

In.Ra.Ci
Avenue Jupiter, 188
1190 Bruxelles

Technique de qualification
Électronique
Année scolaire 2023-2024

Auto-Mate

Sac Dawid
Rue Vanderkindere 150,
1180 Uccle

Sommaire

1.L'introduction	4
2.Le schéma bloc	5
3.Les caractéristiques	7
4.Les principes mis en jeu.....	8
4.1. L'écran graphique Nextion.....	8
4.2. Le moteur pas à pas	17
4.3. Le driver moteur	21
5.L'étude détaillée des sous-ensembles.....	26
5.1 Schéma de principe.....	26
5.2 Interface utilisateur	27
5.2.1. L'écran graphique Nextion.....	27
5.2.3. Les fins de course	28
5.2.4. Le fusible	28
5.2.5. Programme de test pour les fins de course	29
5.3. Le multiplexeur	30
5.4. Les moteurs pas-à-pas	34
5.4.1. Rôle du condensateur C1	34
5.4.3. Programme de test.....	34
5.5. L'électro-aimant.....	35
5.5.5. Le TIP 120	36
6. La programmation	37
6.1. L'ordinogramme général	37
6.2. Le programme.....	41
7.La fabrication	42
7.1. Liste des pièces requises.....	42
7.2. Modèles en 3D	43
7.3. Assamblage	44
8.La mise au point.....	46
8.1. Les problèmes rencontrés et les solutions	46
8.2. Les améliorations à réaliser	48
9. La conclusion	49

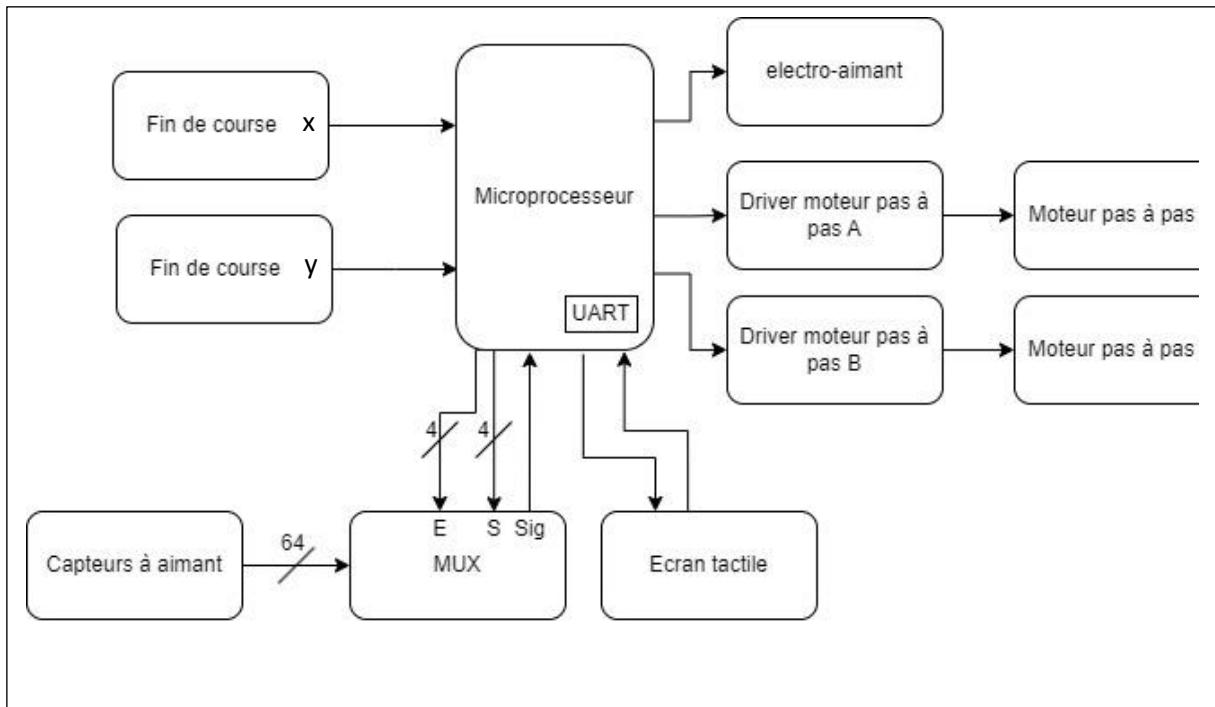
10. Bibliographie et webographie	50
11. Les annexes.....	51
I. Le schéma de principe sans connecteur	51
II. Le schéma de principe avec connecteur.....	52
III. PCB	53
IV. PCB sérigraphie.....	54
VII. Vue 3D.....	55
VIII. Mise en plan	57
IX. Fiches techniques des composants peu courants	66
X. Sources du projet.....	74

