

Rapport de présentation de projet :

Groupe Électrogène hybride Autoporté (Photovoltaïque + Batteries)

Dans un grand nombre d'applications, il peut être nécessaire d'utiliser des outils et dispositifs électriques dans des lieux où il n'y a pas d'électricité (réseau 230 V-50 Hz).

Au-delà de quelques dizaines de mètres, l'usage de rallonges est proscrit et on doit utiliser des groupes électrogènes, la plupart du temps basés sur un moteur thermique polluant et bruyant associé à un alternateur.

L'objectif de ce projet, regroupant 7 personnes, est de développer un groupe électrogène autoporté (GEA) à partir d'un panneau photovoltaïque (PV) et d'un pack de batteries. Ces deux sources d'énergie embarquées seront associées pour alimenter un onduleur et recréer un réseau alternatif local utilisable de jour comme de nuit.

La sortie du GEA sera une tension 230 V-50 Hz obtenue à partir d'un onduleur du commerce.

Ce GEA étant mobile et pouvant être déplacé sur le lieu où on en a besoin, il sera motorisé et piloté par l'opérateur au moyen d'une commande radio.

En position statique, un dispositif permettra de toujours orienter le PV face au soleil pour faciliter la recharge des batteries et/ou contribuer à alimenter les appareils raccordés au GEA.



Nous travaillons plus précisément sur les deux parties suivantes du projet :

➤ **TOU5 :**

Motorisation du dispositif (Moteur à Courant Continu)

➤ **TOU7 :**

Contrôle de l'orientation du panneau photovoltaïque

« face au soleil »

Objectifs du projet

Notre rôle dans dans le projet général est de contrôler la motorisation du Groupe Électrogène Autoporté (GEA) lors d'un déplacement et d'ajuster son orientation en fonction de la position du soleil.

Un GEA est souvent utilisé pour fournir de l'électricité temporairement dans des endroits où le réseau électrique principal n'est pas disponible. Cela peut inclure des chantiers de construction, des événements en plein air ou d'autres besoins plus spécifiques.

Notre GEA est conçu pour être mobile et peut être utilisé dans des applications nécessitant une source d'énergie mobile.

Le deuxième objectif de notre projet est d'orienter le panneau photovoltaïque pour qu'il soit toujours dirigé face au soleil, ce qui permet de garantir une production d'énergie optimale.

Remerciements

*Nous tenons tout particulièrement à remercier notre professeur encadrant **Pierre Toussaint** pour son accompagnement et ses conseils tout au long de notre projet.*

Nous sommes très reconnaissants du temps qu'il nous a accordé et des connaissances qu'il nous a partagées car il a grandement enrichi notre apprentissage pendant cette année.

*Nous remercions également notre professeur auditeur **Christian Josse** pour son attention lors de nos présentations.*

Nous remercions enfin toutes les personnes avec lesquelles nous avons collaboré et eu le plaisir d'échanger lors des différentes séances.

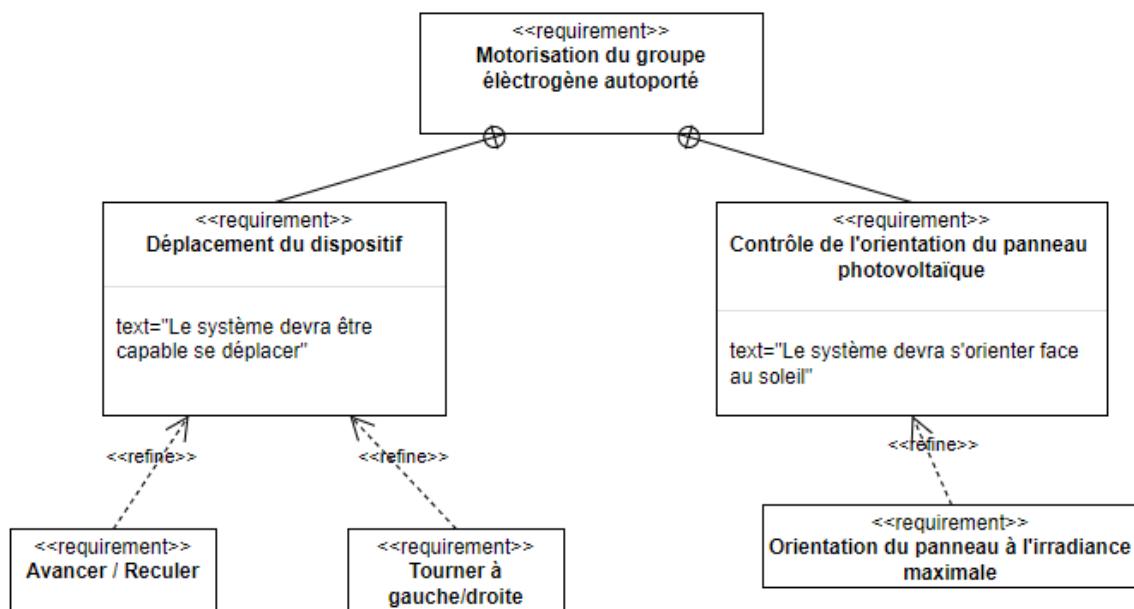
Table des matières :

1. Cahier des charges général du projet.....	5
a. Diagramme des exigences.....	5
b. Diagramme de Gantt (dernière mise à jour le 02/04/2024).....	5
2. Motorisation du dispositif (Moteur à Courant Continu).....	5
a. Détermination des caractéristiques de notre Moteur à Courant Continu (MCC).....	6
i. Détermination du rapport de réduction du réducteur.....	6
ii. Détermination de la vitesse de rotation du moteur.....	6
iii. Détermination de la résistance interne R du moteur.....	7
iv. Détermination du courant nominal I.....	7
v. Détermination de la constante électromagnétique K.....	8
vi. Détermination de l'inductance mutuelle.....	8
b. Construction et simulation de notre schéma électrique sur PSIM.....	8
c. Implémentation du pont en H de type L298N.....	10
d. Récupération du PCB et soudures des composants.....	17
e. Tests de vérifications du PCB.....	17
f. Premier prototype : Contrôle d'un moteur.....	19
g. Second prototype : Contrôle de deux moteurs.....	22
h. Etude de la télécommande : Transmitter and Receiver Connection.....	26
3. Contrôle de l'orientation du panneau photovoltaïque “face au soleil”	33
a. Pistes de recherche et premières solutions.....	33
i. SFH 5711.....	33
ii. Photorésistance.....	34
iii. Panneau photovoltaïque.....	36
b. Réalisation du PCB de capteurs de luminosité.....	38
c. Essais des capteurs de luminosité.....	39
i. Essais en inclinaison.....	40
ii. Essais en fonction de l'irradiance.....	41
d. Utilisation du panneau photovoltaïque comme capteur de luminosité...	43
e. Recherche de l'irradiance maximale.....	43
i. Création d'un banc de tests.....	43
ii. Ajout d'un moteur à courant continu.....	45
iii. Construction d'un hacheur 4 quadrants et finalisation des essais....	47

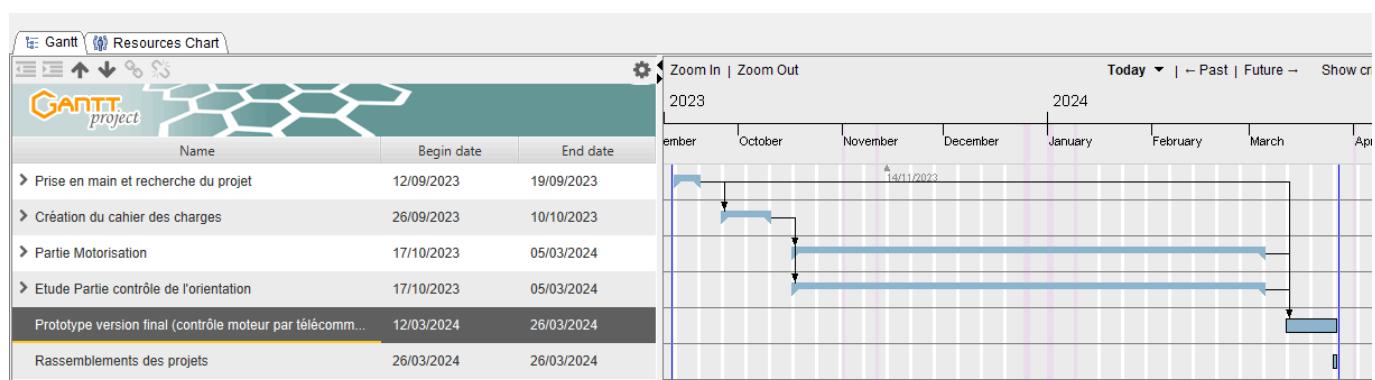
4. État d'avancement du projet.....	50
a. Le travail effectué.....	50
b. Les prochaines étapes.....	51

1. Cahier des charges général du projet

a. Diagramme des exigences



b. Diagramme de Gantt (dernière mise à jour le 02/04/2024)



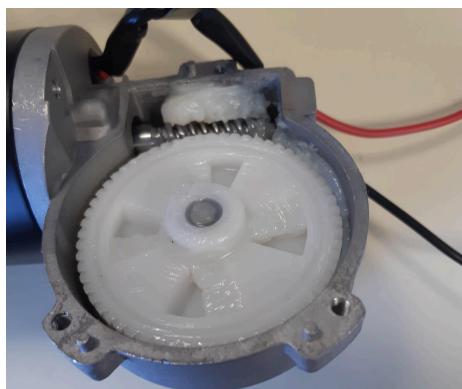
2. Motorisation du dispositif (Moteur à Courant Continu)

a. Détermination des caractéristiques de notre Moteur à Courant Continu (MCC)

Notre premier objectif est de déterminer toutes les caractéristiques de notre moteur à courant continu, afin de pouvoir simuler notre schéma électrique sur PSIM. Le schéma réalisé sur PSIM nous permettra par la suite d'avoir une bonne vue d'ensemble sur les objectifs que nous devrons atteindre.

i. Détermination du rapport de réduction du réducteur

Nous ouvrons notre Moto-réducteur comme représenté ci-dessous.



Nous remarquons que notre réducteur est composé d'une vis sans fin et d'une roue dentée.

On compte "à la main" le nombre de filets de la vis sans fin et le nombre de dents de la roue dentée. On en déduit ensuite le rapport de réduction R.

$$R = \frac{Zv}{Zr} = \frac{1}{60} = 0,05$$

Avec Zv le nombre de filets de la vis sans fin et Zr le nombre de dents dans la roue dentée.

ii. Détermination de la vitesse de rotation du moteur

D'après la datasheet de la MCC ci dessous, la vitesse nominale en sortie du motoréducteur est $N_s=220$ tr/min.



La relation entre le rapport de réduction et les vitesses de rotations en entrée et sortie du motoréducteur est :

$$R = \frac{N_m}{N_s}$$

Avec, N_m la vitesse de rotation du moteur et N_s la vitesse de rotation en sortie du réducteur.

AN : $N_m=220*0,05 = 4400$ tr/min. **Wm = 460 rad/s.**

iii. Détermination de la résistance interne R du moteur

Pour déterminer ce paramètre, nous branchons aux bornes du moteur un ohmètre et on exprime la valeur trouvée à R. On mesure alors **R=1 Ω**.

iv. Détermination du courant nominal I

On se place à vitesse nominale N_m : $W_m = 460$ rad/s, $U = 24V$.

Dans cet état, la puissance du moteur est à son maximum et est égale à 45W. On note cette puissance P_m .

On en déduit donc : $I = P_m / U = 45 / 24 = 1,875 \text{ A.}$

v. Détermination de la constante électromagnétique K

Nous déterminons la constante avec la formule suivante :

$$E = K * \omega_m \Rightarrow K = \frac{E}{\omega_m} = \frac{22,125}{460} = 0,048$$

vi. Détermination de l'inductance mutuelle

$$K = L_{af} * I_f \quad (\text{If : field current}) \Rightarrow I_f = (L_{af} W_m) / E$$

$$L_{af} = (U - I * R) / (I_f W_m)$$

$$\text{Alors : } L_{af} = [(U - I * R) * E] / (W_m^2 L_{af})$$

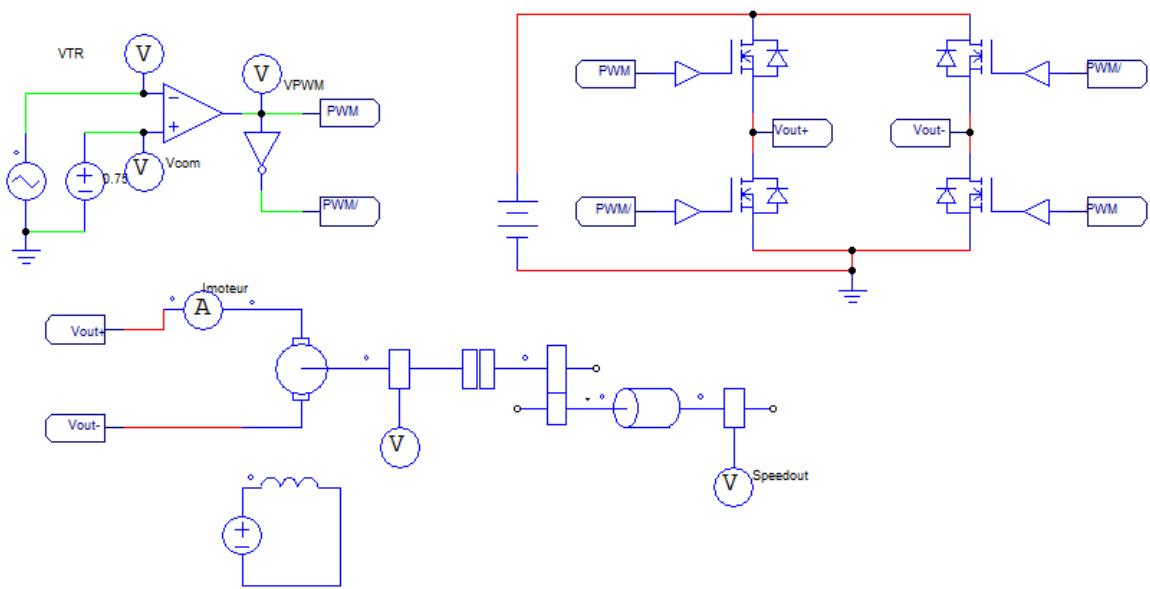
$$\text{On en déduit } L_{af}^2 = [(U - I * R) * E] / W_m^2$$

Ainsi, **Laf = 0,048 H.**

Courant mutuelle : **If = K/Laf = 1**

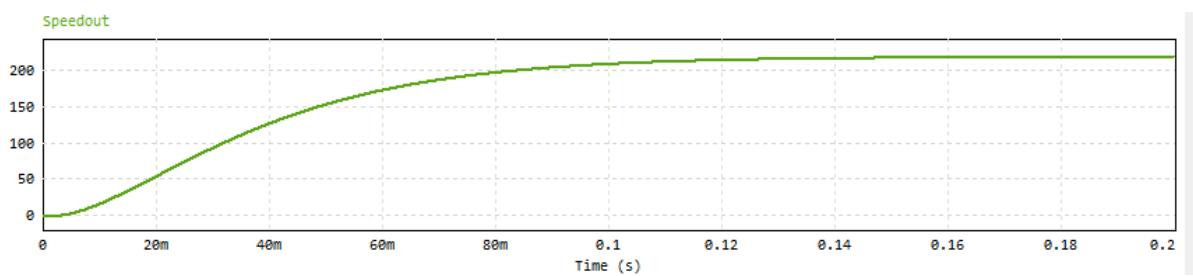
b. Construction et simulation de notre schéma électrique sur PSIM

Nous pouvons désormais construire notre schéma électrique sur PSIM à l'aide des caractéristiques du MCC déterminées précédemment.

