

L3 E3a

Projet TEE

Automate détecteur de source de lumière

Rédigé par:

MEDDOUR Antara Souheib

année universitaire : 2024/2025:

Contents

1	Remerciment	2
2	Introduction	3
2.1	Principe	3
2.2	Matériel utilisé	4
3	Démentionement	5
3.1	Diviseur de tension	5
3.2	Relais	5
3.3	transimpédance	6
3.3.1	Mésure	6
3.4	Amplificateur	7
3.5	Optocoupleur	8
3.5.1	Dimensionnement	9
3.6	ULN2003	10
4	explication en bloc	11
4.1	bloc de mesure :	11
4.2	bloc de commande	12
5	Routage du circuit sur logiciel Eagle	13
6	Les étapes pour crée la carte électronique	14
6.1	Impression	14
6.2	Perçage	15
6.3	Soudure	15
7	Programation	16
7.1	NUCLEO-F303RE	16
8	Bibliographie	20
9	Anexes	21

List of Figures

1	une vias	13
---	--------------------	----

1 Remerciment

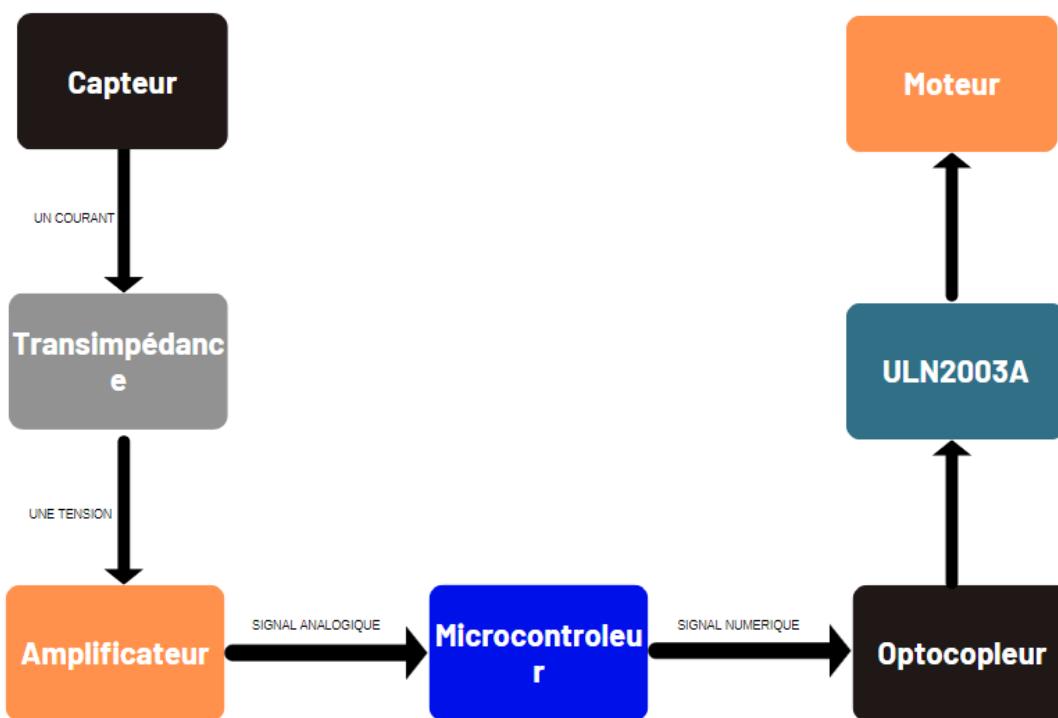
Je tiens à exprimer ma profonde gratitude envers nos encadrants, M.BELKACEM Rafik, M.Pawel Wzietek pour ses guidance précieuse tout au long de ce projet. Son expertise, son soutien et ses conseils avisés ont été d'une importance capitale pour le développement de ce travail. Sa disponibilité et sa patience ont grandement contribué à mon apprentissage et à la réussite de ce projet. Je tiens également à le remercier pour son engagement et son enthousiasme constants, qui ont été une source d'inspiration tout au long de ce parcours. Merci infiniment pour votre précieuse contribution à la réalisation de ce projet.

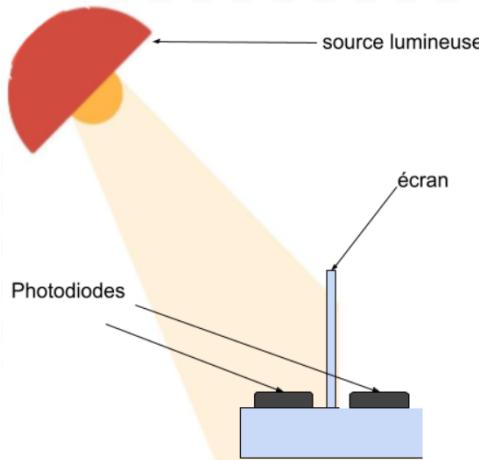
2 Introduction

Dans le cadre de notre formation en Licence 3 en Electronique, Energie Electrique et Automatique (E3A) à l'Université Paris-Saclay, nous avons réalisé un projet qui met en œuvre de nombreux concepts clés de l'électronique. Notre objectif était de concevoir un automate capable de suivre une source lumineuse mobile en utilisant un microcontrôleur et des capteurs photodiodes. Nous avons choisi d'utiliser un amplificateur transimpédance à gain ajustable et un convertisseur A/D 12 bits pour mesurer l'intensité lumineuse captée par les capteurs. Le microcontrôleur a été programmé pour contrôler le mouvement d'un moteur pas à pas en fonction des mesures de l'intensité lumineuse, afin de trouver et de suivre la source de lumière. Dans ce rapport, nous décrirons en détail notre méthode de conception et de réalisation, ainsi que les résultats obtenus.

2.1 Principe

L'objectif principal de ce projet est de concevoir un automate piloté par un microcontrôleur programmable, capable de suivre une source lumineuse en mouvement. Le rôle essentiel du microcontrôleur est de réguler le mouvement du moteur et de mesurer l'intensité lumineuse grâce à deux photodiodes montées sur le rotor de l'automate, permettant ainsi au moteur de s'orienter vers la direction où l'intensité lumineuse est la plus élevée. Pour atteindre cet objectif, nous allons mettre en place deux modes de fonctionnement sur le microcontrôleur : un premier mode détectant la source lumineuse la plus intense afin de s'aligner directement avec elle, et un second mode permettant de suivre cette source lumineuse en cas de déplacement.



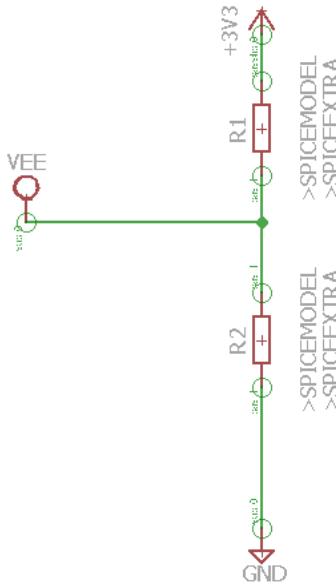


2.2 Matériel utilisé

- Une alimentation DC
- Un oscilloscope
- Une sonde différentielle
- Un microcontrôleur NUCLEO-F303RE, Relai, Amplificateur, Transimpédance.
- Tout le matériel de soudure
- Tout le matériel de perçage
- Des fils électriques, câbles coaxiaux et platine d'essai
- Un optocoupleur.
- Moteur courant continu (pas à pas)
- capacités et résistances, photodiodes.