1.L'introduction

J'ai le plaisir de vous présenter mon projet passionnant : Auto-Mate qui est un jeu d'échecs automatisé dont le but est de donner une nouvelle dimension à ce jeu. Grâce à un déplacement de pions à distance, je crée la possibilité d'amener le jeu d'échecs en ligne ou contre un robot sur un échiquier. Pour cela, j'ai réalisé un boitier s'occupant du déplacement des pions. Auto-mate est équipé d'un écran, permettant de piloter le déplacement automatique des pions. Il est aussi possible de les déplacer manuellement, au moyen des capteurs se trouvant en dessous de la planche.

J'ai décidé de choisir ce projet, car j'apprécie jouer aux échecs. Ce dernier m'a permis de retrouver les sensations d'un vrai jeu d'échecs sur échiquier tout en ayant la possibilité de jouer contre un adversaire en ligne ou un robot.

2.Le schéma bloc



Le microprocesseur

C'est le cerveau du montage, il permet de piloter chaque composant (les moteurs, les drivers-moteurs, l'électro-aimant, les multiplexeurs et l'écran tactile) et lire les données nécessaires (les fins de courses, les boutons, les capteurs à aimant et l'écran tactile).

L'écran tactile

Il offre la possibilité de naviguer dans les menus pour changer la difficulté de l'IA, affichage du minuteur, modifier les paramètres de la partie ou sélectionner le mode de jeu. Il communique avec l'UART, un bus série asynchrone.

Les drivers-moteurs A et B

Ils pilotent les bobines des moteurs pas à pas et fournissent le courant nécessaire pour faire fonctionner ces derniers.

Les moteurs A et B

Ils sont associés à un système de courroie, ils déplacent l'électro-aimant sur les 2 axes, x et y.

L'électro-aimant

Il sera placé en dessous du damier et sert déplacer les pièces aimantées sur le damier.

Le multiplexeur

Il lit successivement le signal des 8 capteurs et envoi la donnée au microprocesseur.

Les capteurs à aimant

Ils déterminent la présence des pièces d'échecs.

Les fins de course x et y

Ils localisent l'emplacement de l'électro-aimant en faisant une remise à zéro lors du démarrage de la partie. Ils permettent de calibrer la position de ce dernier, permettent de contourner les potentielles dérégulations lors de son déplacement.

3.Les caractéristiques

Générales :

- Écran graphique de couleur tactile, résistif, 320×240 pixels
- 2 boutons d'arcade
- 32 pions aimantés
- Mode de fonctionnement : 1/2 joueurs
- Déplacement automatique des pièces
- Détection automatique des pièces
- Zone pour pions éliminés

Électroniques :

- Alimentation 12V, 2A

Mécaniques :

- Dimensions : 496 mm x 496 mm x 100 mm
- 6 faces en contre-plaqué
- Poids : 4 kg

4.Les principes mis en jeu

4.1. L'écran graphique Nextion



[Lien de l'image](#)



[Lien de l'image](#)