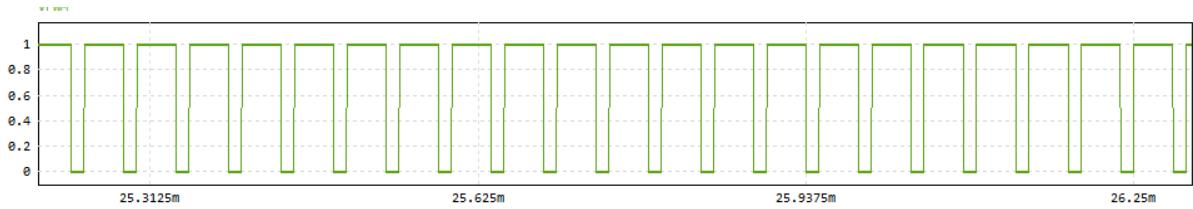


Simulations:

On simule la vitesse de rotation du moteur en sortie du moto-réducteur.



On simule également les signaux PWM qui permettent de contrôler les transistors du hacheur 4 quadrants.



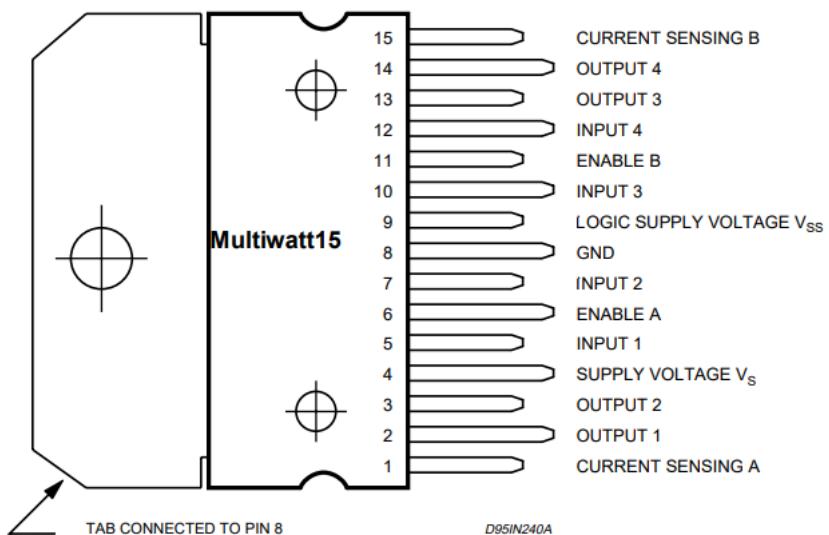
Nos signaux PWM sont bien carrés, ils ont donc été bien construits sur PSIM. Ces signaux seront générés par un microcontrôleur.

c. Implémentation du pont en H de type L298N

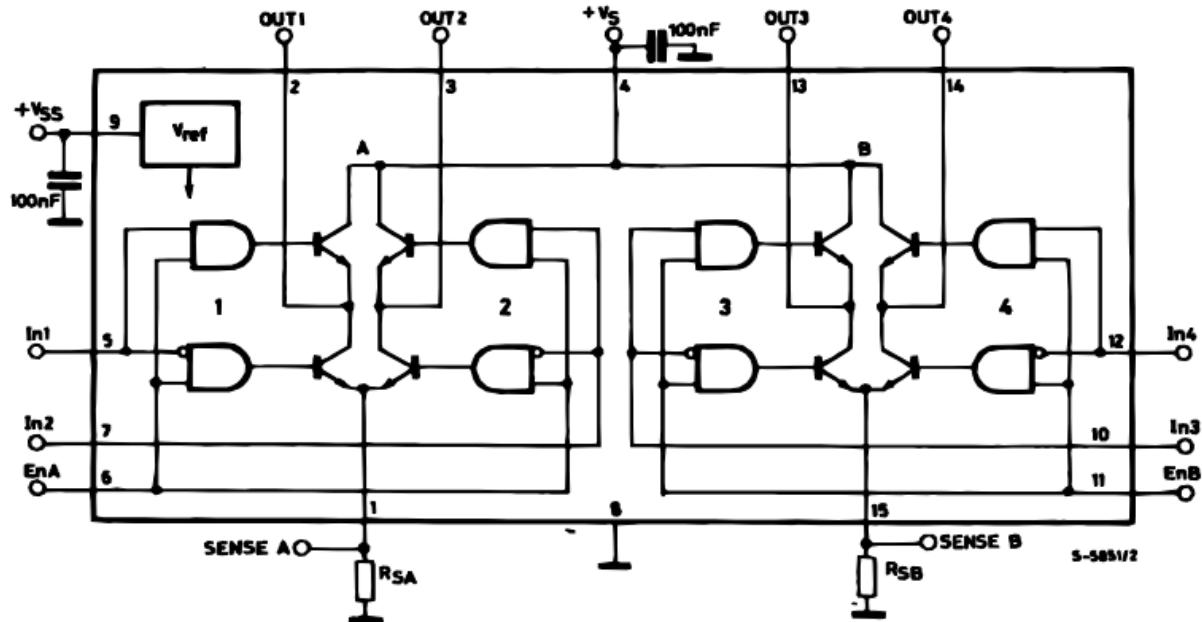
Notre exigence à respecter est claire : nous devons réussir à contrôler la direction du dispositif. Alors, pour cette raison et pour des raisons de coûts, on opte pour l'implémentation d'un pont en H en utilisant un driver L298N.

On étudie donc la datasheet afin de construire un PCB qui nous permettra de relier tous les composants entre eux.

Voici un extrait de la datasheet du driver essentiels à la création du PCB :



Notre driver contient plusieurs entrées et sorties. Il nous faut identifier chaque entrée afin de savoir à quoi ils servent dans notre pont en H.



PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V _s	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	V _{SS}	Supply Voltage for the Logic Blocks. A 100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
-	3;18	N.C.	Not Connected

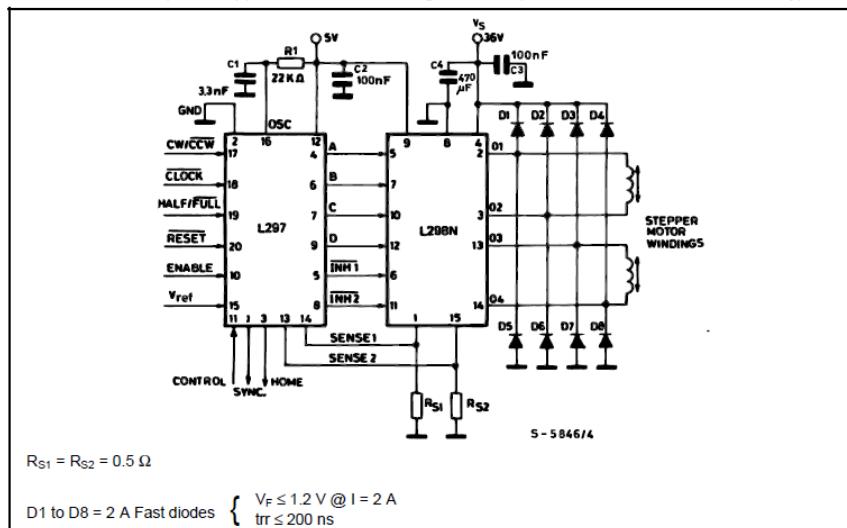
A l'aide de l'extrait du schéma issu de la datasheet ci-dessus, on identifie chaque entrée et sa fonctionnalité :

- Sense A et SenseB nous permettent à l'aide des résistances Rsa et Rsb de contrôler le courant de charge.
- Out1 et Out2 les sorties du moteurs 1.
- Out3 et Out4 sont les sorties du moteur 2.
- Vs est la tension d'alimentation des moteurs (24V dans notre cas).
- Intput1 et Intput2 sont les signaux de type PWM qui nous permettent de contrôler le rapport cyclique du moteur 1.
- Intput 3 et Intput4 sont les signaux de type PWM qui nous permettent de contrôler le rapport cyclique du moteur 2.
- Enable 1 et Enable 2 sont des signaux pour valider ou non le contrôle du moteur.
- Vss est l'alimentation des signaux logiques (les signaux PWM), elle est égale à 5V.
- GND est la terre.

On remarque également que la datasheet contient des exemples d'implémentation d'un hacheur 4 quadrants :

Figure 8 : Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.

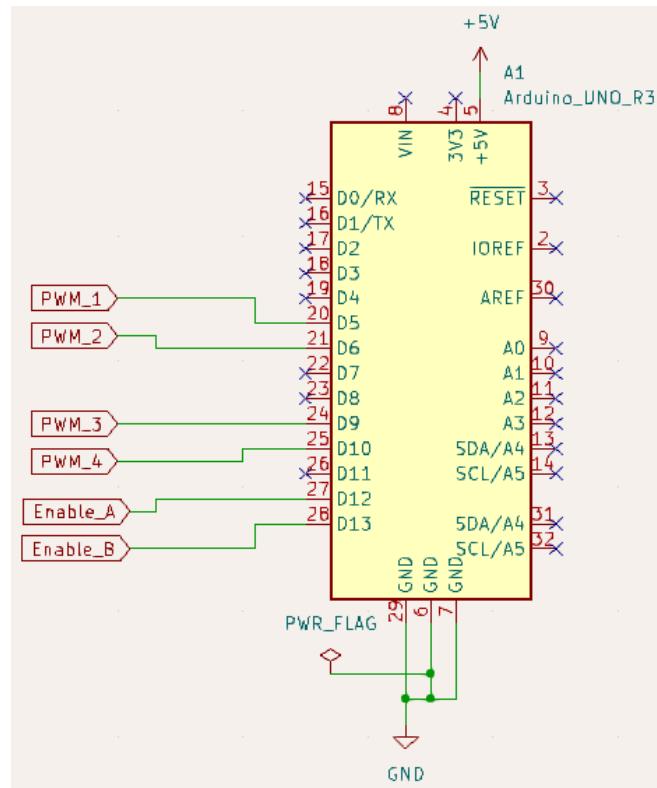
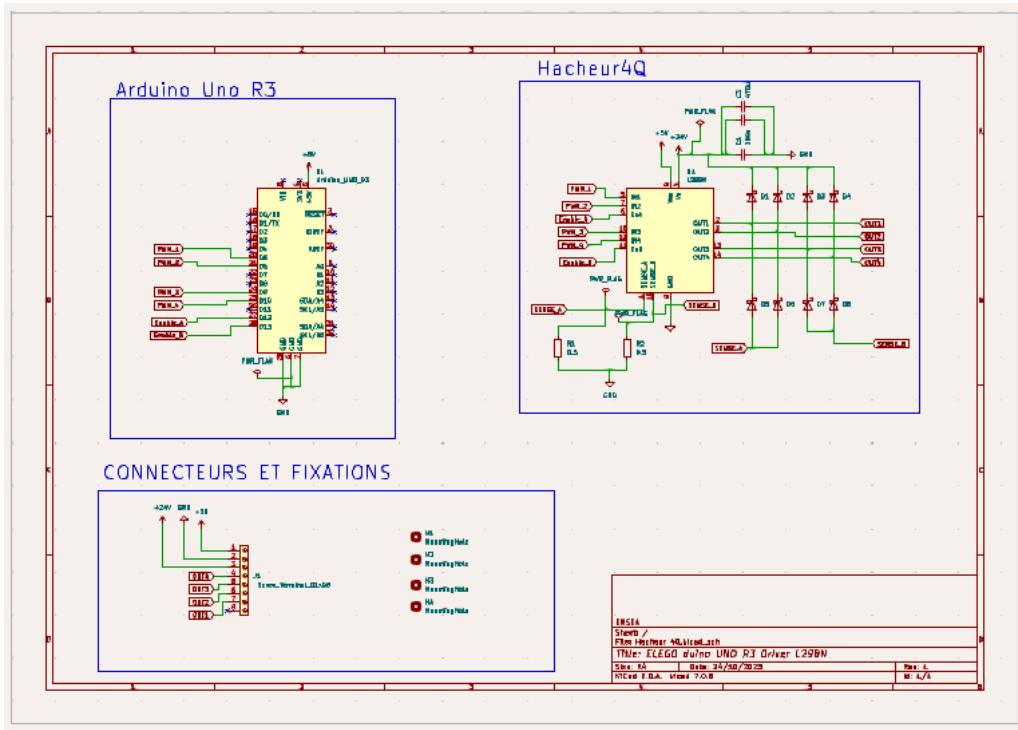


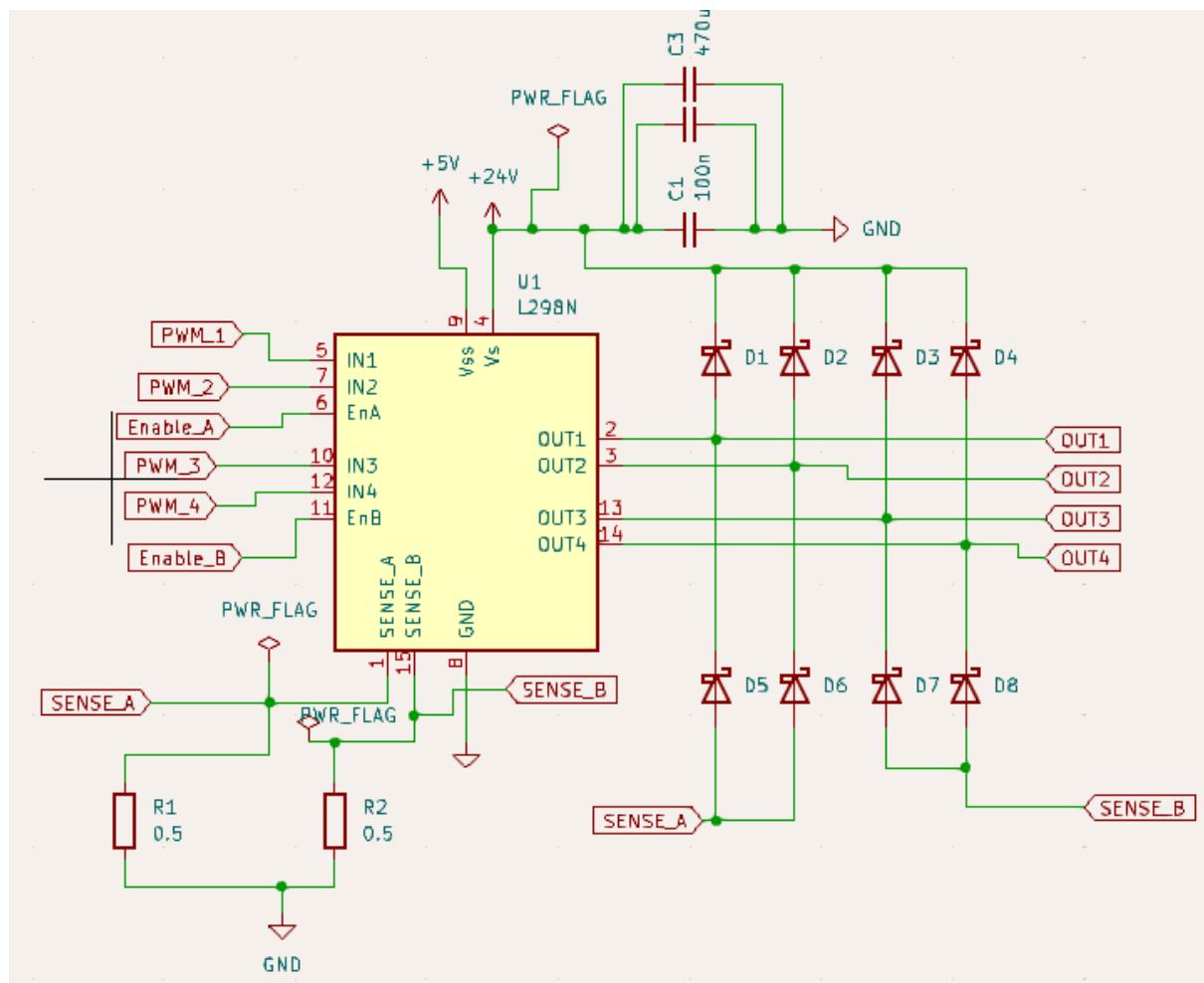
Connaissant toutes les fonctionnalités du driver, on peut construire le schéma électrique sur KICAD afin de faire par la suite son PCB.