3 Démentionement

3.1 Diviseur de tension



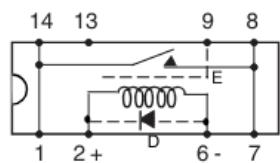
Nous avons d'abord dimensionné R_1 et R_2 pour obtenir une masse virtuelle à $V_{ref}=1,65V$. Nous avons utilisé un simple diviseur de tension pour y parvenir, en choisissant des résistances égales de 1000 ohms pour ($R_1 = R_2 = 1000 \Omega$). Cette valeur est importante car elle correspond à la moitié de la tension d'alimentation de notre système, qui est de 3,3V

$$V_{ref} = \frac{R_1}{R_1 + R_2} \cdot V_e$$

On a $R_1 = R_2 = 1000 \Omega$ et $V_e = 3.3 \text{ V}$, Donc:

$$\begin{aligned} V_{ref} &= \frac{R_1}{2 \cdot R_1} V_e \\ V_{ref} &= \frac{1}{2} V_e = \frac{3,3}{2} \\ V_{ref} &= 1.65V \end{aligned}$$

3.2 Relais



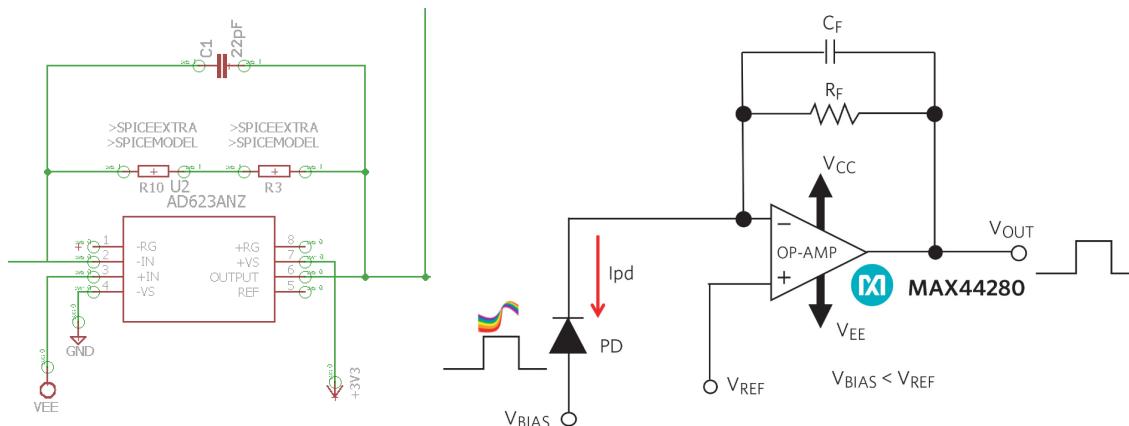
CARACTERISTIQUES ELECTRIQUES / ELECTRICAL CHARACTERISTICS

Nature du contact / contact material	rhodium
Position de fonctionnement / position to operate	indifférente /any
Puissance de communication / power switching	max 10 W
Courant traversant / carrying current	1 A
Courant max commuté / max switching current	500 mA
Tension max de commutation / max switching voltage	100 Vcrête
Fréquence de commutation / switching frequency max	200 Hz
Résistance d'isolation à circuit ouvert / insulation resistance	$10^{10} \Omega$

on voit entre 2-6 y'a une diode «de roue libre» qui assure la protection d'un transistor alimentant des charges inductives (comme les bobinages d'un moteur) .

3.3 transimpédance

Un amplificateur transimpédance est un type spécifique d'amplificateur conçu pour convertir un courant en tension



- Entrée de courant:** Le courant à convertir est généralement fourni par un dispositif tel qu'une photodiode, qui génère un courant proportionnel à la lumière qu'elle reçoit.
- Connexion à l'amplificateur opérationnel:** Le courant de sortie de la photodiode est dirigé vers l'entrée inverseuse (-) de l'amplificateur opérationnel.
- Boucle de rétroaction:** Une résistance, appelée résistance de feedback (R_3 et R_{10}), est placée entre la sortie de l'amplificateur opérationnel et l'entrée inverseuse (-). Cette configuration crée une boucle de rétroaction négative.
- Conversion en tension:** Comme l'amplificateur opérationnel a une très grande impédance d'entrée, pratiquement aucun courant ne circule dans l'entrée inverseuse (-). Par conséquent, tout le courant de la photodiode traverse la résistance de feedback (R_3 et R_{10}). Selon la loi d'Ohm ($V = I * R$), le courant traversant R génère une tension de sortie proportionnelle.
- Amplification:** L'amplificateur opérationnel amplifie cette tension de sortie selon son gain, qui est déterminé par les valeurs des composants et les caractéristiques de l'amplificateur

En résumé, un amplificateur transimpédance utilise la résistance de feedback pour convertir le courant d'entrée en une tension de sortie proportionnelle, offrant ainsi une manière pratique de mesurer et amplifier de faibles courants tels que ceux produits par les photodiodes.

3.3.1 Mésure

- Nous avons effectué deux tests pour déterminer cette valeur : un test sous la lumière directe du soleil et un test avec le flash de notre téléphone. En prenant la moyenne des résultats des deux tests, nous avons trouvé que la valeur de courant maximale $I_{max} = 100 \mu\text{A}$.

On a

$$V_s = V_{ref} - R \cdot I_{max}$$

$$R = \frac{V_s - V_{ref}}{I_{max}}$$

Donc

$$R = \frac{3,3 - 1,65}{100 \cdot 10^{-6}} = 16,5 K\Omega$$

2. Pour atténuer les fréquences de 25 Hz provenant des néons dans la salle, nous avons inclus un condensateur dans notre premier bloc pour créer un filtre passe-bas. Afin de calculer la valeur de la capacité requise pour ce filtre, nous avons utilisé une fréquence de coupure de 25 Hz pour déterminer la valeur de la capacité C