4.1.1. Description

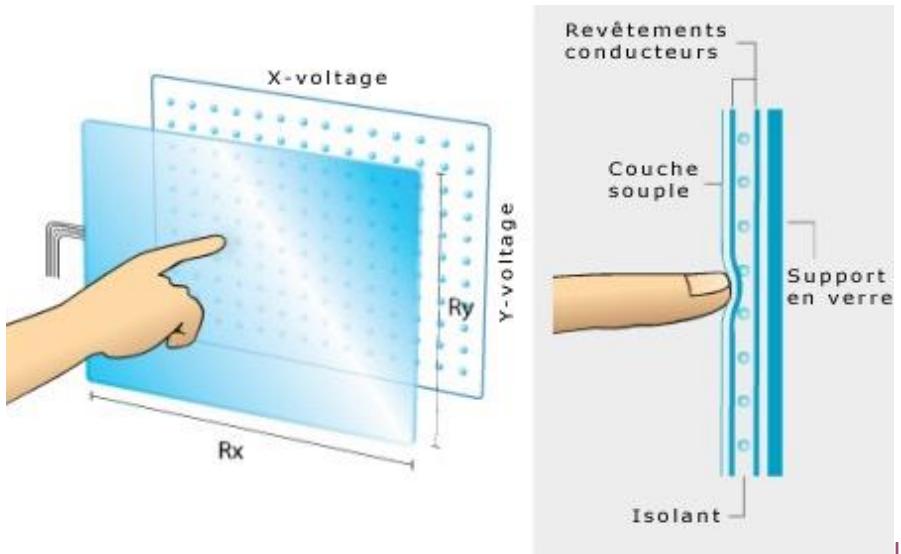
L'écran graphique Nextion est un écran graphique résistif couleur. Il affiche le temps pendant la partie et on peut l'utiliser pour naviguer dans le menu avant de jouer. Il communique avec le microprocesseur par un bus série asynchrone UART.

4.1.2. Caractéristiques

- Modèle : NX3224T024
- Résolution : 320×240 pixels
- Type d'écran : résistif TFT LCD (Liquid Crystal Display)
- Communication : UART (9600 bps utilisés et 115 200 bps max)
- Poids : 20 g
- Tension d'entrée de l'écran : 5V
- Consommation de courant maximale : 90 mA
- Température de fonctionnement minimum : -20 °C
- Température de fonctionnement maximum : 70 °C

4.1.4. Principes de fonctionnement

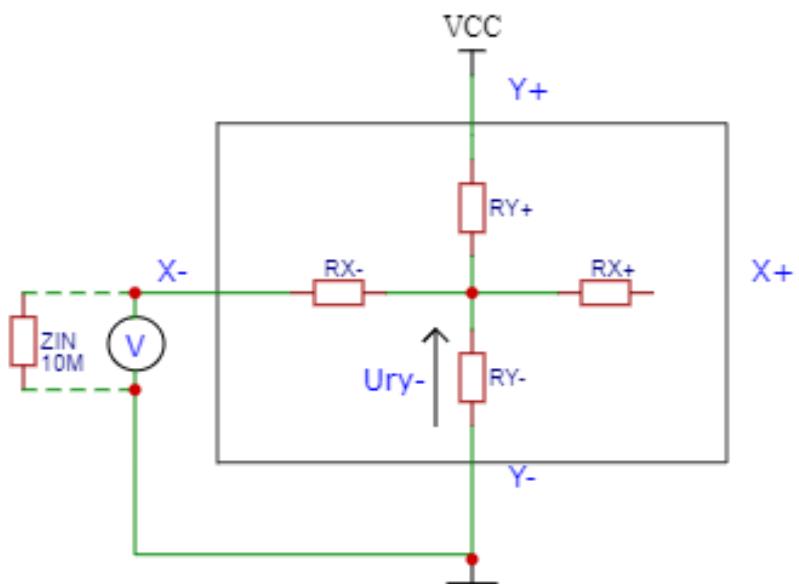
Le fonctionnement du touché de l'écran



[Lien de l'image](#)

L'écran Nextion fonctionne grâce à la pression exercée composée de 3 couches transparentes de ce dernier, la première étant une fine couche de plastique séparée de la couche en verre par un isolant. Lorsque la couche en verre et en plastique entre en contact, cela crée une variation de champs électriques des deux couches, ce qui forme des « résistance » de chaque côté du doigt sur les 2 axes.