Notre schéma est composé de 3 parties : Le micro-contrôleur (Carte arduino R3), hacheur 4 quadrants (Driver L298N + diodes), bornes et des trous de fixation.
On veille bien à ce que le schéma respecte la datasheet et les caractéristiques de notre moteur.

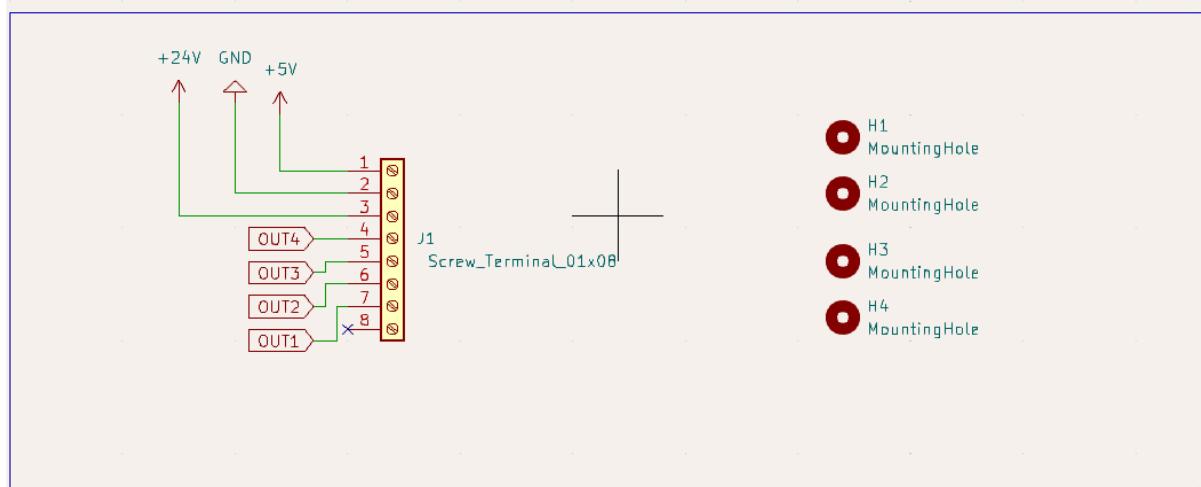
On utilise des diodes Schottky de façon à avoir des commutations rapides, afin d'avoir un temps de recouvrement le plus petit possible.

Notre courant nominal de notre moteur est de presque 2A (calcul fait précédemment), on prend donc (comme sur l'exemple de la datasheet) R_{sa} = R_{sb} = 0,5 Ω.

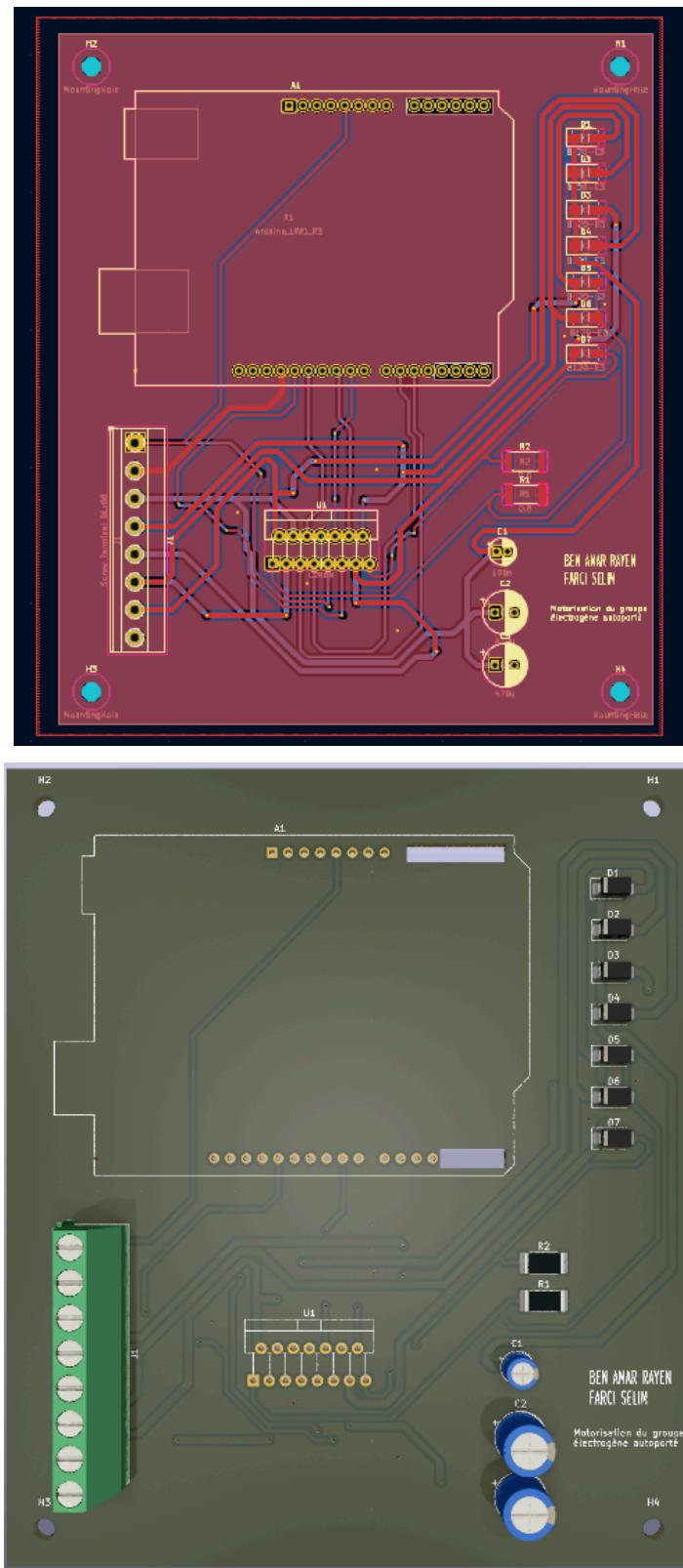




CONNECTEURS ET FIXATIONS



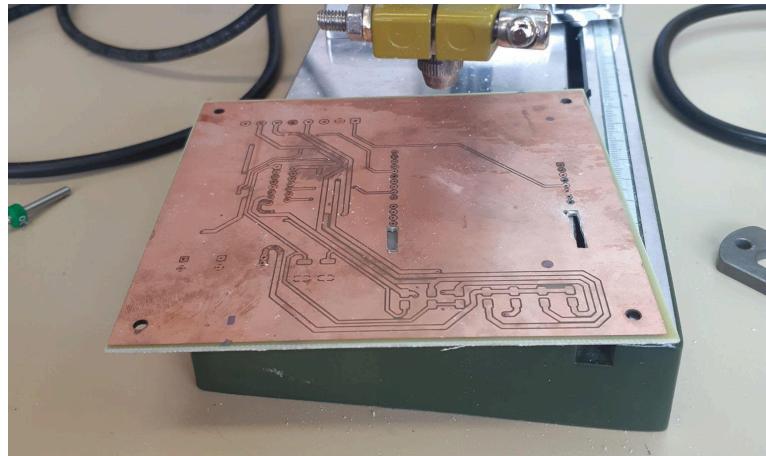
Nous pouvons maintenant effectuer le routage du PCB :



On peut maintenant envoyer le PCB à l'impression.
Le PCB ressemblera au dernier schéma ci-dessus.

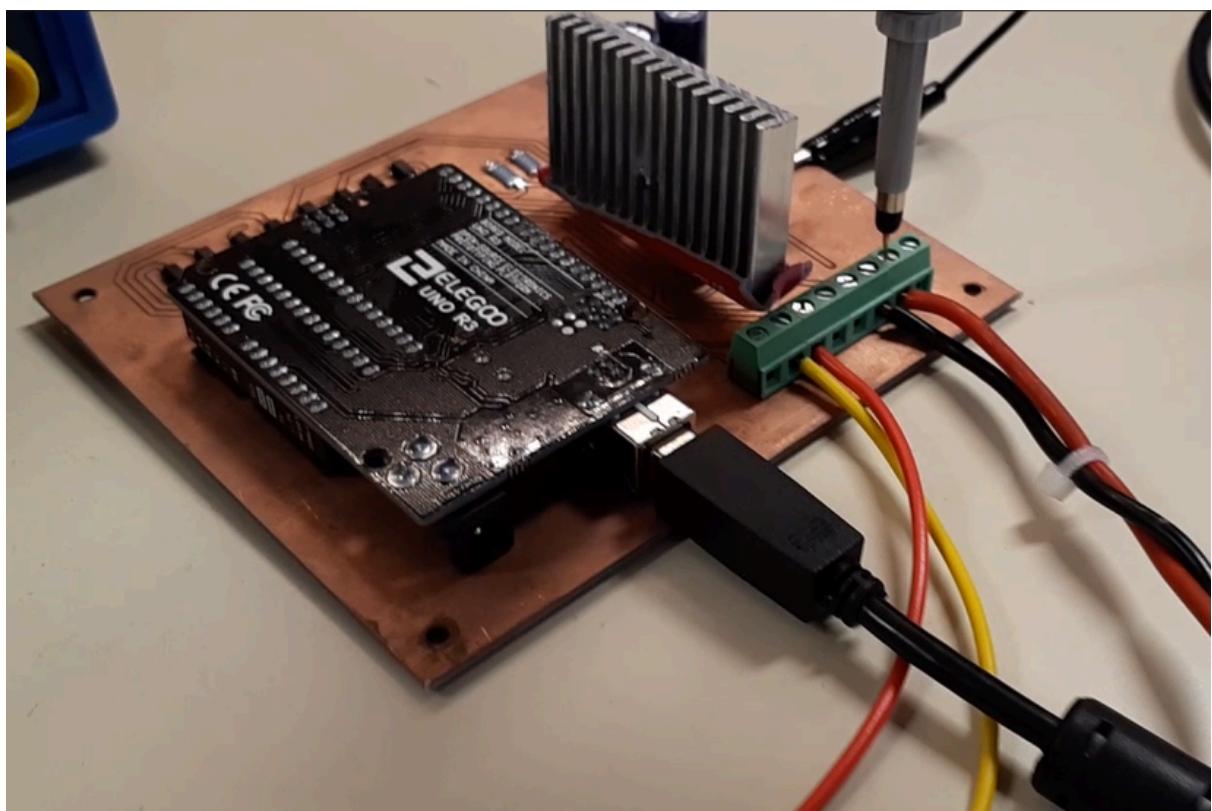
d. Récupération du PCB et soudage des composants

Nous avons récupéré le PCB après l'impression à l'ENSEA :



Nous soudons donc les composants nécessaires. Ensuite, nous pourrons tester si la carte fonctionne correctement.

e. Tests de vérifications du PCB



Avant de tester notre PCB, il nous faut d'abord programmer le microcontrôleur afin de générer des PWMs.

```
CLKPR = (1 << CLKPCE);
CLKPR = (1 << CLKPS0);

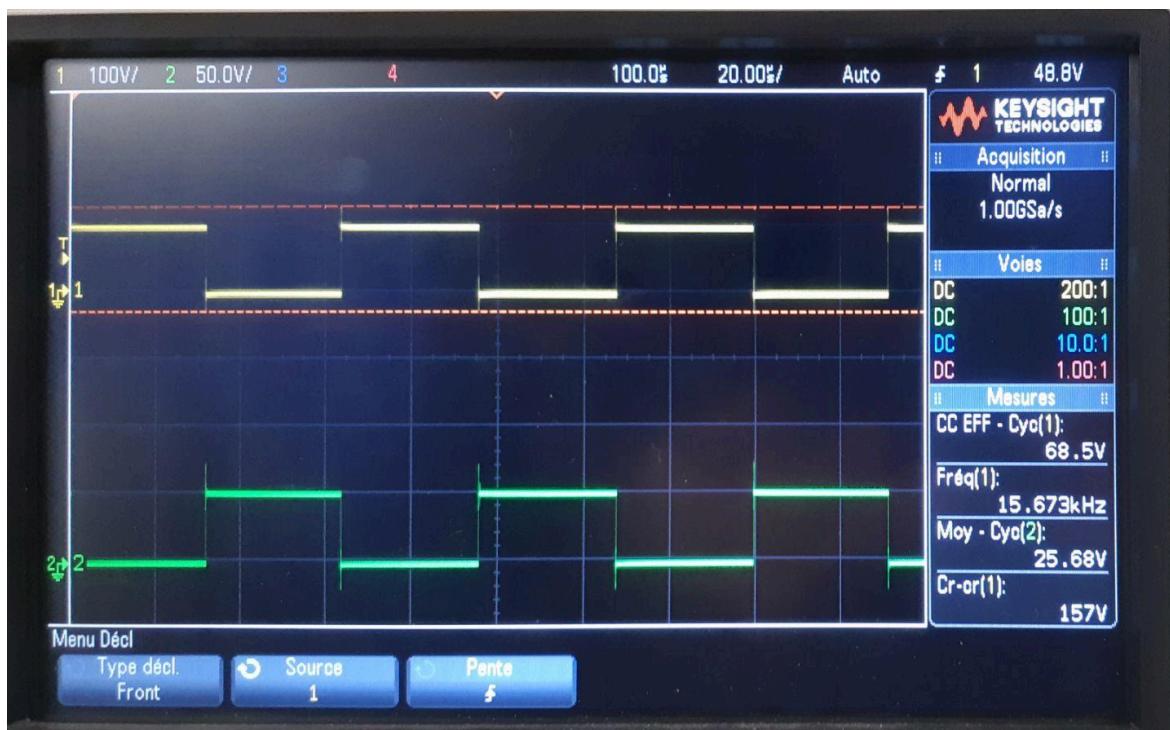
// Autoriser les modifications des registres de division de fréquence
CLKPR &= ~(1 << CLKPCE);

TCCR0A= 0b10110001;
TCCR0B= 0b00000001;
TCCR1A= 0b10110001; // Pin 9
TCCR1B= 0b00000001; // Pin 10

pinMode(5, OUTPUT); // Pin PWMA
pinMode(6, OUTPUT); // PIN PWMB=PWMA/
pinMode(9,OUTPUT);
pinMode(10,OUTPUT);

OCR0A = 127;
OCR0B = OCR0A;
OCR1A = 127;
OCR1B = OCR1A;
```

Voici un exemple des PWM générés pour un rapport cyclique de 50%.



Le micro-contrôleur génère bien les PWM attendues.

Pour tester notre carte, nous allons envoyer les sources de tension d'alimentations, c'est-à-dire la tension de commande 5V et Moteur 10V .

Nous avons d'abord testé avec une tension moteurs de 10V afin de ne pas alimenter les moteurs directement par leurs tensions nominales.

Une fois le test validé, nous pouvons refaire le test avec leurs tensions nominales.

Vérifions maintenant si les signaux reçus par le moteur correspondent bien au rapport cyclique défini (25%).



Sortie moteur sur la borne + sous 24V.

Le rapport cyclique est bien de 25%.

Le moteur reçoit donc bien le signal qu'il devrait recevoir.

Nous effectuons le même test avec l'autre moteur et cela correspond bien aux attentes.

f. Premier prototype : Contrôle d'un moteur

Nous faisons maintenant un premier prototype avec le contrôle d'un moteur pour vérifier que tout fonctionne correctement.

Nous testerons les deux ensemble par la suite.