On a

$$w_c = \frac{1}{R.C}$$

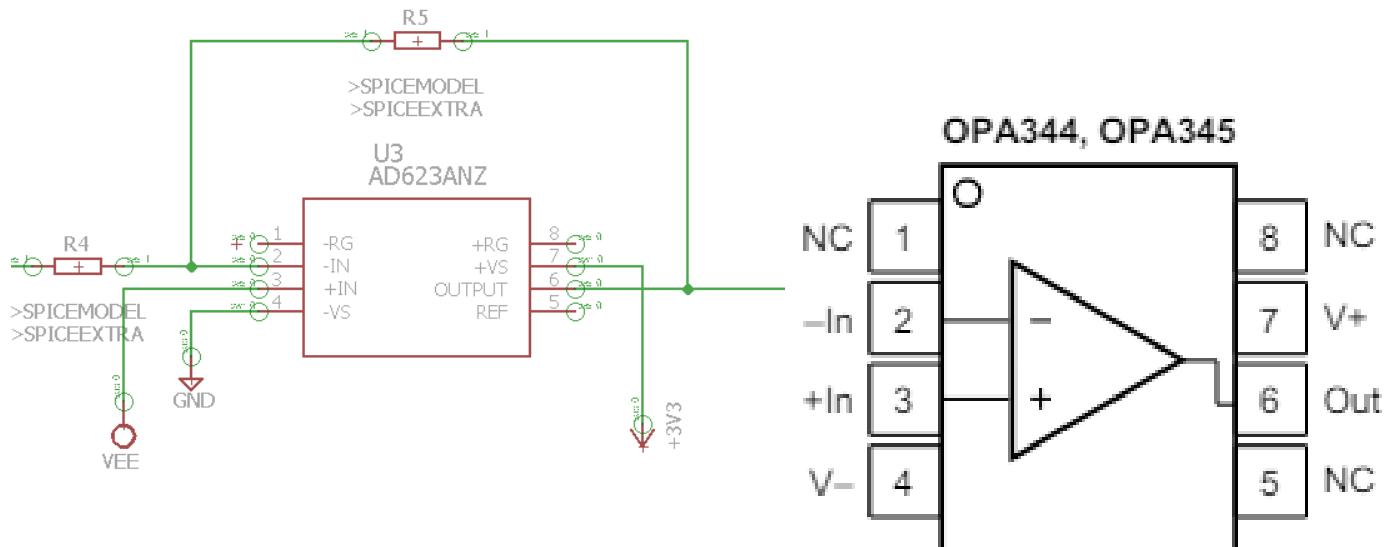
$$2.\pi.f_c = \frac{1}{R.C}$$

$$C = \frac{1}{2.\pi.R.f_c}$$

$$C = \frac{1}{2.\pi.16500.25}$$

$$C = 0.38 \mu F$$

3.4 Amplificateur



1 Vs1: la sortie de Transimpédance
2 Vs2: la sortie de l'Amplificateur

selon Millmanne

$$V^- = \frac{\frac{V_{s1}}{R_4} + \frac{V_{s2}}{R_5}}{\frac{1}{R_4} + \frac{1}{R_5}} = 0$$

Alors

$$V^- = \frac{V_{s1}}{R_4} + \frac{V_{s2}}{R_5} = 0$$

Donc

$$\frac{V_{s2}}{V_{s1}} = -\frac{R_4}{R_5}$$

$$|G| = \frac{V_{s2}}{V_{s1}} = \left| \frac{R_4}{R_5} \right|$$

On a Choisie un **Gain de 10** Donc

$$\frac{R_4}{R_5} = 10$$

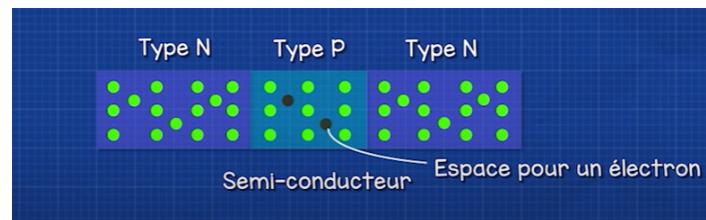
$$R_4 = 10.R_5$$

Nous prurons $R_4 = 1000 \Omega$ Donc $R_5=10000 \Omega$

3.5 Optocoupleur

Les optocoupleurs sont des composants électroniques intégrés qui se présentent sous forme de boîtiers comprenant quatre broches principales : l'anode, la cathode, le collecteur et l'émetteur, leur rôle principal est de protéger un circuit sensible des interférences électriques ou des surtensions pouvant être présentes dans un autre circuit et isoler électriquement deux circuits tout en permettant la transmission de signaux optiques entre eux.

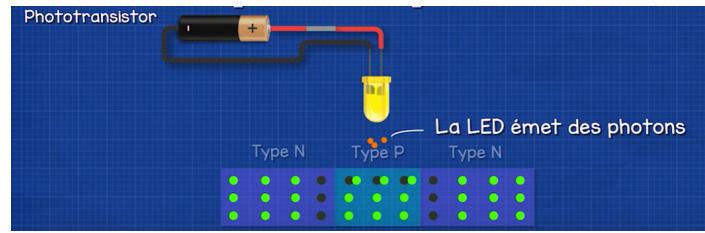
- à l'intérieur du photo transistors nous avons différentes couches de matériaux semi conducteurs il y a le type n est le type p qui sont pris en sandwich le type n est le type p sont tous deux fabriqués à partir de silicium mais ils ont été mélangées avec d'autres matériaux pour modifier leurs propriétés électriques le type n a été mélangée avec un matériau qui lui donne beaucoup d'électrons supplémentaires et inutile ceux ci sont libres de se déplacer vers d'autres atomes le type p a été mélangée avec un autre matériau qui contient moins d'électrons il y a donc beaucoup d'espaces vides ou les électrons peuvent se déplacer lorsque les matériaux sont assemblés une barrière électrique se forment et empêche les électrons de circuler.



- lorsque la led est allumée elle émet des photons qui frappent le matériau de type p et font tomber les électrons à travers la barrière et dans le matériau de type n où les électrons situés au niveau de la première barrière sont maintenant capables de faire le saut. Il y a alors un courant qui se développe. Une fois la led éteinte les photons cessent de faire franchir la barrière aux électrons et le courant s'arrête donc du côté secondaire.

Son rôle:

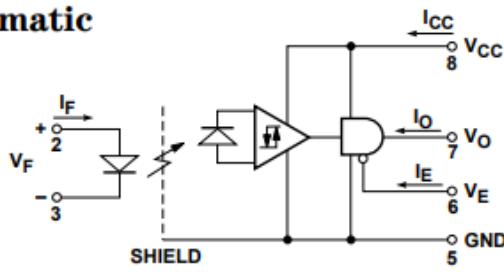
- Isolation électrique :** Le rôle principal des optocoupleurs est de fournir une isolation électrique entre deux circuits. Cela protège un circuit sensible des interférences électriques, des surtensions et des bruits présents dans l'autre circuit.



2. **Transfert de signal** : Les optocoupleurs permettent de transmettre des signaux entre deux circuits sans connexion électrique directe, en utilisant la lumière comme vecteur de communication. Cela peut être utile pour transmettre des signaux sur de longues distances ou dans des environnements bruyants.
3. **Niveau de tension différentiel**: Les optocoupleurs permettent aux circuits de fonctionner à des niveaux de tension différents tout en maintenant une isolation électrique. Cela peut être crucial dans les applications où des niveaux de tension différents sont nécessaires.
4. **Protection contre les surtensions** : En raison de leur isolation électrique, les optocoupleurs offrent une protection contre les surtensions et les tensions transitoires dans un circuit, car elles empêchent les surtensions de se propager d'un circuit à l'autre.

3.5.1 Dimensionnement

Schematic



Minimum CMR		Input On-Current (mA)	8-Pin DIP (300 Mil)	
dV/dt (V/μs)	V _{CM} (V)		Single Channel Package	Dual Channel Package
1,000	50	1.6	HCPL-2200 ^[1] HCPL-2201 HCPL-2202	

Des résistances ont été ajoutées avant les optocoupleurs pour que la LED associée fonctionne à son point nominal. Afin de calculer la valeur de la résistance associée à l'optocoupleur, on prend en compte la tension de chute de la led indiquée par le constructeur:

$$V_R = V_{max} - V_{chutuled}$$

$$V_R = 3,3 - 1,5 = 1,8V$$

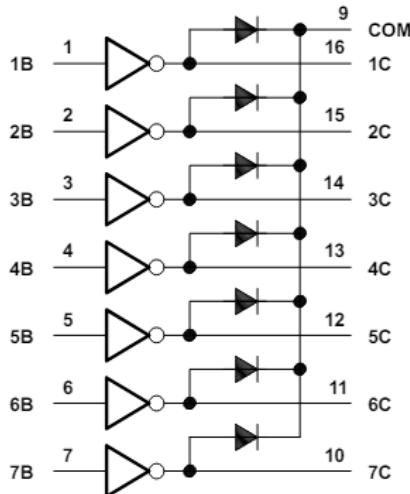
On souhaite obtenir un courant $i_R = 2$ mA dans la résistance, donc d'après la loi d'Ohm :

$$R = \frac{V_R}{i_R} = \frac{1,8}{2 \cdot 10^{-3}}$$

$$R = 900 \Omega$$

3.6 ULN2003

ULN2003a, un circuit intégré comprenant une matrice de 7 transistors Darlington PNP capables de fournir jusqu'à 500 mA de courant et une sortie de 50 volts. Il est connu pour ses capacités de courant élevé et de haute tension, avec des versions pouvant même gérer jusqu'à 100 volts. L'IC comprend des diodes de roue libre en cathode commune pour la commutation de charges inductives.



