , LOW);  
delay(500);  
digitalWrite(C3 , LOW);  
delay(500);  
digitalWrite(C4 , LOW);  
delay(500);  
digitalWrite(C5 , LOW);  
delay(500);  
}  
  
return 0;  
  
}
```