On code la carte de la façon suivante pour le premier prototype :

```
void setup() {
    // frequency scalar parameter
    CLKPR = (1 << CLKPCE);
    CLKPR = (1 << CLKPS0);

    // Autoriser les modifications des registres de division de fréquence
    CLKPR &= ~(1 << CLKPCE);

    // Timer parameter for OUTPUT PIN 5 and 6 and 9 and 10
    TCCR0A = 0b10110001; // pin 5
    TCCR0B = 0b00000001; // Pin 6
    TCCR1A = 0b10110001; // Pin 9
    TCCR1B = 0b00000001; // Pin 10

    // PWM OUTPUT
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(9, OUTPUT);
    pinMode(10, OUTPUT);

    // Buttons INPUT
    pinMode(1, INPUT);
    pinMode(2, INPUT);
    pinMode(3, INPUT);
    pinMode(4, INPUT);

    // Enable OUTPUT
    pinMode(12, OUTPUT);
    pinMode(13, OUTPUT);

    // State buttons
    digitalWrite(1, HIGH); // Forward
    forward = digitalRead(1) == HIGH

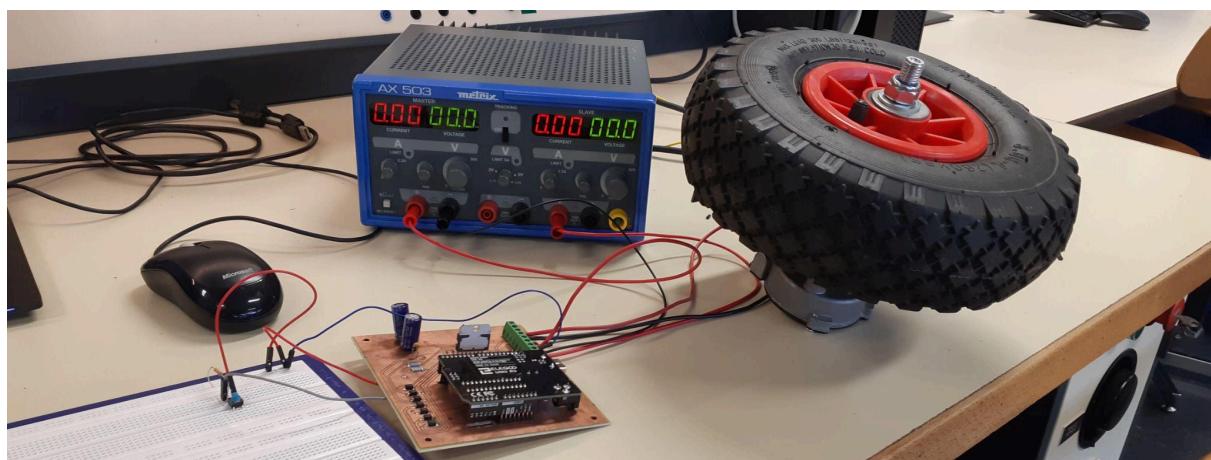
    //ENABLE A AND B
    digitalWrite(12, HIGH); // Activation du Hacheur 4Q Moteur 1
    digitalWrite(13, HIGH); //Activation du hacheur 4Q Moteur 2
```

```

if (forward) { // if the turn forward button is HIGH
    OCR0A = 191;
    OCR0B = OCR0A;
    OCR1A = 63;
    OCR1B = OCR1A;
    delay(3000);
    OCR0A = 127;
    OCR0B = OCR0A;
    OCR1A = 127;
    OCR1B = OCR1A;
}
else {
    OCR0A = 127; // If none button is pressed
    OCR0B = 127;
    OCR1A = 127;
    OCR1B = 127;
}

```

Voici une photo du prototype :



Le générateur est un générateur de 24V, le 5 V provient directement de l'ordinateur pour le moment.

Le prototype fonctionne très bien comme on le souhaite.

g. Second prototype : Contrôle de deux moteurs

Nous allons maintenant améliorer notre prototype afin de commander les deux moteurs en même temps.

Ce prototype aura donc 4 modes : “aller tout droit”, “reculer”, “tourner à gauche” et “tourner à droite”.

On modifie donc notre programme arduino afin de répondre à cette exigence.

Voici le code utilisés :

```
bool forward;
bool back;
bool turnright;
bool turnleft;
byte direction = 0b000;

void setup() {
    //Button INPUT
    pinMode(2,INPUT); // mettre le moteur dans un sens
    pinMode(3,INPUT);
    pinMode(4,INPUT);
    pinMode(7,INPUT);
```

On utilise des booléens afin de simuler l’appuie sur les boutons et on utilise 3 bits afin de conclure la direction dans laquelle il faut aller. (Attention, 3 bits car il y a pas 4 modes mais 5, en effet il faut prendre en compte le cas où aucun bouton n'est enfoncé, soit le cas où le moteur est à l'arrêt).

```
// Enable A and B
pinMode(12,OUTPUT);
pinMode(13,OUTPUT);

digitalWrite(2,LOW); // forward
digitalWrite(3,LOW); // back
digitalWrite(4,LOW); // turn right
digitalWrite(7,LOW); // turn left

digitalWrite(12,HIGH) ; // Activation du Hacheur 4Q Moteur 1
digitalWrite(13,HIGH); //Activation du hacheur 4Q Moteur 2
```

```

CLKPR = (1 << CLKPCE);
CLKPR = (1 << CLKPS0);

// Autoriser les modifications des registres de division de fréquence
CLKPR &= ~(1 << CLKPCE);

TCCR0A= 0b10110001;
TCCR0B= 0b00000001;
TCCR1A= 0b10110001; // Pin 9
TCCR1B= 0b00000001; // Pin 10

pinMode(5, OUTPUT); // Pin PWMA
pinMode(6, OUTPUT); // PIN PWMB=PWMA/
pinMode(9,OUTPUT);
pinMode(10,OUTPUT);

void loop() {
    // Buttons state

    forward = digitalRead(2);
    back = digitalRead(3);
    turnright = digitalRead(4);
    turnleft = digitalRead(7);

    if (forward == true) { // Si on appuie sur forward
        direction = 0b000;
    } else if (back == true) { // Si on appuie sur back
        direction = 0b001;
    } else if (turnright == true) { // Si on appuie sur tourner à droite
        direction = 0b010;
    } else if (turnleft == true) { // Si on appuie sur tourner à gauche
        direction = 0b011;
    } else {
        direction = 0b100; // Par défaut, si aucun bouton n'est enfoncé, direction est stop
    }
}

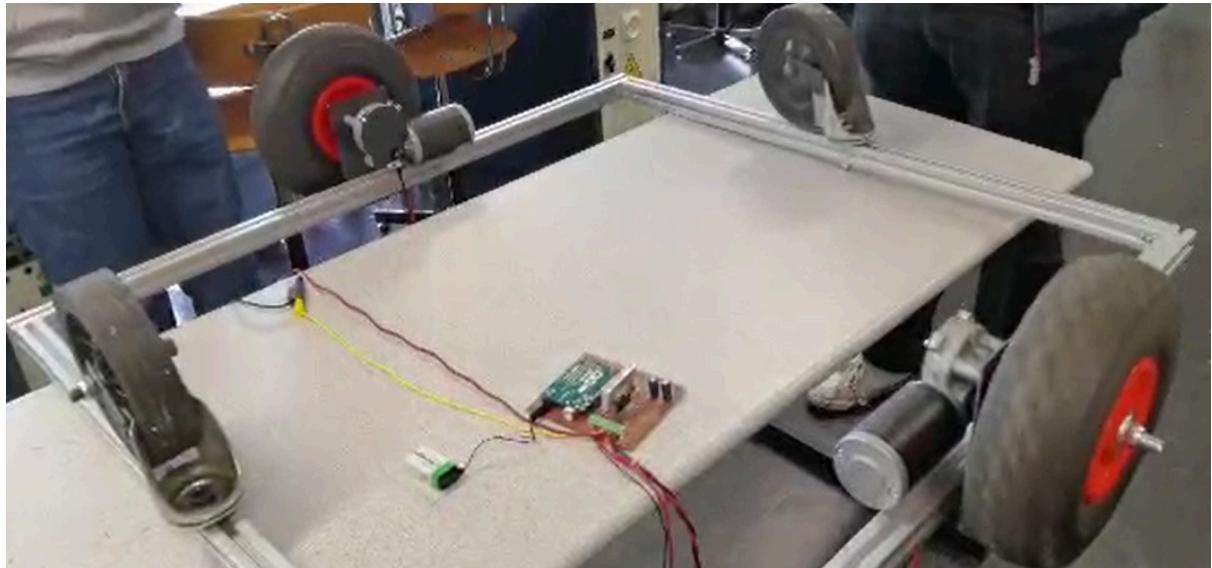
```

```

switch (direction) {
    case 0b000: // Si seulement forward est vrai
        OCR0A = 63;
        OCR0B = OCR0A;
        OCR1A = 63;
        OCR1B = OCR1A;
        break;
    case 0b001: // si seulement back est vrai
        OCR0A = 190;
        OCR0B = OCR0A;
        OCR1A = 190;
        OCR1B = OCR1A;
        break;
    case 0b010: // si seulement turnright est vrai
        OCR0A = 63;
        OCR0B = OCR0A;
        OCR1A = 190;
        OCR1B = OCR1A;
        break;
    case 0b011: // Si seulement turnleft est vrai
        OCR0A = 190;
        OCR0B = OCR0A;
        OCR1A = 63;
        OCR1B = OCR1A;
        break;
    case 0b100: // Si aucun n'est vrai, arrêt
        OCR0A = 127;
        OCR0B = 127;
        OCR1A = 127;
        OCR1B = 127;
        break;
}
}

```

On construit donc notre deuxième prototype. Voici une image de celui ci :



Prototype version 2

Notre prototype fonctionne très bien.

On notera que lors de la construction de ce prototype nous avons eu un incident car nous avons mal branché un câble et cela a créé un court-circuit, que nous avons réussi à réparer fort heureusement.

Ce court-circuit n'a pas eu de conséquences significatives sur le projet.

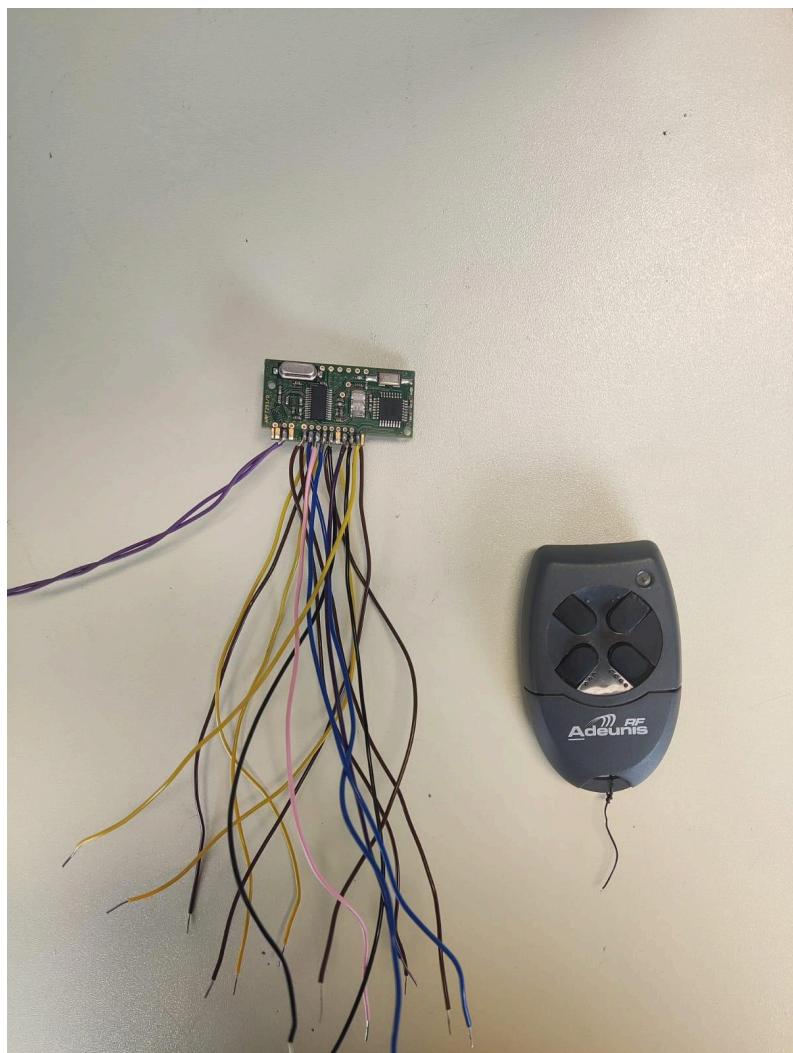
La prochaine étape est maintenant d'améliorer ce prototype en la télécommande qui commandera les valeurs binaires des bits "Forward", "Back", "Turn Left" et "Turn right".

h. Etude de la télécommande : Transmitter and Receiver Connection

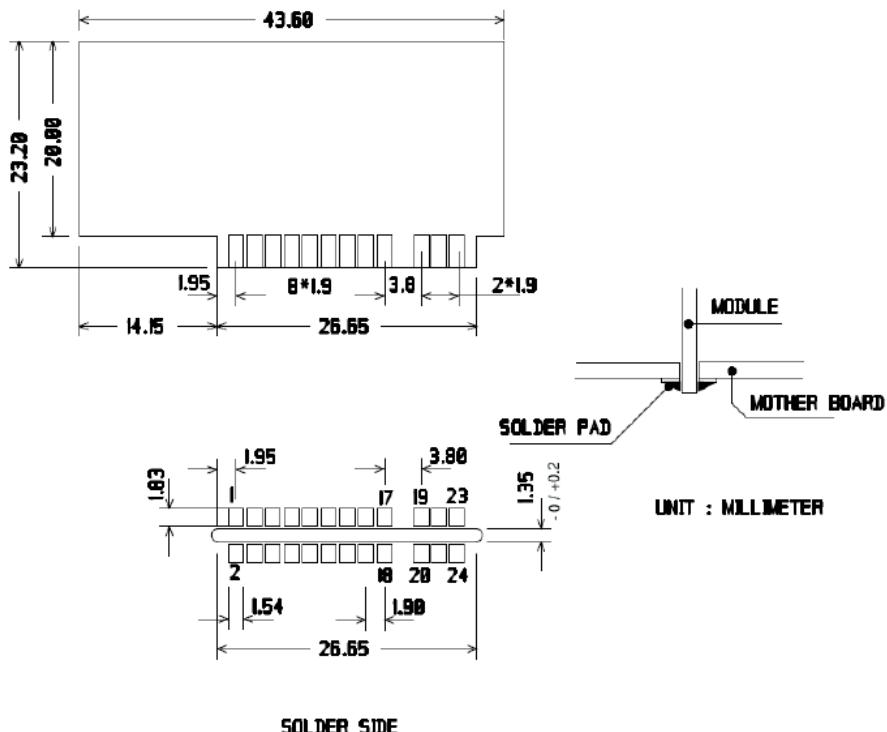
The goal was to program the RF Adeunis Transmitter and Receiver as per our need.

1. Button 1- Forward
2. Button 2- Right
3. Button 3- Back
4. Button 4- Left

We used a RF Adeunis ARF18 4-key type transmitter and receiver.