PARAMETER	TEST FIGURE	TEST CONDITIONS	ULN2003A			ULN2004A			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
$V_{I(on)}$ On-state input voltage	6	$V_{CE} = 2\text{ V}$	$I_C = 125\text{ mA}$					5	V
			$I_C = 200\text{ mA}$			2.4		6	
			$I_C = 250\text{ mA}$			2.7			
			$I_C = 275\text{ mA}$					7	
			$I_C = 300\text{ mA}$			3			
			$I_C = 350\text{ mA}$					8	
$V_{CE(sat)}$ Collector-emitter saturation voltage	5		$I_I = 250\text{ }\mu\text{A}, I_C = 100\text{ mA}$			0.9	1.1	0.9	1.1
			$I_I = 350\text{ }\mu\text{A}, I_C = 200\text{ mA}$			1	1.3	1	1.3
			$I_I = 500\text{ }\mu\text{A}, I_C = 350\text{ mA}$			1.2	1.6	1.2	1.6
I_{CEX} Collector cutoff current	1	$V_{CE} = 50\text{ V}, I_I = 0$				50		50	μA
			$V_{CE} = 50\text{ V}, I_I = 0$			100		100	
V_F Clamp forward voltage	8	$I_F = 350\text{ mA}$				1.7	2	1.7	2
			$V_I = 3.85\text{ V}$						
$I_{I(off)}$ Off-state input current	3	$V_{CE} = 50\text{ V}, I_C = 500\text{ }\mu\text{A}, T_A = 70^\circ\text{C}$	50	65		50	65		μA
I_I Input current	4	$V_I = 3.85\text{ V}$				0.93	1.35		mA
		$V_I = 5\text{ V}$						0.35	
I_R Clamp reverse current	7	$V_I = 12\text{ V}$						1	1.45
		$V_R = 50\text{ V}$				50		50	μA
C_i Input capacitance		$V_I = 0, f = 1\text{ MHz}$				15	25	15	25
									pF

Selon le datasheet du ULN2003A à $V_i = 3,85\text{ V}$ On a $I_{max} = 1,35\text{ mA}$ et $I_{typ} = 0,93\text{ mA}$ Alors:

$$R_{max} = \frac{V_i}{I_{max}} = \frac{3,85}{1,35} = 2851,85\text{ }\Omega$$

et

$$R_{typ} = \frac{V_i}{I_{typ}} = \frac{3,85}{0,93} = 4139,78\text{ }\Omega$$

Alors ça justifier de prendre

$$R_8 = R_9 = 1000\text{ }\Omega$$

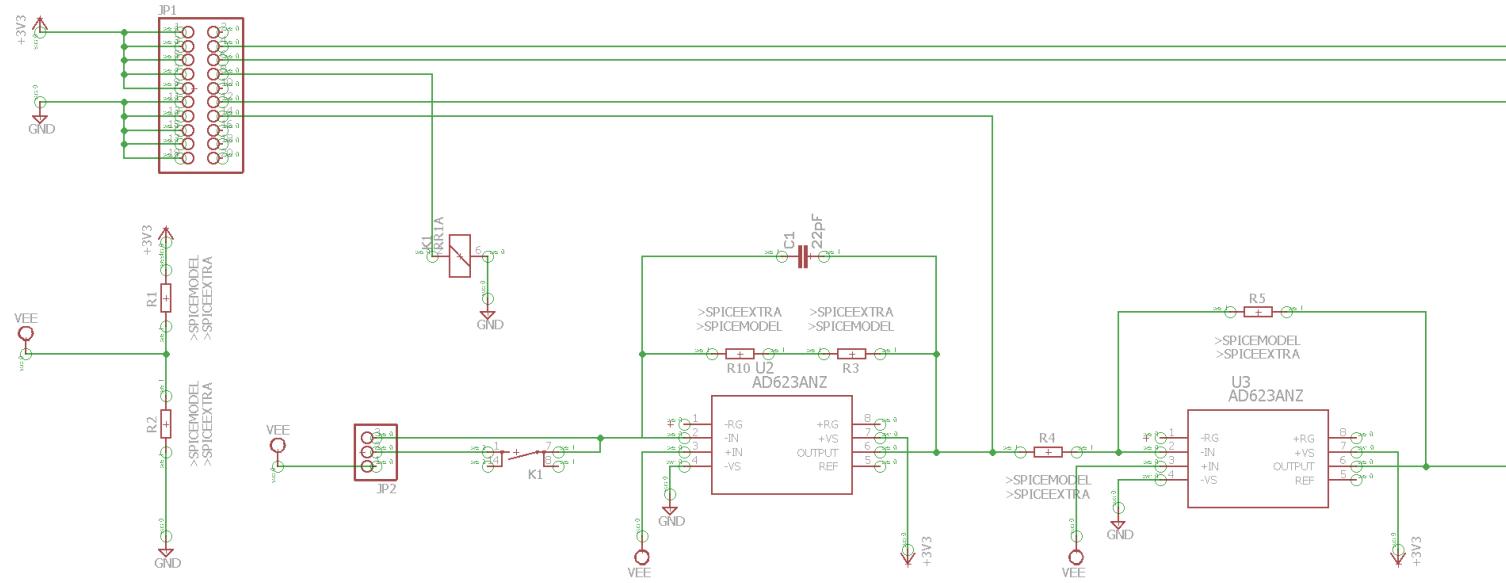
ULN 2003 contient des diodes de roue libre qui assure la protection d'un transistor alimentant des charges inductives (comme les bobinages d'un moteur)

pourquoi aurait-on besoin d'inverseurs logiques pour commander un moteur unipolaire?

En résumé, l'intégration d'inverseurs logiques dans la commande d'un moteur unipolaire avec l'ULN2003 permet de gérer efficacement le contrôle des bobinages, d'adapter les niveaux de signal et de simplifier la conception globale du circuit, tout en maximisant l'utilisation des composants disponibles pour des performances optimales et une flexibilité accrue.