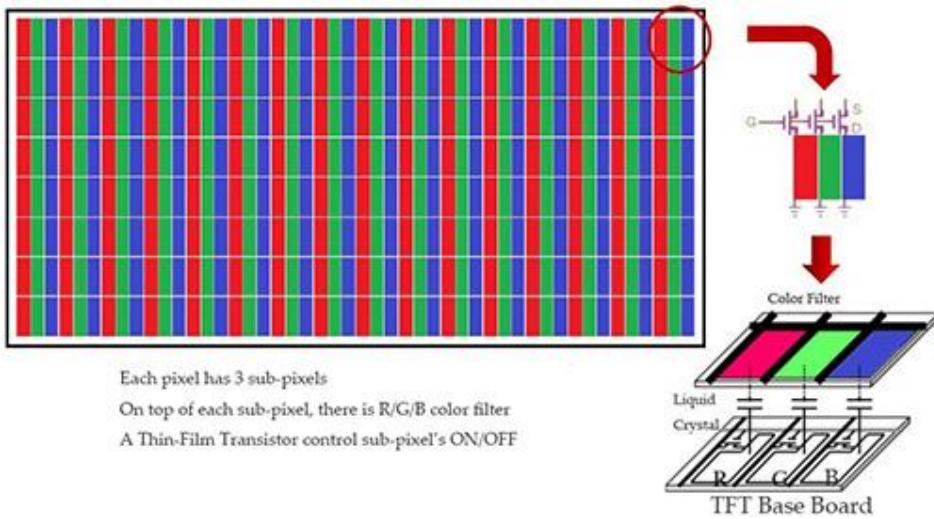
Source : *Definition de tft.* [29.05.2024]. <https://www.lcd-compare.com/definition-de-tft.htm>



Les résistances dans ce genre de circuit font souvent aux alentours de 1 k ohms cependant, cette information ne figure pas dans la datasheet de l'écran Nextion. Pour mesurer la position du doigt, on va mesurer la tension aux bornes de Ry-, qui varie en fonction de la position de l'appui en alimentant Rx- et Ry-. La tension Ur_{x-} ne perturbera pas la mesure, car Rx- est beaucoup plus faible que Zin du voltmètre, rendant la tension à ces bornes négligeable. Pour

avoir la position du doigt sur l'axe X, il suffit de refaire la même opération, en plaçant la masse sur X+, Vcc à X- et mesurer la tension entre Y- et X+. Pour convertir la mesure de tension en position de l'appui sur l'écran par rapport à ce dernier, il faut comparer la tension mesurée à VCC. Par exemple si Vcc = 5V et la mesure de l'axe X est à 2,5 et l'axe y est à 1V alors, l'appui serait sur le milieu de l'axe X de l'écran et un 5e de l'axe Y de l'écran.

Le fonctionnement de l'affichage de l'écran



[Lien de l'image](#)

L'écran Nextion est un écran à matrice TFT LCD (thin-film-transistor liquid-crystal display). Un écran TFT LCD est composé de plusieurs couches. Au cœur de l'écran se trouve une matrice de transistors à couche mince (TFT), qui contrôle l'activation des pixels individuels. Chaque pixel est formé de trois sous-pixels rouges, verts et bleus qui s'activent à différentes intensités pour créer une large gamme de couleurs.

Source : *resistif ou capacitif quel-est le bon ecran tactile.* [29.05.2024].