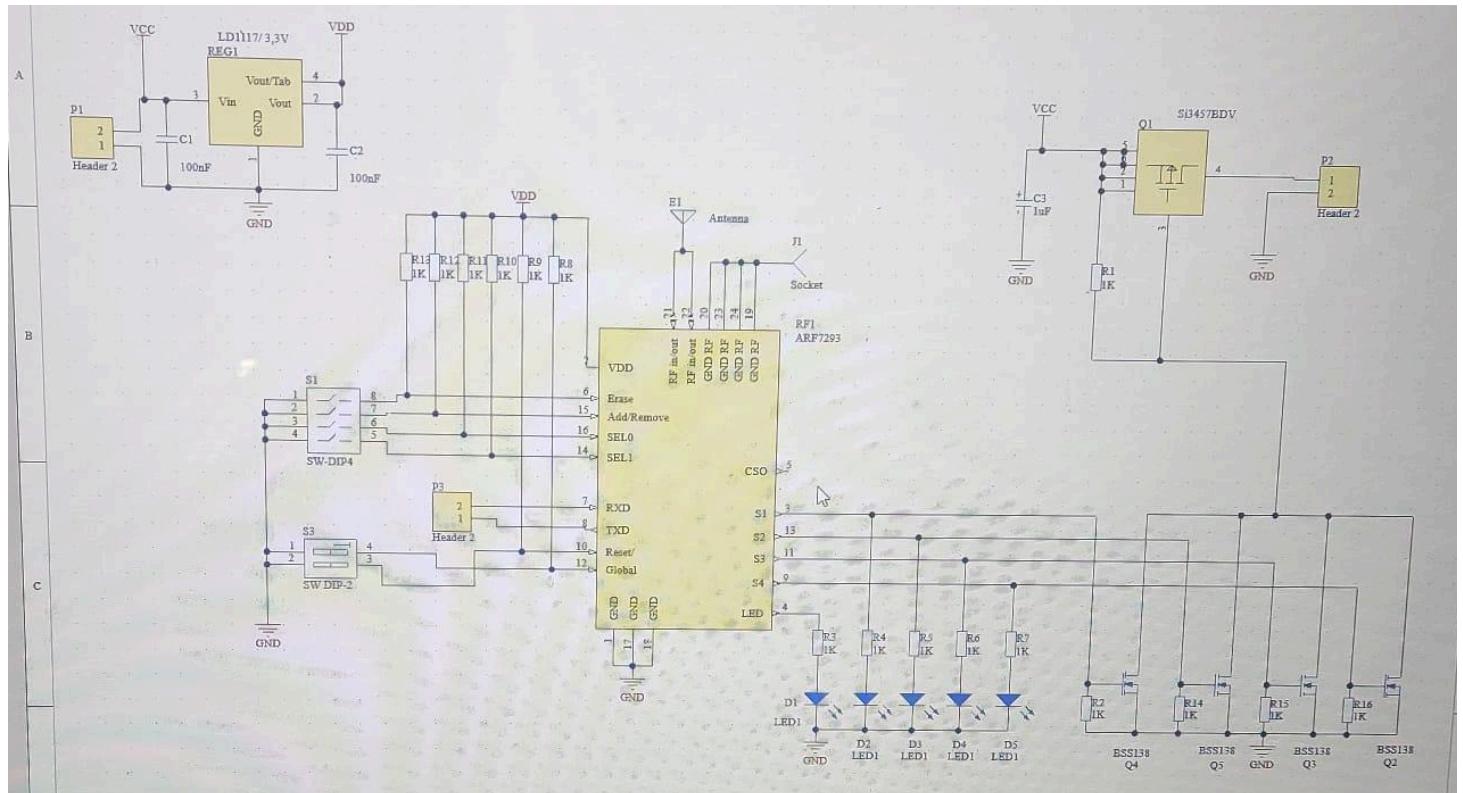
The receiver <ARF7293> has the following architecture and specifications:



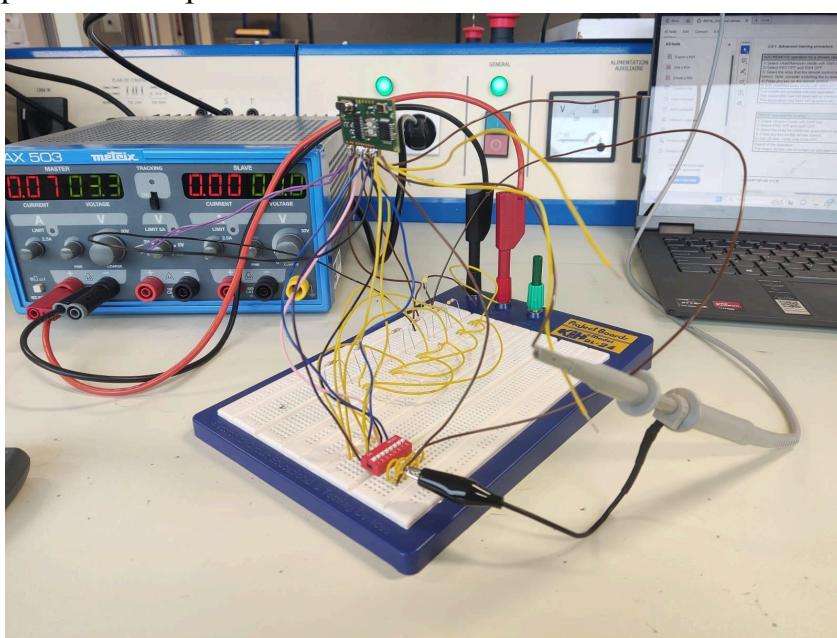
Pin interface	Name	I/O	Description
Digital interface			
1	GND		Ground
2	VDD		Voltage 3.3V +/-10%
3	Digital output 1	O	Digital output 1 (cf section 6 «Training procedures»)
4	LED	O	Light emitting diode (LED - cf section 6 «Training procedures»)
5	Current saving output	O	Can be used with relays. When a digital output 1 to 4 is set to «1» this output is set to «1» for 50 ms then returns to «0».
6	Erase	I	Erase (cf section 6 «Training procedures»)
7	RXD	I	Reception of serial data
8	TXD	O	Transmission of serial data
9	Digital output 4	O	Digital output 4 (cf section 6 «Training procedures»)
10	/RESET	I	Reinitialization of the transmitter-receiver hardware, LOW active. Can be disconnected.
11	Digital output 3	O	Digital output 3 (cf section 6 «Training procedures»)
12	Global	I	Global (cf section 6 «Training procedures»)
13	Digital output 2	O	Digital output 2 (cf section 6 «Training procedures»)
14	Sel1	I	Sel1 (cf section 6 «Training procedures»)
15	Add/Remove	I	Add/Remove (cf section 6 «Training procedures»)
16	Sel0	I	Sel0 (cf section 6 «Training procedures»)
17, 18	GND		Ground
RF interface			
19, 20	GND RF		RF aerial: ground
21, 22	RF in/out		RF aerial IN/OUT.
23, 24	GND RF		RF aerial: ground

Soldering was made to all of the pins for training the receiver. The process and code was as follows:

1. Making the connection with a pull up resistor to control the training modes of different switches based on requirement.



2. The system works on a 3.3V input. All the connections were made as per the picture in Step 1.



3. Removing any instructions that might have been stored in the receiver during the previous use. The erasing works in steps (OUTPUT 1>2>3>4), it's not compulsory but makes the work less confusing.

i. First is selecting the output we wish to operate first by connecting it to the button we wish for it to respond to. Example Output 1.

SW1	SW2	SW3	SW4	SW5
ON	ON	OFF	OFF	ON

ii. Once the switches are selected accordingly, pressing the button it should respond to will connect them.

Global ERASE operation
1/ Select «Erase» mode with SW4 ON. 2/ Select SW3 ON and SW5 OFF. 3/ Press any key on any valid remote control. 4/ Exit «Erase» mode with SW4 OFF.
Result of the operation: No relay can now be activated by any remote-control. The LED flashes twice.

iii. Selecting these switches prepares the receiver to be erased, by pressing the button assigned to the respective output (Output 1 -> Button 1).

4. Step number 3 is then followed for all of the outputs in a similar way for all of the buttons and outputs. By doing this we can erase all the previously stored instructions present in the Output pin after connecting them to their respective Button to which it shall respond.

- Key 1 on the remote control to Rx1
On receiver 1, configure the switches as shown below.

SW1	SW2	SW3	SW4	SW5
ON	ON	OFF	OFF	ON

Press key 1 on the remote control
Then select SW5 OFF
Switch SW1 and SW2 to OFF if monostable mode is required. Leave switch 2 on ON if bistable mode is required.

- Key 2 on the remote control to RX2
On receiver 2, configure the switches as shown below.

SW1	SW2	SW3	SW4	SW5
ON	ON	OFF	OFF	ON

Press key 2 on the remote control
Select SW5 OFF
Switch SW1 and SW2 to OFF if monostable mode is required. Leave switch 2 on ON if bistable mode is required.

- Key 3 on the remote control to RX3
On receiver 2, configure the switches as shown below.

SW1	SW2	SW3	SW4	SW5
ON	ON	OFF	OFF	ON

Press key 3 on the remote control
Select SW5 OFF
Switch SW1 and SW2 to OFF if monostable mode is required. Leave switch 2 on ON if bistable mode is required.

- Key 4 on the remote control to RX4
On receiver 2, configure the switches as shown below.

SW1	SW2	SW3	SW4	SW5
ON	ON	OFF	OFF	ON

Press key 4 on the remote control
Select SW5 OFF
Switch SW1 and SW2 to OFF if monostable mode is required. Leave switch 2 on ON if bistable mode is required.

SW1	SW2	Relay 1	Relay 2	Relay 3	Relay 4
OFF	OFF	Chosen			
ON	OFF		Chosen		
OFF	ON			Chosen	
ON	ON				Chosen

5. All the outputs are supposed to be Monostable, so that as soon as the button is released the receiver halts running the instruction. To make it so, the following switch setup had to be set up.

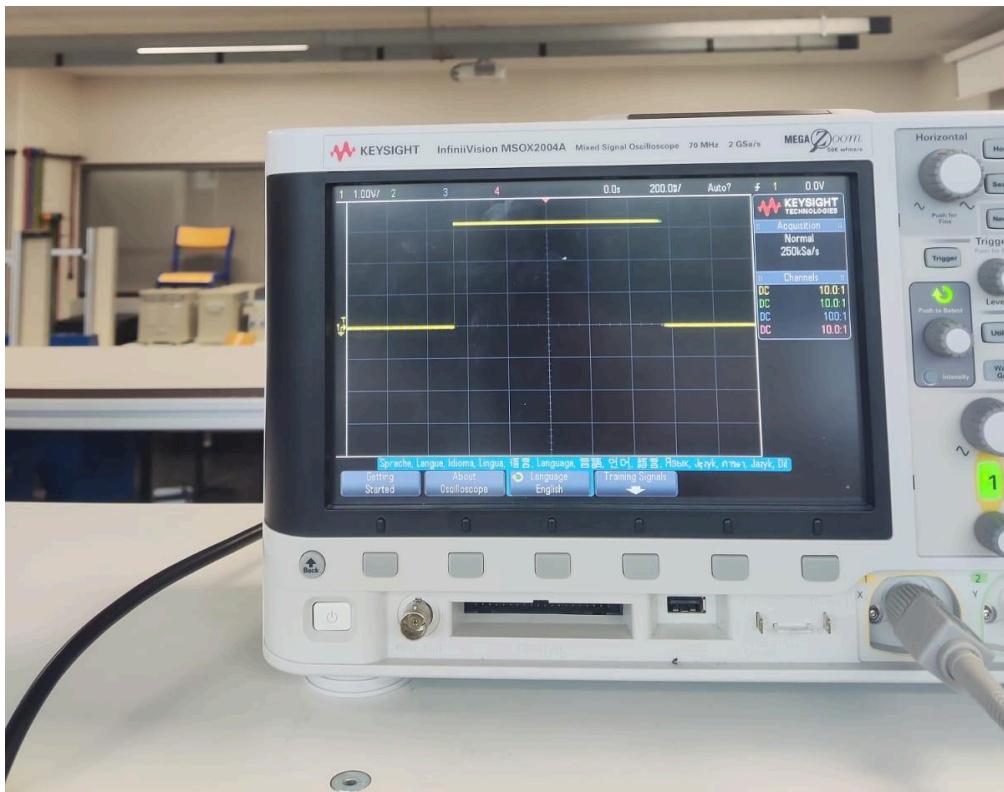
SW1	Relays 1 and 2	SW3-SW4-SW5
OFF	Monostable	OFF-OFF-OFF
ON	Bistable	OFF-OFF-OFF

SW2	Relays 3 and 4	SW3-SW4-SW5
OFF	Monostable	OFF-OFF-OFF
ON	Bistable	OFF-OFF-OFF

6. Then, add new instructions to the receiver individually, individual outputs. The switches were set up according to the instructions.

ADD/REMOVE operation for a chosen relay	
1/ Select «Add/Remove» mode with SW5 ON.	
2/ Select SW3 OFF and SW4 OFF.	
3/ Select the relay that the remote control is to be used to control by setting switches SW1 and SW2 (cf Table below). Note: consider switching the switches to ON again if bistable mode is required.	
4/ Press any key on the remote control.	
5/ Exit «Add/Remove» mode with SW5 OFF.	
Two results are possible with this operation:	
This association had not been set up previously. The association is now created. The LED flashes once.	
This association had been set up previously. The association is erased. The LED flashes twice.	

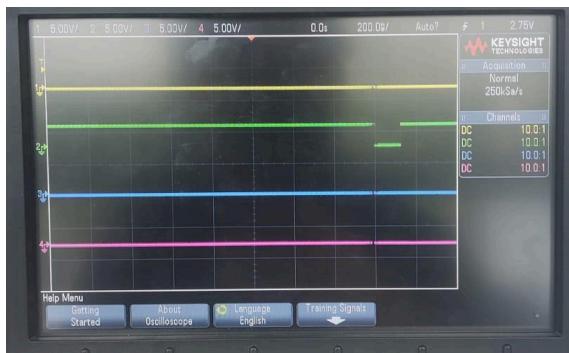
7. Once the switches are in place, pressing the registered key causes a spike in the oscilloscope, causing the motor to operate.
8. For Example, you can see Output 1 after the implementation of the new instructions.



9. Similarly, the Step 6 was repeated for all the Relays (Outputs) and the desired outputs were achieved.
Note: Once the instructions are added to the receiver, only Relays (Output) 1,2,3 and 4 (Pin 3, 13, 11, 9 respectively) are required to be connected to the circuit and oscilloscope.



Output 1 (Yellow)



Output 2 (Green)



Output 3 (Blue)



Output 4 (Pink)

And with that, we have established the connection between the remote and receiver.

Next Step: Making the connection between the motors and the Receiver.

3. Contrôle de l'orientation du panneau photovoltaïque “face au soleil”

L'exigence à respecter pour cette partie est la suivante : “**Orientation du panneau à l'irradiance maximale**”.

Pour répondre à cette condition et pour des questions de praticité, nous allons balayer à des angles de plus ou moins 90° autour de l'exposition Sud. En effet, une orientation du panneau photovoltaïque au Sud permet de capturer le maximum de lumière tout au long de la journée.

Une méthode de mesure de luminosité serait de commencer à -90° et de se décaler petit à petit vers $+90^\circ$ jusqu'à obtenir la luminosité maximale. Celle-ci peut être identifiée lorsque la luminosité va commencer à baisser une fois que l'orientation optimale aura été dépassée. Nous nous attendons alors à obtenir une courbe de luminosité sous forme de gaussienne avec une valeur maximale atteinte lorsque le capteur est face au soleil.

Notre système est lent (dû à la trajectoire du soleil dans le ciel), nous pouvons donc mesurer la luminosité toutes les 15 minutes par exemple pour ajuster l'exposition du panneau photovoltaïque.

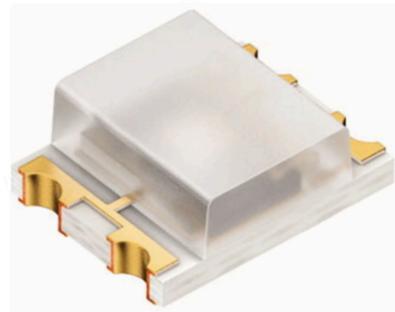
a. Pistes de recherche et premières solutions

Les premières étapes du contrôle de l'orientation du panneau photovoltaïque dans notre projet consistent à se renseigner sur les différentes possibilités qu'il est possible d'utiliser pour mesurer la luminosité.

i. SFH 5711

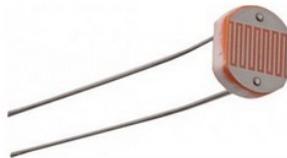
Une première possibilité consiste à placer un capteur au fond d'un cône pour concentrer les mesures vers une seule orientation et gagner en précision lors de la mesure de la luminosité. Notre professeur encadrant nous propose d'utiliser le **capteur de luminosité SFH 5711** pour essayer cette solution.