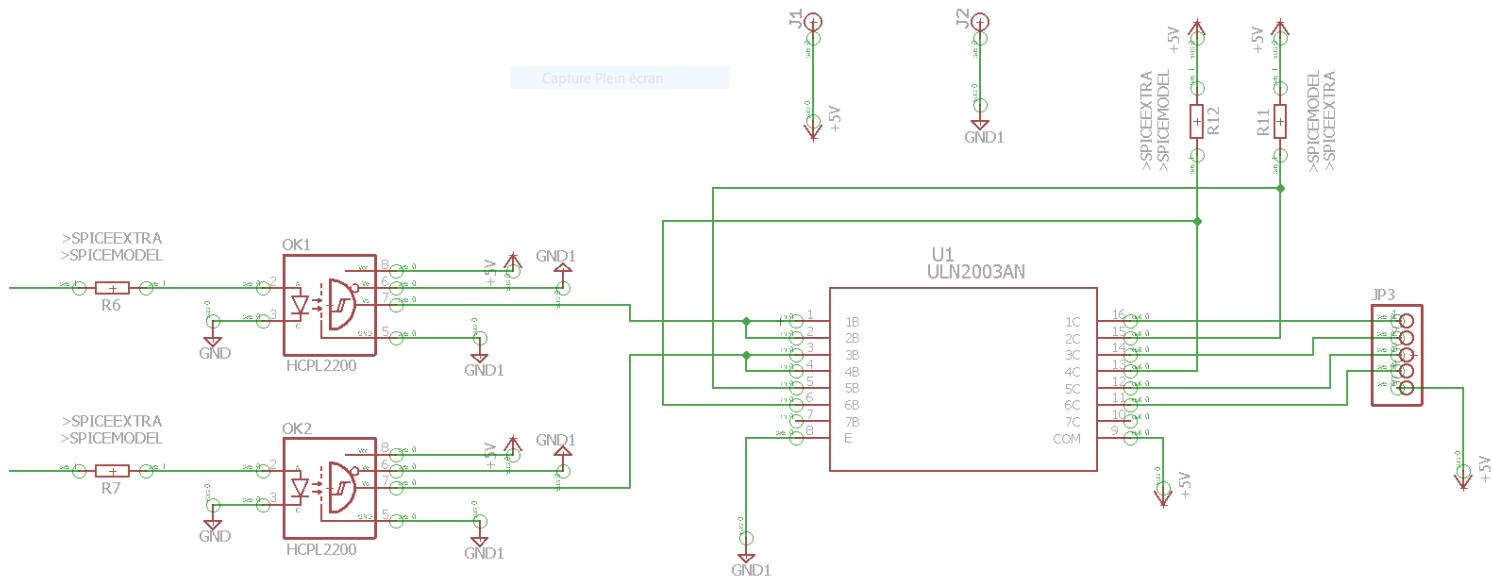
4 explication en bloc

4.1 bloc de mesure :



Dans cette configuration, l'objectif est de mesurer la quantité de lumière traversant un photodiode (un capteur de lumière). Pour ce faire, nous utilisons un photodiode avec deux modes différents : mode 1:(utilisant une seule diode) et mode 2 :(utilisant deux diodes). Le courant généré par la photodiode est converti en tension à l'aide d'un transimpédance avec un condensateur pour éliminer les interférences lumineuses externes. Ensuite, pour amplifier le signal de sortie, nous avons ajouté un deuxième étage d'amplification utilisant un autre amplificateur. Les deux signaux de sortie amplifiés sont ensuite envoyés à un microcontrôleur via un connecteur à 10 pin. En ce qui concerne l'alimentation des deux amplificateurs, nous utilisons la tension de 3,3 V du microcontrôleur pour V_+ et V_{ref} pour V_- (en utilisant un pont diviseur de tension avec la tension de 3,3 V). Pour commuter entre les deux modes, nous avons introduit un relais équipé d'une diode de roue libre pour protéger le relais. Ce relais est activé par une tension de 5 V provenant du microcontrôleur. Lorsque le relais est activé, il permet d'utiliser le mode 2 . Cette utilisation du relais permet de sélectionner efficacement le mode de fonctionnement souhaité en réponse à une commande du microcontrôleur.

4.2 bloc de commande

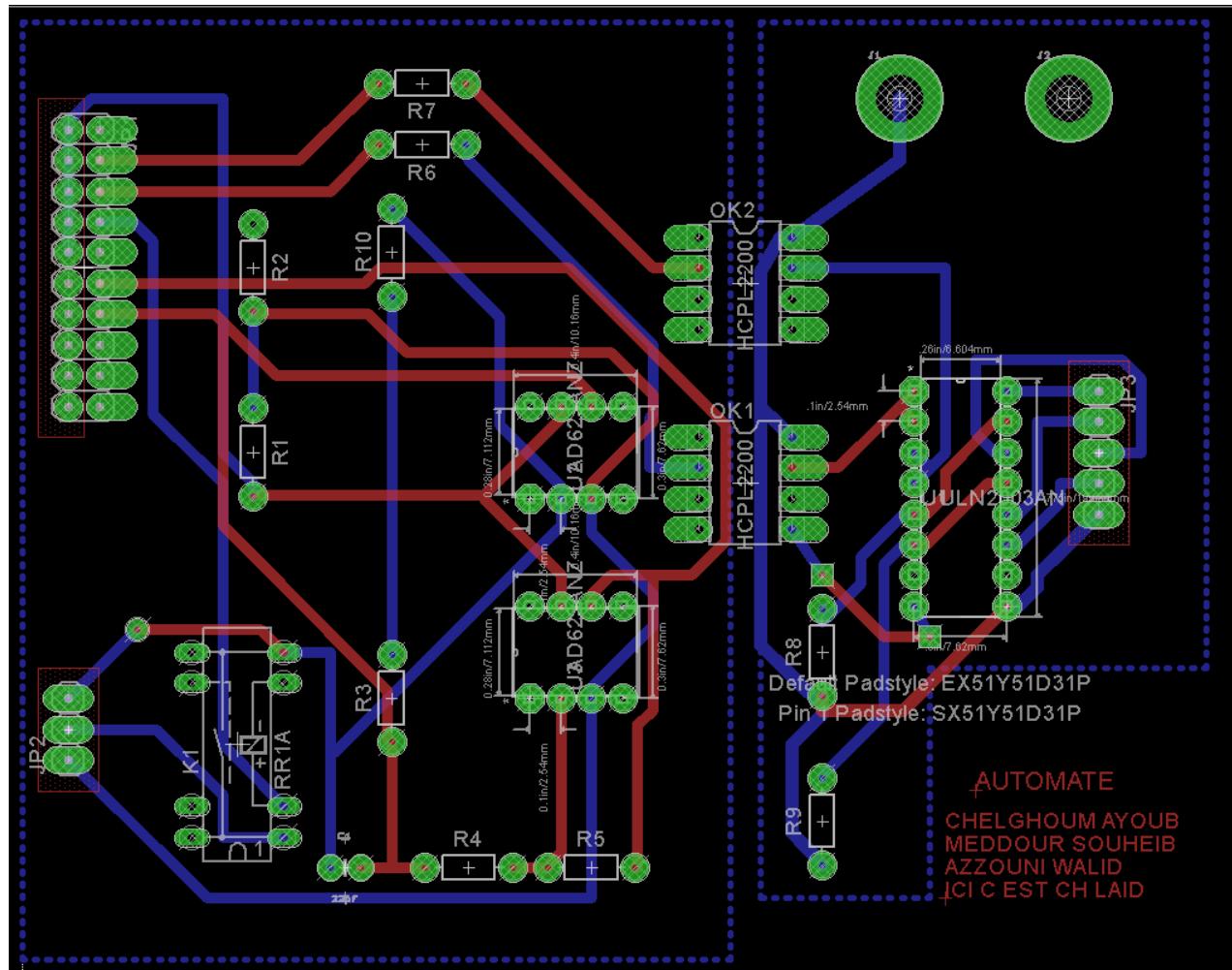


Dans cette partie, nous utilisons un moteur que nous commandons à l'aide de signaux provenant du microcontrôleur. Pour assurer une isolation électrique et protéger les composants, nous envoyons les signaux du microcontrôleur à deux optocoupleurs. Le rôle de ces optocoupleurs est de séparer électriquement les deux parties du circuit. Ensuite, les signaux provenant des optocoupleurs sont transmis à un circuit ULN -un ensemble de transistors Darlington- qui génère (avec deux résistances) les signaux inversés nécessaires pour alimenter le moteur pas à pas.