<https://riverdi.com/fr/blog/resistif-ou-capacitif-quel-est-le-bon-ecran-tactile>

Le fonctionnement de l'UART



[Lien de l'image](#)

L'UART (Universal Asynchronous Receiver/Transmitter) est un protocole de communication permettant la communication entre deux composants en utilisant trois fils uniquement (Rx, Tx, GND). Ce système est full-duplex, ce qui veut dire que les deux composants peuvent envoyer et recevoir des données simultanément. L'UART est aussi asynchrone, ce qui veut dire que le transmetteur et le récepteur n'ont pas besoin d'être synchronisés avec une horloge. Cependant, les deux composants doivent utiliser la même vitesse de communication (baud rates). Pour communiquer, les deux composants devront utiliser la même structure : le bit de start (passage de l'état haut à l'état bas), les bits de données (5 à 9 bits en ASCII), éventuellement un bit de parité (valeur paire si le nombre de bits de données est paire et inversement) et un bit de stop (un 1 numérique). En ce qui concerne notre écran en particulier, on utilise 8 bits de données et un bit de parité.

Librairie utilisée

Software Serial

Remarque : Comme on n'utilise pas les pins Rx/Tx de l'Arduino nano, mais les pins A5, A6, on doit utiliser la librairie [SoftwareSerial](#) qui permet d'utiliser des pins de l'Arduino comme Rx/Tx. Avec cette dernière, la librairie [Nextion](#) ne fonctionne pas. On doit donc récupérer et envoyer les données à l'écran Nextion « manuellement ».

4.1.3. Brochage



[Lien de l'image](#)

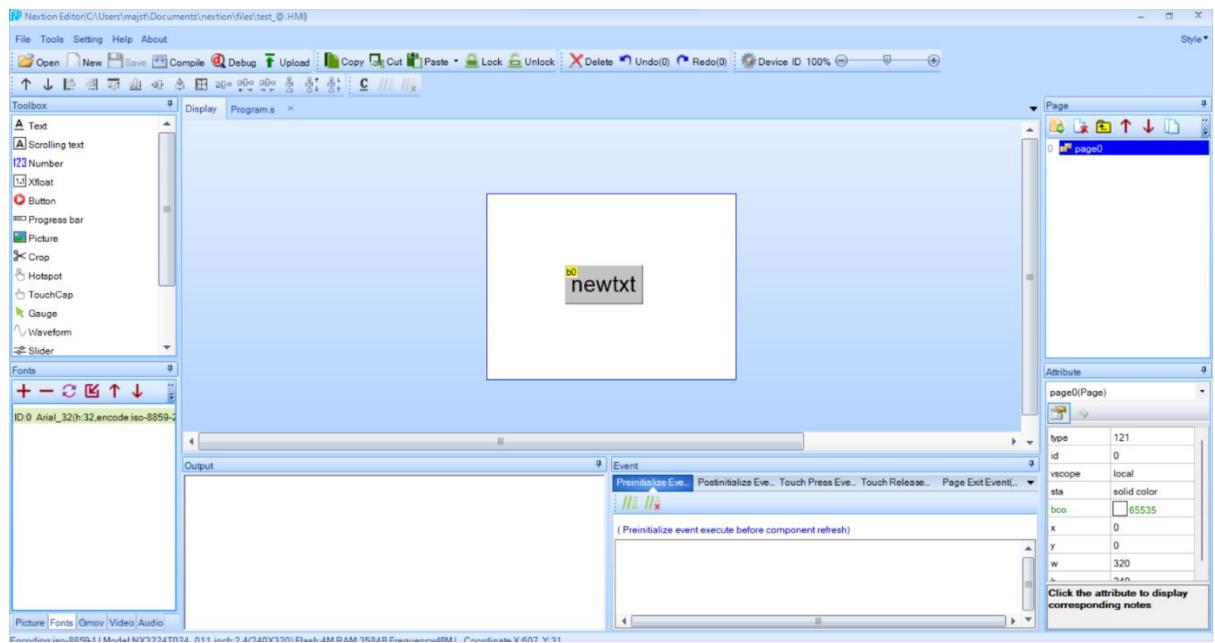
GND connectée à la masse du microprocesseur.