Ce capteur de luminosité ambiante CMS utilise une photodiode sensible à la lumière visible comme technologie de détection. Il fournit une sortie analogique proportionnelle à l'intensité lumineuse détectée.



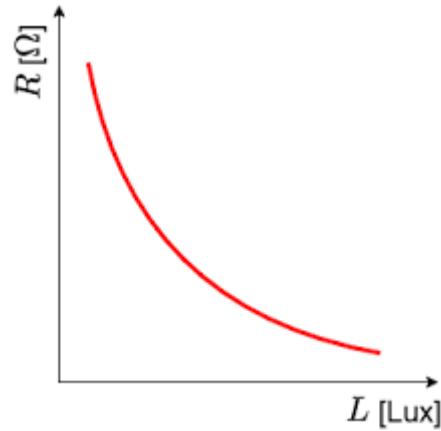
ii. Photorésistance

Une autre solution que nous avons utilisée lors d'un précédent projet consiste à utiliser une **photorésistance** en tant que capteur de luminosité. C'est un composant électronique dont la résistance varie en fonction de la luminosité auquel il est soumis.



Son principe de fonctionnement est le suivant :

- la valeur de la résistance diminue lorsque la luminosité augmente
- inversement, la résistance augmente lorsque la luminosité diminue

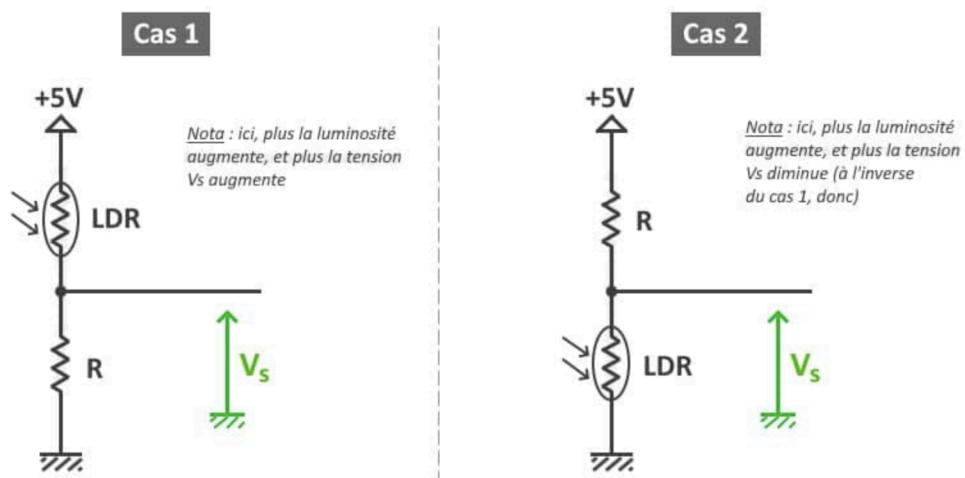


Pour trouver la valeur de la résistance, nous utilisons la méthode du pont diviseur de tension :

La photorésistance est mise en série avec une résistance de valeur fixe (220 Ω dans mon cas), on obtient alors une tension qui varie en fonction de la valeur de la photorésistance.

Comme les valeurs de la résistance et de la tension d'alimentation sont connues, il suffit de mesurer la tension avec un pont diviseur de tension pour déterminer la photorésistance.

2 cas sont alors possibles en fonction du placement des composants dans le montage :



Cas 1 :

La formule du pont diviseur de tension nous donne :

$$V_S = \frac{R}{R + LDR} * V_{CC} \Leftrightarrow LDR = R * \frac{V_{CC} - V_S}{V_S}$$

Ainsi, nous remarquons que dans ce cas, plus la luminosité augmente, plus la tension Vs augmente.

Cas 2 :

La formule du pont diviseur de tension nous donne :

$$V_S = \frac{LDR}{R + LDR} * V_{CC} \Leftrightarrow LDR = R * \frac{V_S}{V_{CC} - V_S}$$

Ainsi, nous remarquons que dans ce cas, plus la luminosité augmente, plus la tension Vs diminue.

Après avoir réalisé notre routage, nous réalisons que nous sommes dans le deuxième cas. Nous lisons puis affichons la valeur de l'entrée analogique, il s'agit d'une valeur sur 10 bits fournie par le convertisseur analogique-numérique de la carte Arduino. La tension de sortie varie donc entre 0 et 5 V (valeurs affichées comprises entre 0 et 1023). Pour contourner cet erreur de conception, nous changeons la façon de lire les données dans notre code arduino :

LCD.print(1023 - readingPR);

iii. Panneau photovoltaïque

Les panneaux photovoltaïques, qui convertissent la lumière du soleil en électricité, peuvent également être utilisés comme capteurs de luminosité.

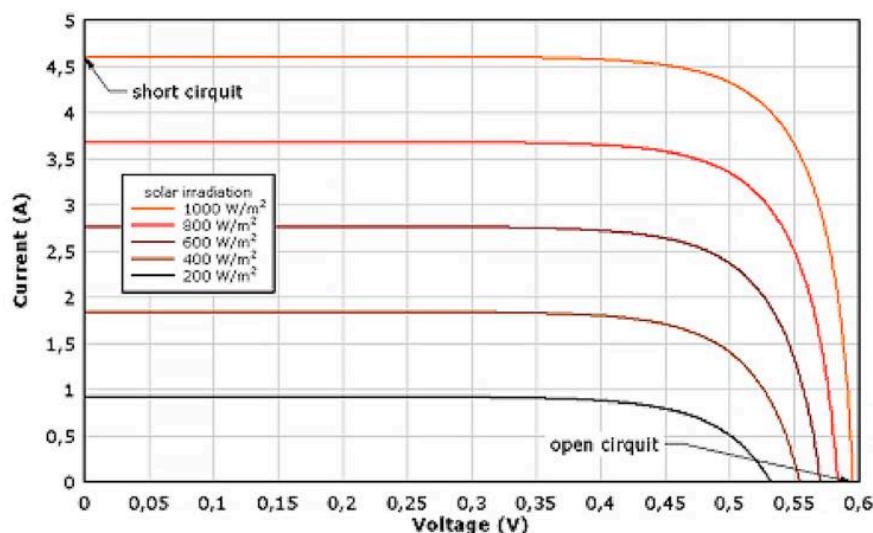
En effet, deux propriétés sont intéressantes à garder en considération pour cet outil :

- Les panneaux photovoltaïques génèrent un courant électrique lorsqu'ils sont exposés à la lumière.
- Ses cellules génèrent une tension électrique lorsque des photons de lumière entrent en interaction avec le matériau semi-conducteur (silicium) de la cellule.

En exploitant ces principes, les panneaux photovoltaïques peuvent être utilisés comme capteurs de luminosité, le courant généré par un panneau photovoltaïque est proportionnel à l'intensité lumineuse incidente. À mesure que la luminosité augmente, le courant de sortie du panneau photovoltaïque augmente également.

La tension générée est également proportionnelle à l'intensité lumineuse incidente. Cependant, elle ne varie pas autant que le courant (cf graphique ci-dessous). Lorsqu'on passe d'une irradiance de 1000 W/m^2 à 500 W/m^2 , le courant de court-circuit est divisé par 2 car il existe une relation de proportionnalité avec l'éclairement : $I_{SC} = k * E$.

La tension quant à elle ne varie que très peu.

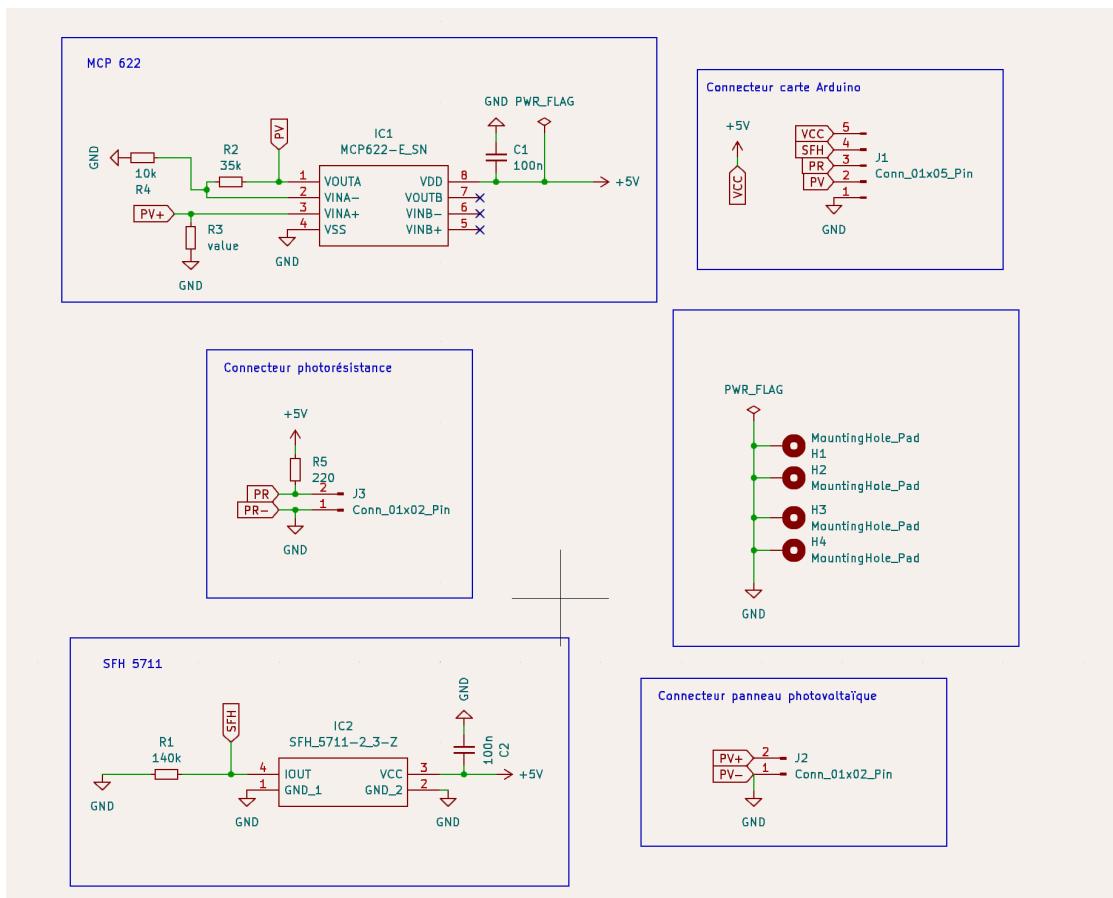


Caractéristique courant-tension du panneau photovoltaïque pour différents éclairements

b. Réalisation du PCB de capteurs de luminosité

Pour essayer les trois solutions détaillées ci-dessus, nous réalisons un premier PCB de tests. Celui-ci nous permet de comparer les différents moyens de mesure de la luminosité et de sélectionner la solution la plus adaptée.

Nous commençons par réaliser le schématique de notre projet :

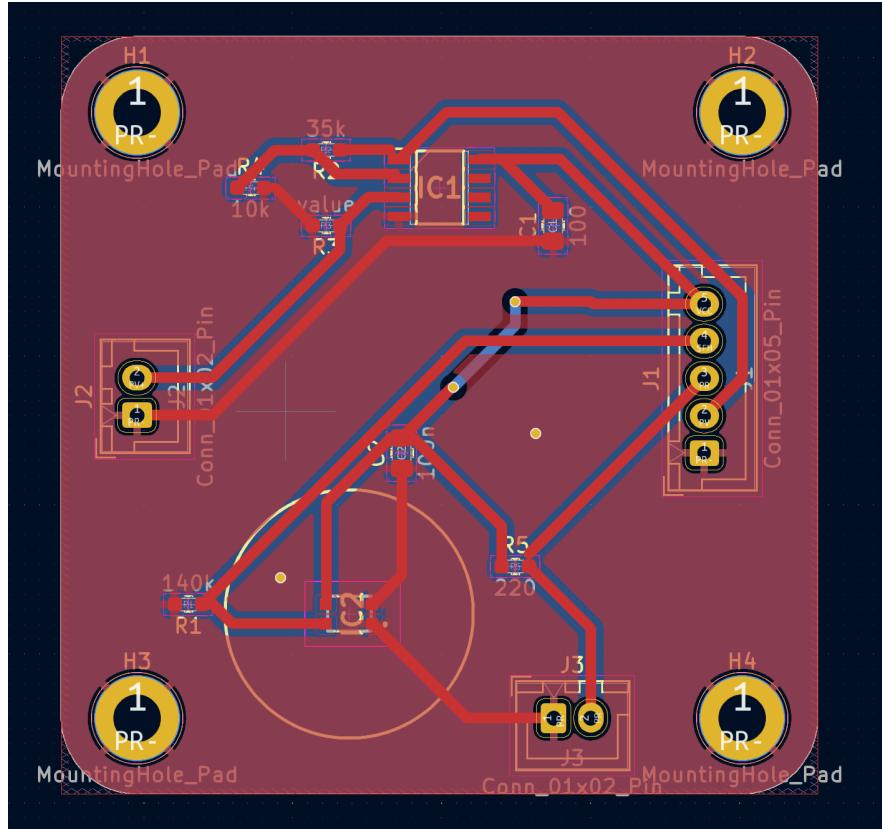


Le composant MCP 622 est un amplificateur opérationnel, il est utilisé pour amplifier le signal fourni par le panneau photovoltaïque (utilisé en tant que capteur).

Nous pouvons déterminer son gain avec les valeurs de nos composants :

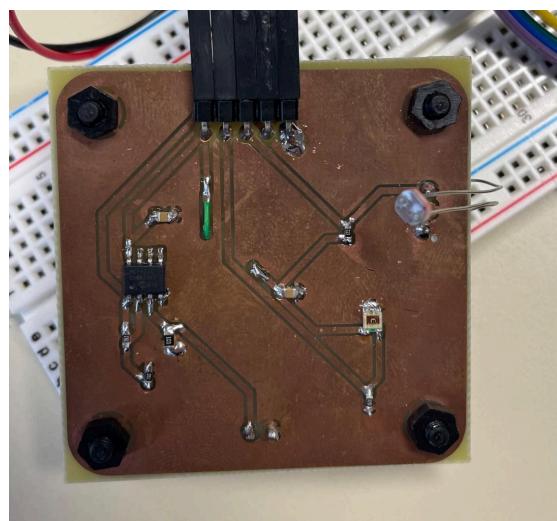
$$V_{IN} = \frac{10\,000}{10\,000 + 35\,000} * V_{OUT} \Leftrightarrow g = \frac{V_{OUT}}{V_{IN}} = 4,5$$

Nous passons ensuite au routage de notre PCB :



c. Essais des capteurs de luminosité

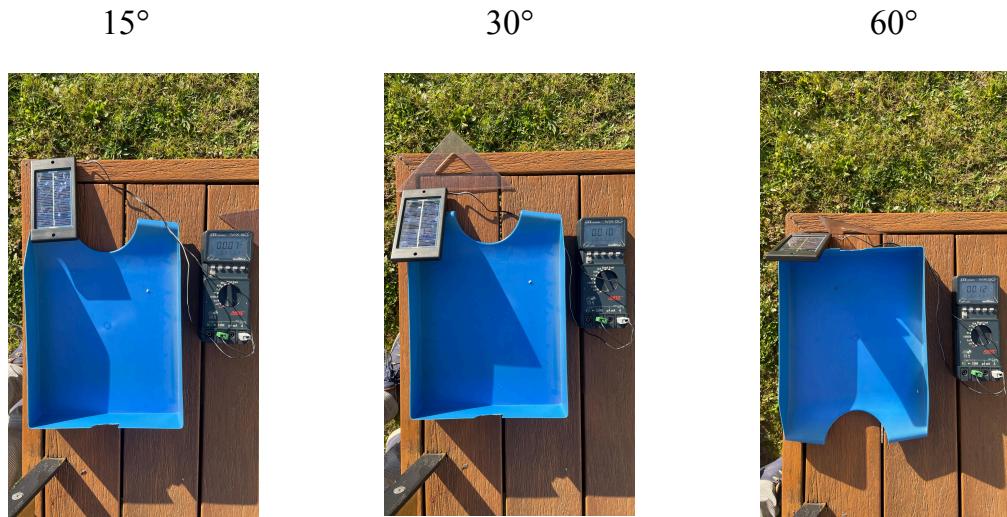
Après avoir terminé la conception de notre PCB, nous avons soudé l'ensemble des capteurs et composants :



Nous pouvons alors réaliser différents essais :

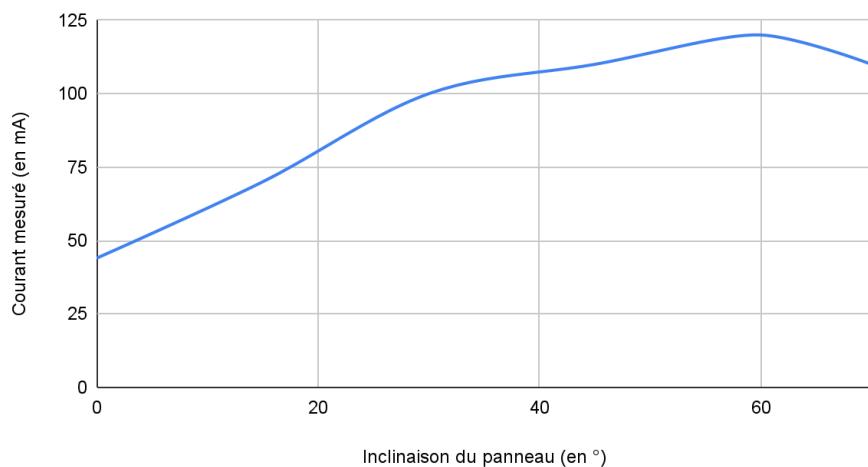
i. Essais en inclinaison

La première série de tests que nous avons réalisée consiste à mesurer le courant dans le panneau photovoltaïque (qui sert ici en tant que capteur de courant) en fonction de son l'inclinaison pour connaître l'angle de configuration optimal.