Pour alimenter à la fois le moteur pas à pas et le circuit ULN, nous utilisons une autre alimentation de 5 V avec un autre GRND distincte de celle du microcontrôleur. Cela garantit une séparation adéquate des deux parties du circuit et évite les interférences indésirables entre le moteur et le reste du système.

5 Routage du circuit sur logiciel Eagle

Nous avons pu visualiser en 3D les différents éléments du circuit imprimé grâce au logiciel Eagle. Cette fonctionnalité nous a permis de vérifier l'emplacement des composants électroniques et leur interconnexion avant la fabrication du circuit imprimé. La visualisation en 3D a également facilité l'identification des erreurs de conception et des conflits d'interconnexion, ce qui a réduit le risque d'erreurs lors de la fabrication du circuit imprimé



On a créé une vias manuellement pour la résolution d'une piste.

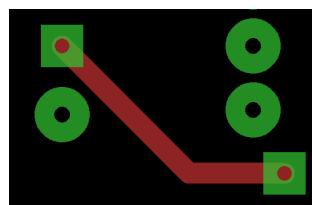
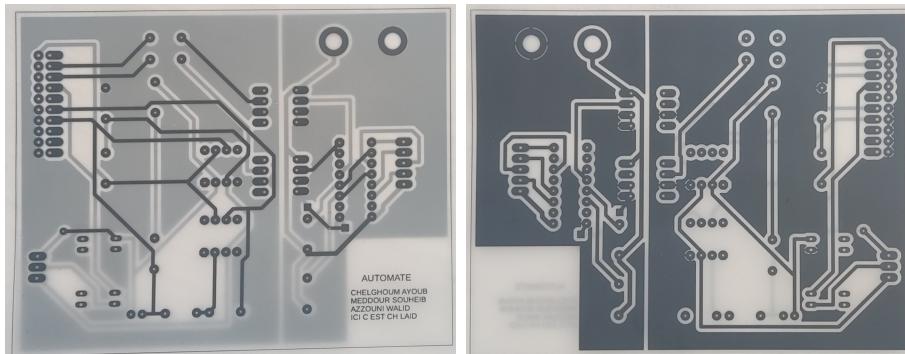


Figure 1: une vias

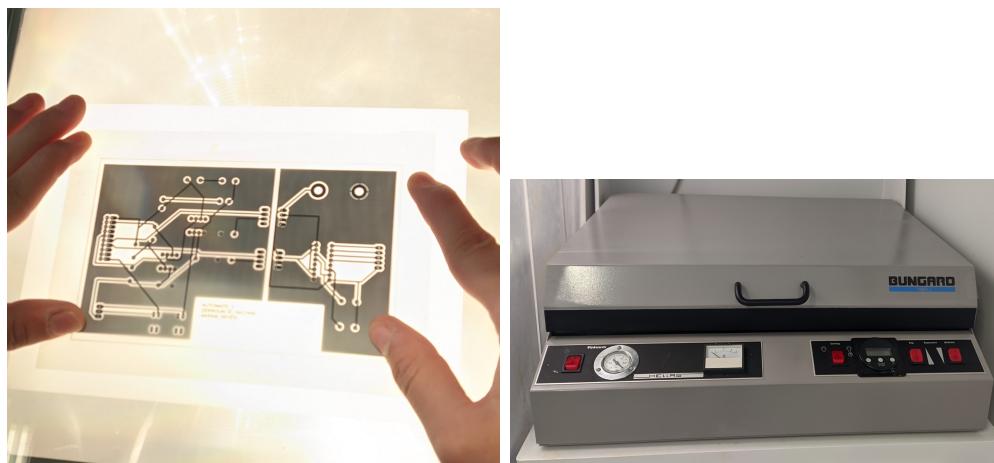
6 Les étapes pour crée la carte électronique

6.1 Impression

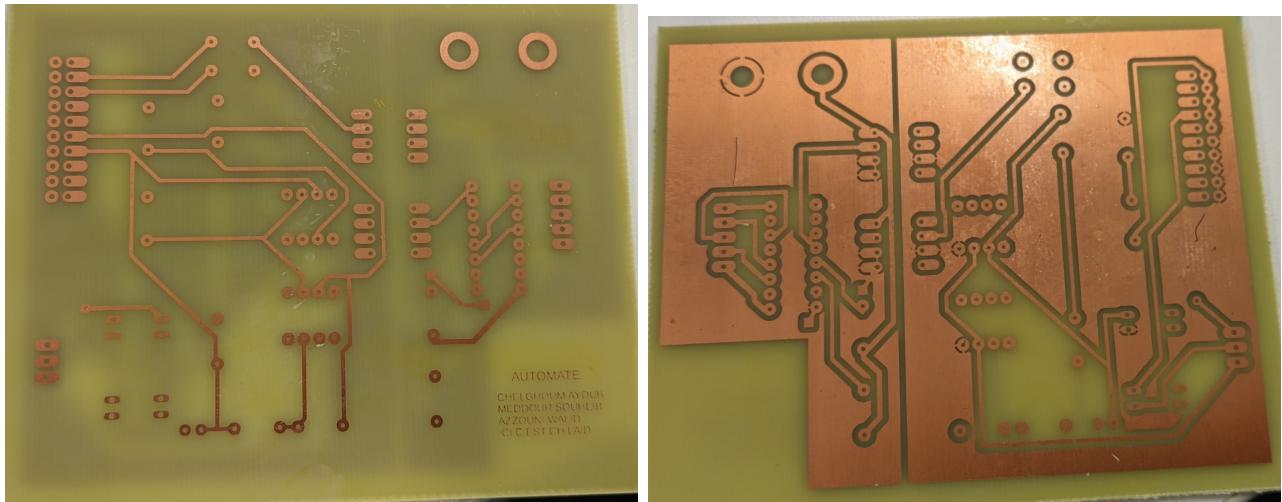
1. Imprimer le schéma sur un papier transparent avec une imprimante laser.