Rx et Tx ils communiquent avec le microprocesseur à 9600 Bauds rate. Le Tx de l'écran est branché au Rx du microprocesseur et inversement pour le Rx de l'écran. Ils communiquent en UART (voir le point « 4.1.4. Principes de fonctionnement » pour l'explication de l'UART).

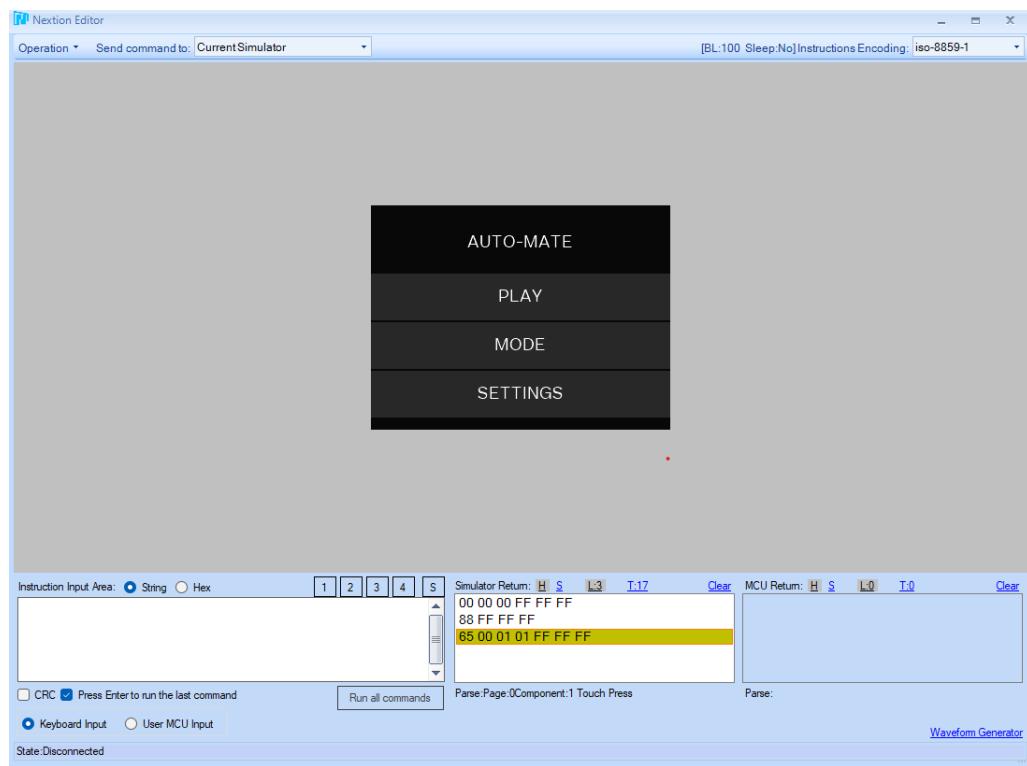
Remarque : l'écran n'est pas réellement connecté aux pattes Rx et Tx du microprocesseur, mais à ces pattes A4 et A5.

VCC connectée au microprocesseur (+ 5V).

4.1.5. Fonctionnement du programme Nextion



En créant des boutons dans le programme, il va générer une id à ce dernier qui peut être envoyé vers le microprocesseur à l'appui ou au relâchement du bouton. Il est possible de récupérer les données de l'Arduino vers l'écran et les afficher, créer des variables et les modifier dans le programme.