Après avoir effectué les mesures de courant, nous avons tracé la courbe suivante :

Courant mesuré en fonction de l'inclinaison du panneau



Nous remarquons donc que l'inclinaison du panneau optimale dans ces conditions de test est de 60°. Cependant, ces valeurs dépendent de la date de réalisation des essais.

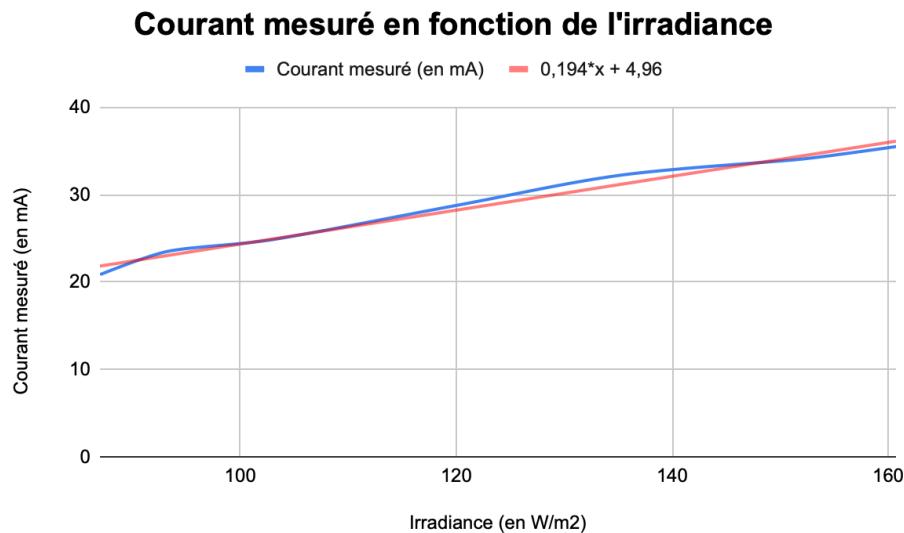
En effet, le soleil n'a pas la même trajectoire toute l'année. **En France, l'inclinaison optimale pour un panneau photovoltaïque se situe entre 30° et 40°.** Cela permet d'être perpendiculaire aux rayons du soleil et ainsi de produire un maximum d'électricité.

ii. Essais en fonction de l'irradiance

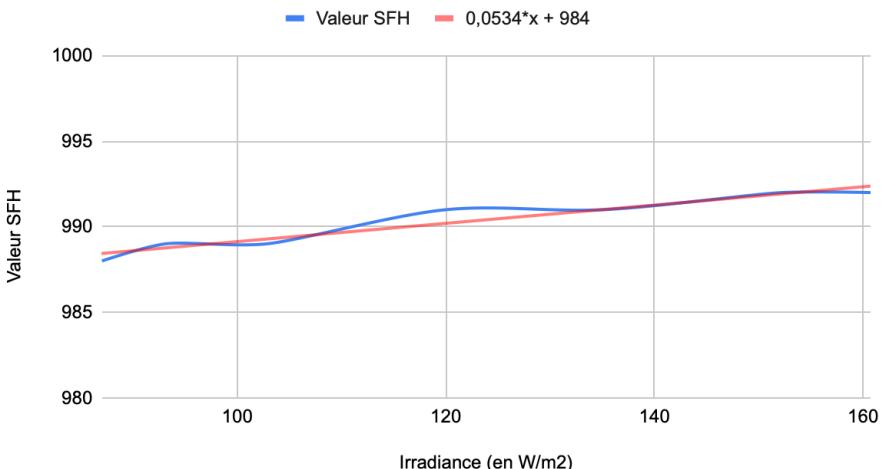
Nous utilisons ici un irradiance-mètre, il s'agit d'un capteur qui mesure l'éclairement en W/m². En branchant nos capteurs à un écran LCD et notre petit panneau photovoltaïque à un ampèremètre, nous pouvons déterminer la relation qui existe entre les valeurs obtenues et l'éclairement.

Dans le cas de notre panneau photovoltaïque, nous pouvons mesurer le courant et ainsi trouver la relation linéaire qui existe entre l'éclairement et le courant.

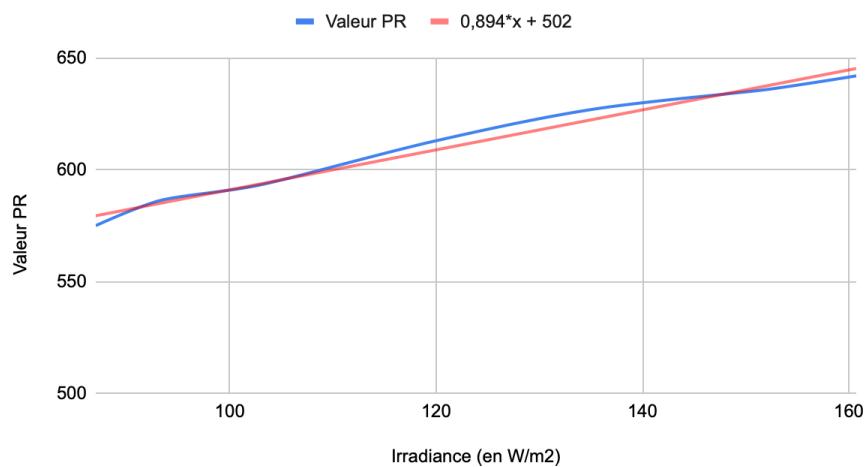
Nous regroupons les différents relevés dans un tableau et traçons les courbes de chaque capteur :



Valeur SFH en fonction de l'irradiance



Valeur PR en fonction de l'irradiance



Le panneau photovoltaïque et la photorésistance permettent d'observer une évolution des valeurs de sortie en fonction de l'irradiance qui varie. Cependant, les valeurs du capteur SFH sont très rapprochées les unes des autres et ne permettent pas d'avoir une mesure fiable de l'éclairement.

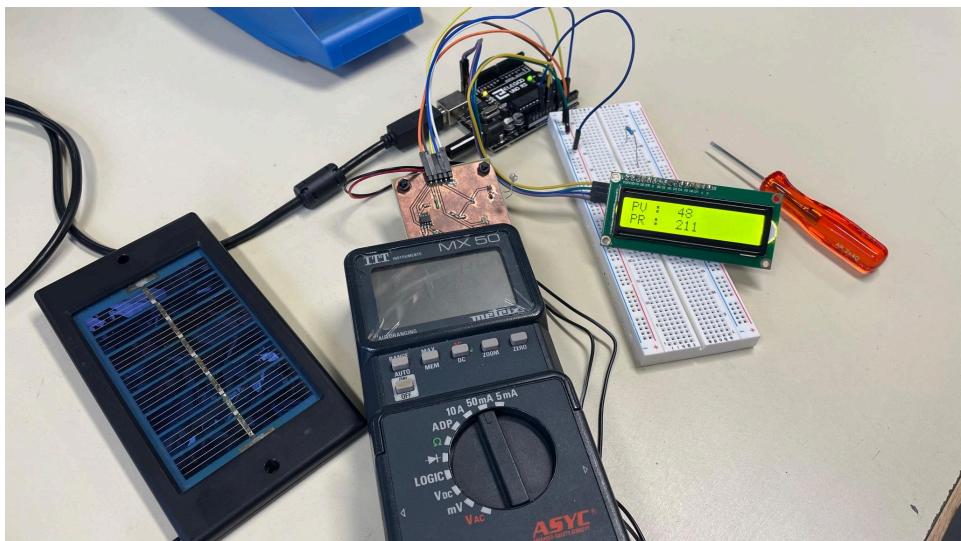
Nous nous concentrerons donc désormais sur l'utilisation du panneau photovoltaïque en tant que capteur de luminosité.

d. Utilisation du panneau photovoltaïque comme capteur de luminosité

Le premier objectif est de lire la valeur délivrée par le panneau photovoltaïque à l'aide de la carte Arduino qui remplace l'ampèremètre utilisé jusqu'ici.

Nous relions donc désormais notre petit panneau solaire à notre PCB via les connecteurs. Le signal est alors amplifié à l'aide du MCP 622 avant d'arriver à la carte Arduino.

Nous pouvons ensuite lire l'image du courant et l'afficher sur l'écran LCD :



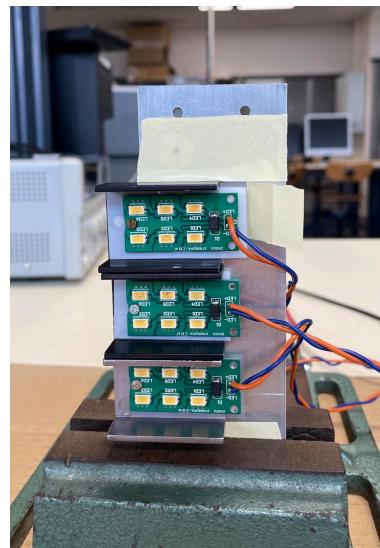
e. Recherche de l'irradiance maximale

Maintenant que notre capteur est correctement branché et fonctionnel, nous pouvons passer à l'étape finale qui consiste à rechercher l'irradiance maximale. Pour cela, commençons par créer une plateforme pour réaliser des essais avec notre panneau.

i. Crédation d'un banc de tests

Étant donné que le soleil n'est pas au rendez-vous à chacune de nos séances de projet, nous créons une source lumineuse puissante pour simuler les rayons du soleil.

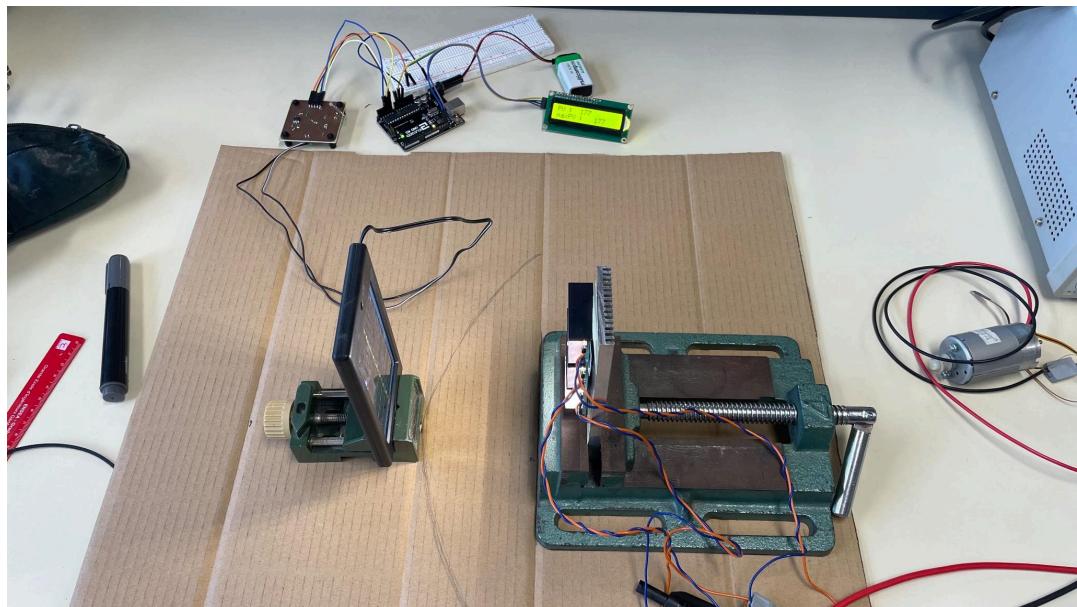
Pour créer ce système, nous recyclons 3 cartes de LEDs déjà existantes que nous relierons en série à l'aide de dominos électriques.



Chaque carte de LEDs doit être alimentée sous 9 V, nous alimentons donc le tout avec 27 V.

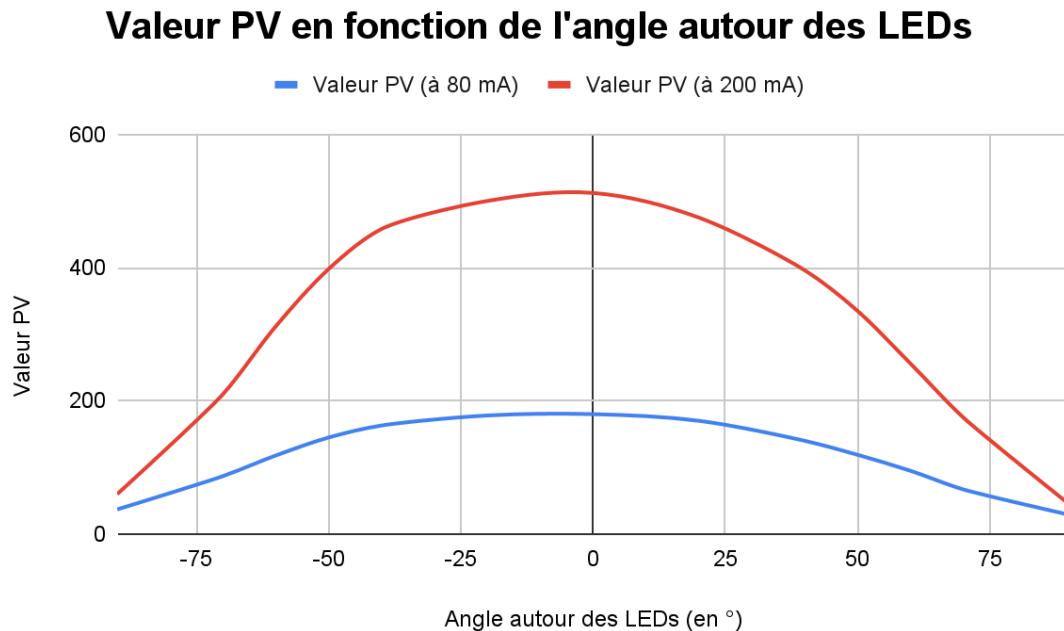
Notre banc d'essai est constitué d'un support fixe pour la barre de LEDs et d'un support à tourner sur lequel est fixé le panneau photovoltaïque.