2. Positionner la plaque de cuivre au milieu du papier transparent, entre le côté top et le côté bottom sur le schéma imprimé et la fixer avec scotch et aussi mettre la carte dans la machine à rayons ultra-violets pendant 2 minutes pour le transfert du circuit sur la carte



3. Mettre la plaque sur le convoyeur de la machine contenant de la soude afin de sensibiliser la couche protectrice et avoir un circuit bien marqué. cette étape dure environ 15 minutes.
Etape suivante consiste à plonger la carte dans du Perchlorure de Fer à une température de 26°, qui va graver le circuit imprimé en détruisant tout le cuivre non recouvert de résine photosensible par réaction chimique
4. Rincer la plaque de cuivre à l'eau courante et la sécher.
5. réexposer la plaque de cuivre à la lumière UV pendant 2 minutes pour le nettoyage
6. rerincer une autres fois la plaques et la sécher soigneusement

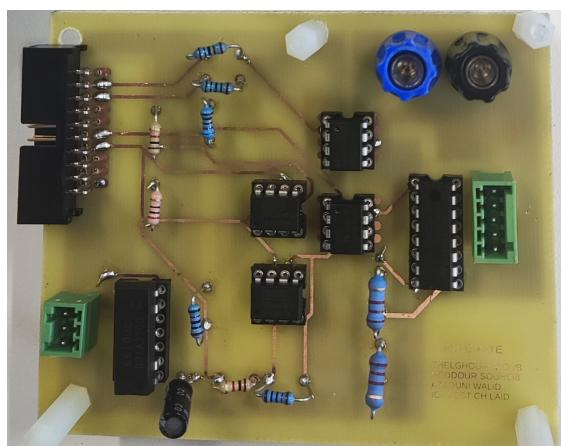
Résultat :

6.2 Perçage

Après avoir vérifié que la carte a été correctement imprimée, nous passons à l'étape suivante. Nous commençons par percer la carte en utilisant une perceuse. Nous faisons des petits trous (0.8mm) pour les résistances et les condensateurs, des trous moyens (environ 1.2mm) pour les relais et les connecteurs externes, et des grands trous (4mm) pour l'alimentation et les points de masse.

6.3 Soudure

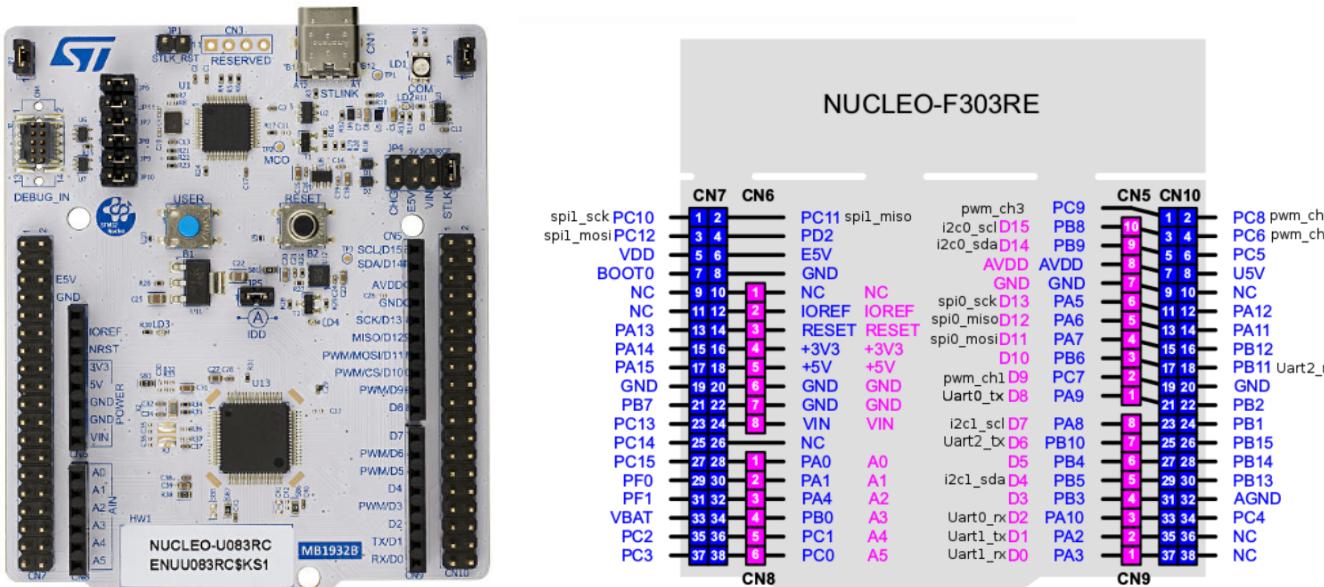
Après avoir effectué toutes les étapes précédentes, y compris l'impression du schéma et le perçage des trous, nous passons maintenant à l'étape cruciale du soudage des composants. Il est important de faire attention au sens de la carte (top ou bottom) et de bien souder les composants, en particulier là où il y a un contact avec une ligne, tout en vérifiant l'emplacement correct des composants. Il est recommandé que la personne qui a conçu le circuit soit également en charge du soudage. Après chaque soudage de composant, nous vérifions les connexions à l'aide d'un multimètre. Enfin, nous vérifions également toutes les connexions entre les composants pour nous assurer que tout est correctement relié.

Résultat:

7 Programmation

7.1 NUCLEO-F303RE

Le microcontrôleur utilisé dans le projet est le NUCLEO-F303RE de STM, la programmation de celui-ci se fait sur la plateforme Mbed en C++.



```
(a)
1 #include "mbed.h"
2
3 #define WAIT_TIME 20//msec
4
5
6
7 DigitalOut V1(D9);
8
9 DigitalOut V2(D4);
10
11 DigitalOut Vr(D7) ;
12
13 AnalogIn Vs1(A1);DS
14
15 AnalogIn Vs2(A0);
```

Dans ce début de programme utilisant la plateforme Mbed sur la carte Nucleo-F303RE, nous avons configuré des interfaces de sortie et d'entrée pour interagir avec le matériel externe. Les objets DigitalOut nommés V1, V2, et Vr (relais) sont associés respectivement aux pins D9, D4, et D7. Ces sorties numériques sont utilisées pour commander le moteur ainsi que l'utilisation de la deuxième photo diode avec l'activation ou la désactivation du relais. Parallèlement, les objets AnalogIn nommés Vs1 et Vs2, connectés aux pins A1 et A0, sont prévus pour lire des valeurs de tension analogiques. Ces entrées permettent de mesurer des signaux provenant des photo-diodes. La ligne include "mbed.h" est une directive de préprocesseur qui inclut la

bibliothèque principale de Mbed OS dans le programme, et la ligne define $WAITIME$ 20 définit une macro nommée $WAITIME$ avec une valeur de 20 millisecondes. Cette macro sert à établir une constante de temps qui peut être utilisée à travers le programme pour gérer des délais ou des temporisations, assurant ainsi que le microcontrôleur exécute certaines tâches avec une pause précise de 20 millisecondes entre elles.

(b)

```

1 int n = 200 ;
2
3 int i, U, p =0, index=0 ;
4
5 float maxx = 0 ,
6
7 float current_value;
```

Dans cette partie du programme nous avons défini n qui est le nombre de pas dans un tour du moteur, i pour l'utilisation des boucles ,la position p ou il y a plus de lumière, max pour le maximum de la valeur calculé par Vs1 et $current_{value}$ pour calculer la valeur de Vs1.

(c)

```

1 int main()
2 {
3
4     while (index==0)
5     {
6
7         Vr = 0 ;
8
9         maxx = 0;
10
11        for (i = 0; i < n; i++)
12        {
13
14            V1 = tab_V1[i%4] ;
15
16            V2 = tab_V2[i%4] ;
17
18
19            current_value= Vs1.read()*3.3;
20
21
22            if ( current_value > maxx)
23            {
24
25                maxx = current_value ;
26
27                p = i ;
```

```

31         U=i%4;
32
33     }
34
35     thread_sleep_for(WAIT_TIME);
36
37 }
```

Dans le main du programme , Dans une boucle infinie conditionnée par la variable index, il commence par désactiver le relais pour commencer dans le premier mode, puis il parcourt une série d'itérations (déterminée par n qui est le nombre des pas du moteur) où V1 et V2 sont mis à jour selon des séquences cycliques issues des tableaux tab_V1 et tab_V2 pour tourner le moteur. Parallèlement, il lit une valeur de tension de Vs1, ajustée à l'échelle réelle par multiplication par 3.3 car la fonction read() nous donne un pourcentage. Si cette tension excède la valeur maximale précédemment enregistrée (maxx), il met à jour maxx et stocke l'indice de cette lecture pour revenir à cette position après avoir terminé le tour. Entre chaque itération, il impose un délai de 20 millisecondes pour stabiliser les opérations. Ce programme permet au moteur de faire un tour et revenir à sa position initiale.