Lors de l'appui d'un bouton sur l'écran, il va envoyer la trame de données suivante :

0x 65	La page sur laquelle le bouton se situe	l'ID du bouton appuyé	00 si l'information est envoyée lors du relâchement du bouton ou 01 si l'information est envoyée lors de l'appui du bouton	FF	FF	FF
-------	---	-----------------------	--	----	----	----

Exemple :

0x 65	0x 00	0x 01	0x 01	0x FF	0x FF	0x FF
-------	-------	-------	-------	-------	-------	-------

La trame de données est envoyée à la suite. À cause de cela, il n'est pas possible de récupérer l'information et l'enregistrer sur une variable tel quel, car seule la dernière information envoyée serait enregistrée dans la variable. Pour résoudre ce problème, il suffit de créer un tableau et enregistrer chacune des données, une à la fois, grâce à une boucle.

for :

```
int data[7];  
  
if (mySerial.available() > 0) {  
    for (int i = 0; i < 7; i++) {  
        data[i] = mySerial.read();  
        Serial.println(data[i]);  
    }  
}
```

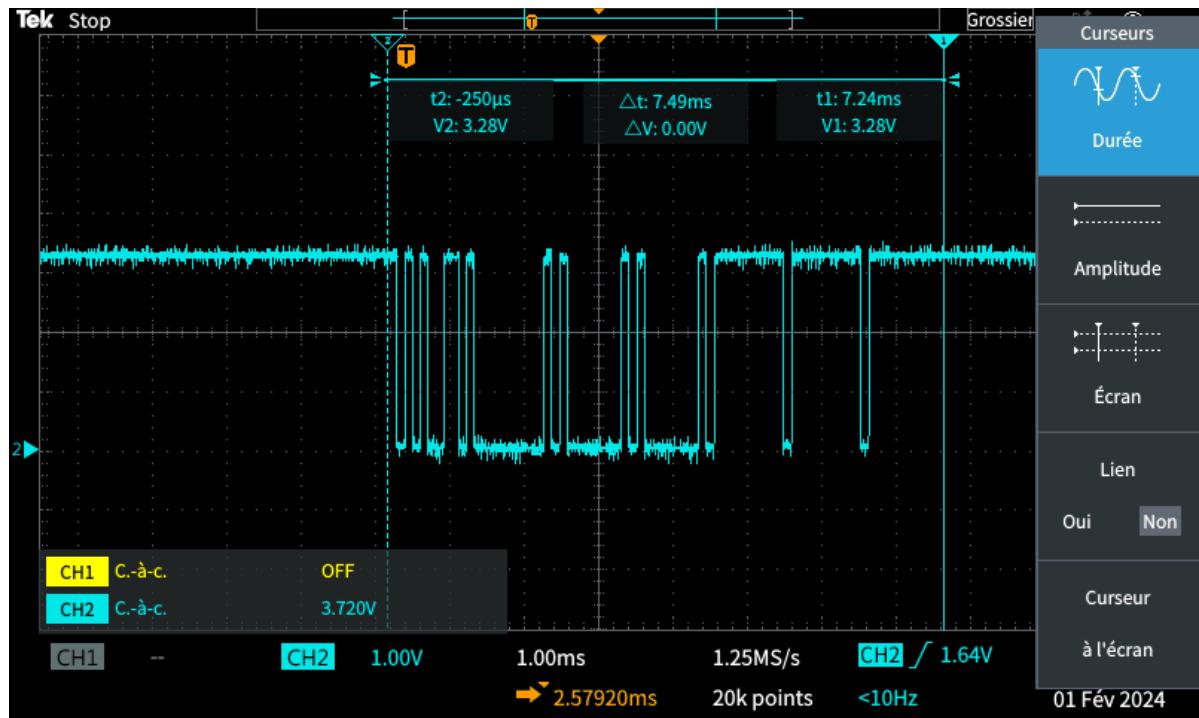
4.1.6. Programme de test

```
#include <SoftwareSerial.h> // Include the SoftwareSerial  
Library  
#define rxPin 18 // A4  
#define txPin 19 // A5  
  
SoftwareSerial mySerial = SoftwareSerial(rxPin, txPin);  
  
void setup() {  
    Serial.begin(9600); // Initialise the Serial port at  
2400 bps.  
    mySerial.begin(9600); // Initialise the mySerial object  
at 9600 bps.  
    while (!Serial) {} // Attente de l'ouverture du port  
série  
    pinMode(rxPin, INPUT);  
    pinMode(txPin, OUTPUT);  
}  
  
void loop() {  
    char data;  
    if (mySerial.available()) {  
        Serial.println(mySerial.read());  
    }  
}
```