Nous traçons donc un arc de cercle autour de la source lumineuse pour observer la variation en sortie du capteur en fonction de l'angle par rapport à la lumière :



Nous pouvons alors mesurer les valeurs fournies par le panneau photovoltaïque lorsque nous déplaçons le support autour de la barre de LEDs.

Nous obtenons les résultats suivants pour deux valeurs de courant différentes :



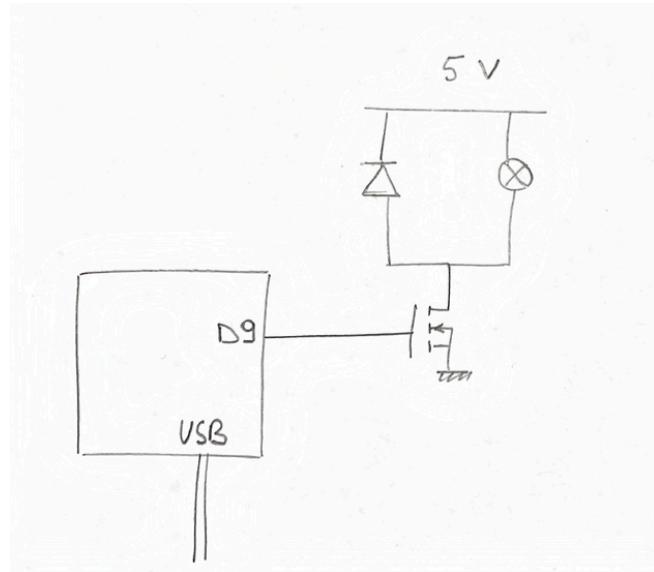
Même si l'intensité n'est pas la même, le comportement est identique à 80 mA et à 200 mA : les courbes ont une forme de gaussienne et la valeur maximale est obtenue lorsque le panneau solaire est situé face à la source lumineuse.

ii. Ajout d'un moteur à courant continu

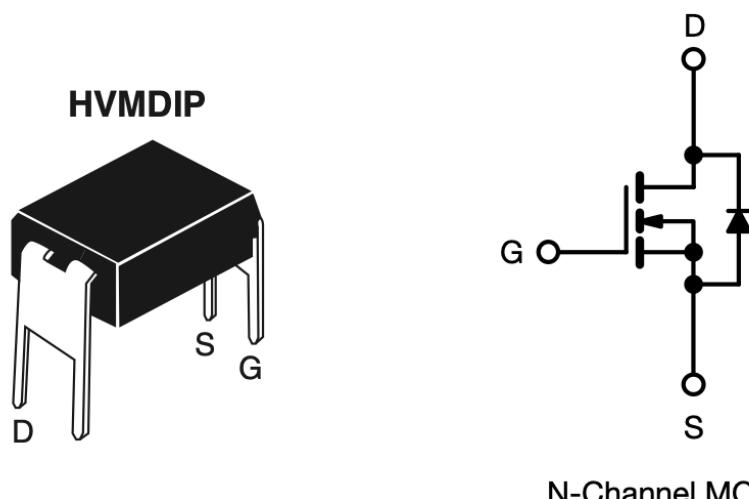
Nous gardons en tête notre objectif final qui est de positionner le groupe électrogène autoporté face au soleil pour garantir une production d'énergie optimale.

Nous ajoutons donc un petit moteur à courant continu à notre dispositif pour simuler le déplacement du chariot.

Cependant la carte Arduino délivre un courant trop faible pour faire tourner le moteur, nous ajoutons un transistor MOSFET IRFD 110 en sortie du pin digital de la carte Arduino et une diode de roue libre (1N 4148) en parallèle avec le moteur.



La diode est une simple protection, elle permet de fournir un chemin de retour pour le courant lorsque le transistor MOSFET est désactivé. Cela aide à éviter les pics de tension induits par l'inductance de la charge lors de la commutation.



N-Channel MOSFET

Extrait datasheet IRFD 110

Après le branchement des différents composants, nous connectons notre moteur au reste du circuit et nous retrouvons un fonctionnement normal et une vitesse de rotation convenable.

Un premier essai que nous réalisons à l'aide du banc de tests est le suivant :

- le moteur tourne lorsque le panneau photovoltaïque s'éloigne de sa position 0° , c'est à dire que le capteur reçoit moins de lumière
- le moteur s'arrête lorsque la valeur du capteur est maximale. Autrement dit, le panneau se situe à la position souhaitée : devant le soleil.

Après avoir mis au point cet essai et l'avoir validé avec notre professeur, nous passons à une deuxième étape plus complexe :

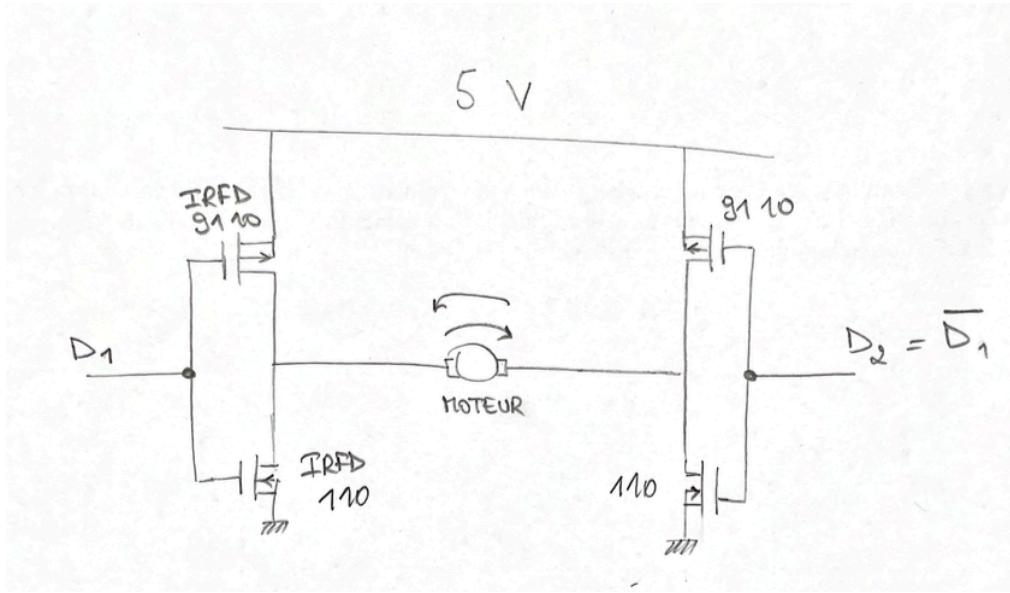
Nous capturons la valeur de la luminosité à un moment donné, nous faisons tourner le moteur si cette valeur est inférieure à la valeur maximale. Puis nous capturons la luminosité peu de temps après :

- si la valeur est inférieure à la précédente, cela signifie que nous nous éloignons de la source de lumière, le sens de rotation doit changer
- si la valeur est supérieure, cela signifie que nous nous approchons du soleil, le moteur continue à tourner dans le même sens.

Après avoir écrit le code, nous réalisons que cette méthode ne peut pas fonctionner avec la configuration actuelle (1 seul transistor MOSFET). Il faudrait un hacheur 4 quadrants composé de 4 MOSFETs pour faire tourner le moteur dans les 2 sens de rotation.

iii. Construction d'un hacheur 4 quadrants et finalisation des essais

Nous réalisons le schéma suivant que nous connectons à la carte Arduino :



Il y a 2 transistors MOSFETs canal N (*IRFD 110*) et 2 transistors canal P (*IRFD 9110*).

D1 est à 0 ou à 1, D2 est son complémentaire. Ce sont les positions HIGH et LOW dans le code Arduino.

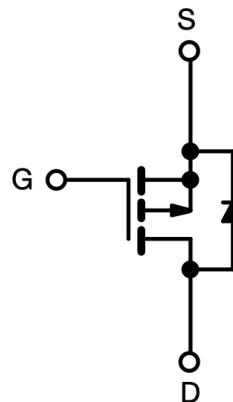
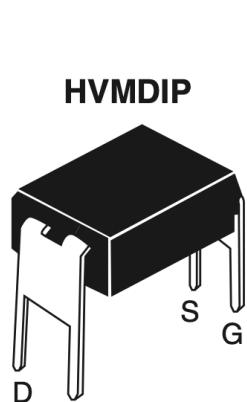
```

71   if(readingPV > lastReadingPV) {
72     digitalWrite(3, HIGH);
73     digitalWrite(motorPin1, HIGH);
74     digitalWrite(motorPin2, LOW);
75   }
76   else {
77     digitalWrite(3, LOW);
78     digitalWrite(motorPin1, LOW);
79     digitalWrite(motorPin2, HIGH);
80   }
81

```

Le IRFD 110 est un MOSFET de canal N, et il est utilisé lorsque la tension de commande est positive par rapport à la source.

Alors que le IRFD 9110 est un MOSFET de canal P, ce qui signifie qu'il est utilisé lorsque la tension de commande est négative par rapport à la source.



Extrait datasheet IRFD 9110

Les MOSFETs sont utilisés comme des amplificateurs de tension, permettant d'amplifier un signal électrique appliqué à la grille pour contrôler un courant plus important entre le drain et la source.

Comme nous voulons faire un petit driver de moteur, nos MOSFETs sont aussi utilisés comme interrupteurs électroniques, ils commutent. En particulier, ils permettent de laisser passer du courant, ou pas, entre le drain et la source en fonction du signal appliqué à la grille.

Dans un hacheur à 4 quadrants, qui permet le contrôle bidirectionnel du courant, il y a généralement des transistors (MOSFETs) qui commutent la polarité du courant dans la charge. Lorsque le transistor est éteint, la diode de roue libre fournit le chemin pour le courant induit par l'inductance de la charge.

Nous arrivons maintenant à faire tourner le moteur dans un sens et dans l'autre. Nous pouvons alors réaliser notre dernier essai avec cette maquette pour nous approcher le plus possible du comportement souhaité sur le système final :

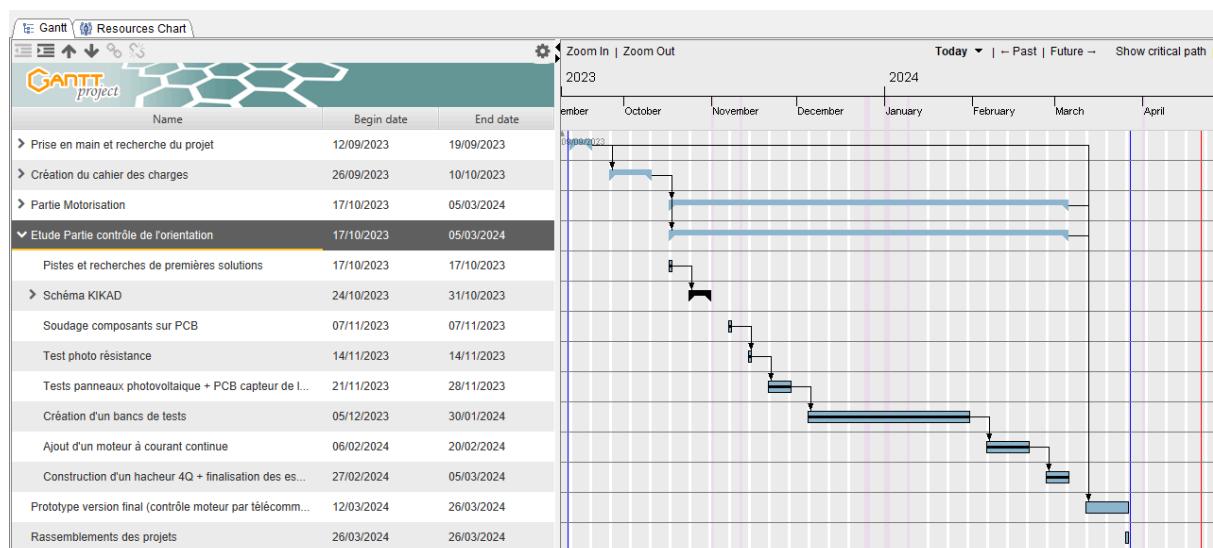
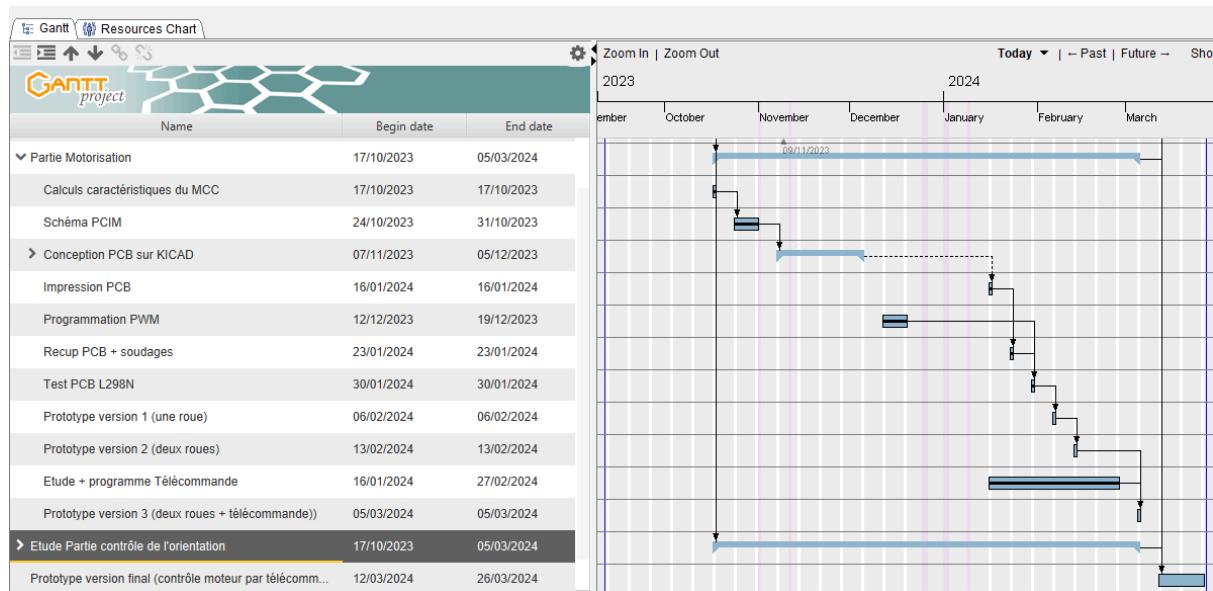
Lorsqu'on balaye autour de la source de lumière, si la luminosité augmente, nous nous approchons du soleil donc le moteur continue de tourner dans le même sens. Sinon, le moteur change de sens de rotation.

C'est exactement le système de fonctionnement que nous souhaitons et qui nous permet de nous positionner et osciller légèrement lorsqu'on se situe devant la source de lumière (au maximum) car le moteur va dans un autre puis immédiatement dans l'autre etc...

4. État d'avancement du projet

a. Le travail effectué

L'ensemble des tâches effectuées a été rassemblé dans le diagramme Gantt suivant :



Nous avons également attentivement suivi l'évolution du montage mécanique de la structure de l'ensemble du Groupe Électrogène Autoporté avec Monsieur Éric LE BIDAN.

b. Les prochaines étapes

Maintenant que les différentes parties sont terminées indépendamment les unes des autres, nous aurions grandement souhaité rassembler nos travaux pour n'en faire qu'un seul projet complet à la fin.

Il nous reste à unir le contrôle manuel des moteurs et la recherche de l'irradiance maximale pour contrôler automatiquement ou avec la télécommande le mouvement de rotation du groupe électrogène et l'orienter “face au soleil”.

Merci pour votre attention.

***** Fin du Rapport *****