(d)

```

1  for (j =i; j>= p ; j--)
2
3  {
4
5      V1 = tab_V1 [(j+U)%4] ;
6
7      V2 = tab_V2 [(U+j)%4] ;
8
9      thread_sleep_for(WAIT_TIME);
10
11 }
12
13 index=index+1;
```

La boucle ‘for‘ décroissante dans ce segment contrôle les sorties ‘V1‘ et ‘V2‘ en décrémentant à partir de la position actuelle ‘i‘ jusqu’à ‘p‘, une position précédemment déterminée comme celle où la valeur maximale a été enregistrée (la position où il y a plus de lumière). À chaque itération, les valeurs de ‘V1‘ et ‘V2‘ sont mises à jour en utilisant les éléments des tableaux ‘ tab_V1 ‘ et ‘ tab_V2 ‘, respectivement pour tourner le moteur dans l’autre sens, avec les indices ajustés par ‘U‘, un décalage dérivé de la position de la valeur maximale pour que le moteur ne passe pas des pas. Le délai entre chaque mise à jour est contrôlé par ‘ $WAIT_TIME$ ‘ pour assurer la stabilité des changements de sortie. Après avoir terminé cette boucle, la variable ‘index‘ est incrémentée, suggérant une transition ou un changement d’état dans le comportement global du programme. Ce programme nous permet de faire un tour dans le sens opposé au premier pour revenir à la position où il y a plus de lumière.

(e)

```

1  Vr = 1 ;
2
3  i=0;
```

```
4
5         while(true)
6
7         {
8
9             if(Vs1.read() *3.3 < 1.47 )
10
11            {
12            {
13
14                i=(i+1)%4;
15
16                V1 = tab_V1[i] ;
17
18                V2 = tab_V2[i] ;
19
20                thread_sleep_for(WAIT_TIME);
21
22            }
23            }
24
25        else if(Vs1.read() *3.3 > 1.47)
26
27        {
28        {
29
30            i=(i-1)%4;
31
32            if (i<0){
33
34                i=3;
35
36            }
37
38            V1 = tab_V1[i] ;
39
40            V2 = tab_V2[i] ;
41
42            thread_sleep_for(WAIT_TIME);
43        }
44    }
```

Dans cette deuxième partie du programme on bascule dans le deuxième mode en activant le relais, ‘Vr’ est d’abord activé en étant mis à 1. Le programme entre ensuite dans une boucle infinie, où il ajuste les sorties ‘V1’ et ‘V2’ basées sur la lecture analogique de ‘Vs1’. La tension de ‘Vs1’ est comparée à 1.47 volts : si elle est inférieure, l’index ‘i’ est incrémenté cycliquement entre 0 et 3 pour parcourir les valeurs dans ‘tabV1’ et ‘tabV2’ ce qui fait que notre moteur va tourner à droite, si elle est supérieure, ‘i’ est décrémenté (tourner dans l’autre sens) avec une condition pour maintenir ‘i’ dans la plage de 0 à 3. À chaque itération, un délai fixe est

respecté grâce à $thread.sleepfor(WAITIME)$, ce qui permet de réguler la fréquence de mise à jour des sorties. Dans ce deuxième mode le moteur va suivre la lumière juste en comparant la valeur de Vs1 à 1.47.

8 Bibliographie

- (a) <https://resources.altium.com/fr/p/which-type-opto-isolator-right-your-signal>
- (b) prendre une photo du Rapport de projet TEE d'année précédente.
- (c) <https://www.youtube.com/watch?v=bm1FiMcINakt=913s>
- (d) <https://www.youtube.com/watch?v=LGFNh-JwpEkt=1s>
- (e) <https://www.youtube.com/watch?v=QcQnvve5R3s>

9 Anexes

```
1 #include "mbed.h"
2 #define WAIT_TIME 200//msec
3
4 DigitalOut V1(D9);
5 DigitalOut V2(D4);
6 DigitalOut Vr(D7) ;
7
8
9 DigitalIn relais_control(USER_BUTTON);
10 int tab_V1[4] = {1 , 1, 0, 0} ;
11 int tab_V2[4] = {0 , 1, 1, 0} ;
12 int tab_V1G[4] = {0 , 1, 1, 0} ;
13 int tab_V2G[4] = {0 , 0, 1, 1} ;
14
15 int n = 200 ;
16 int i,j , p=0 , U, V, m =100000 ,index=0 ;
17 float maxx = 3.3 , maxx1 , minn , minn2;
18 float current_value;
19
20
21 AnalogIn Vs1(A1);
22 AnalogIn Vs2(A0);
23
24
25
26
27 int main()
28 {
29     while (index==0)
30     {
31         Vr = 0 ;
32         printf("hello");
33
34
35
36         maxx = 0;
37         for (i = 0; i < n; i++)
38         {
39             printf(" i = %d \n" , i);
40             V1 = tab_V1[i%4] ;
41             V2 = tab_V2[i%4] ;
42
43             current_value= Vs1.read()*3.3;
44             if ( current_value > maxx)
45             {
46                 maxx = current_value ;
47             }
48         }
49     }
50 }
```

```
48         p = i ;
49
50         U=i%4;
51         printf("%f    maxx, p= %d \n", maxx,p) ;
52     }
53
54
55
56     printf("a0=%f a1=%f", Vs1.read()*3.3, Vs2.read()*3.3);
57
58     thread_sleep_for(WAIT_TIME);
59
60 }
61
62
63
64
65     printf("on est sorti de la boucle\n") ;
66     printf(" p = %d \n" , p) ;
67     for (j =i; j>= p ; j--)
68     {
69         printf("je suis rentr \n") ;
70
71
72         V1 = tab_V1[(j+U)%4] ;
73         V2 = tab_V2[(U+j)%4] ;
74         thread_sleep_for(WAIT_TIME);
75     }
76     printf("je suis sorti de la deuxieme boucle\n")      ;
77     index=index+1;
78
79     Vr = 1 ;
80     while(true)
81     {
82
83
84
85     if(Vs1.read() *3.3 > 1.67 )
86     {
87         for (i = 0; i < n; i++)
88     {
89         printf("je tourne a droite \n") ;
90         printf("value final = %f \n" ,Vs1.read() *3.3 ) ;
91
92         V1 = tab_V1G[i%4] ;
93         V2 = tab_V2G[i%4] ;
94         thread_sleep_for(WAIT_TIME);
95
96 }
```

```
97
98      }
99      }
100     if(Vs1.read() *3.3 < 1.67)
101    {
102        for (i = 0; i < n; i++)
103    {
104        printf("je tourne a gauche \n") ;
105        printf("value final = %f \n" ,Vs1.read() *3.3 ) ;
106
107
108        V1 = tab_V1[i%4] ;
109        V2 = tab_V2[i%4] ;
110        thread_sleep_for(WAIT_TIME);
111    }
112    }
113
114
115
116    }
117
118 }
119
120 }
```