Le programme permet de détecter si une donnée est envoyée sur la patte A5 du microprocesseur et de l'afficher sur le moniteur Série. Pour lire le message sur le port série (en utilise la librairie SoftwareSerial()), il faut utiliser la fonction «`mySerial.read()` ». Pour envoyer une donnée à l'écran, il faut utiliser la fonction «`mySerial.print()` », que l'on n'utilise pas encore.

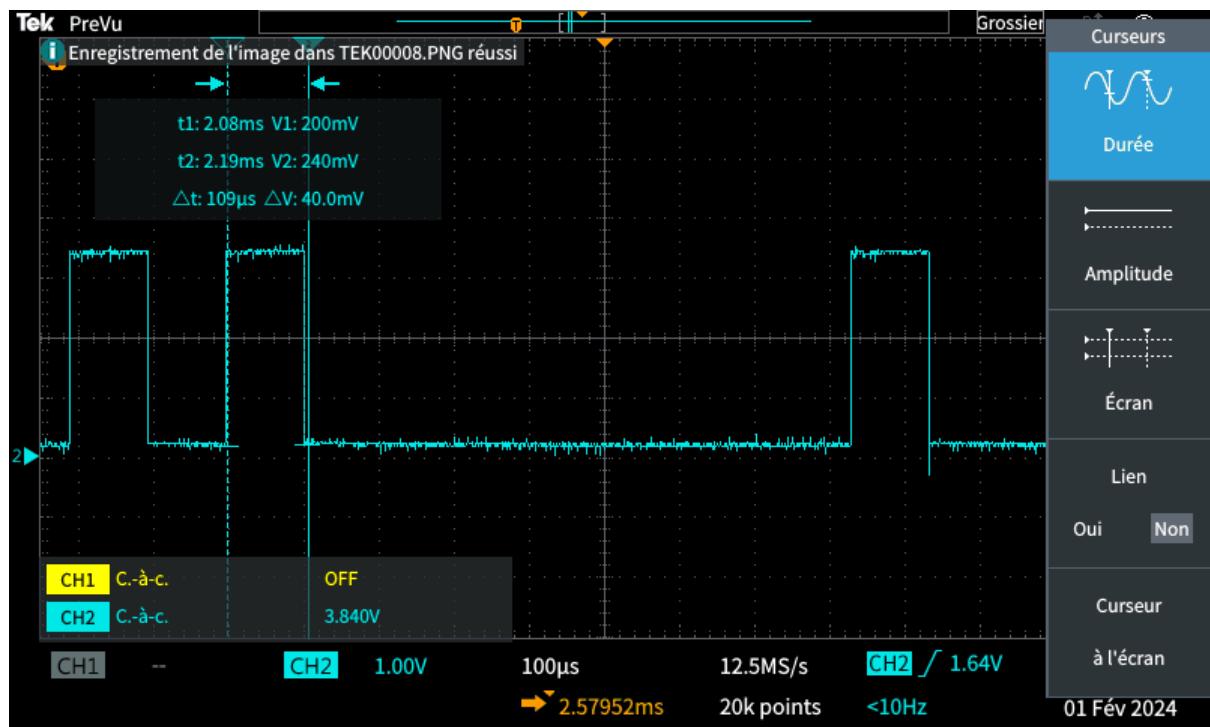
4.1.7. Mesures

Longueur d'une trame de données (65 00 01 00 FF FF FF) :



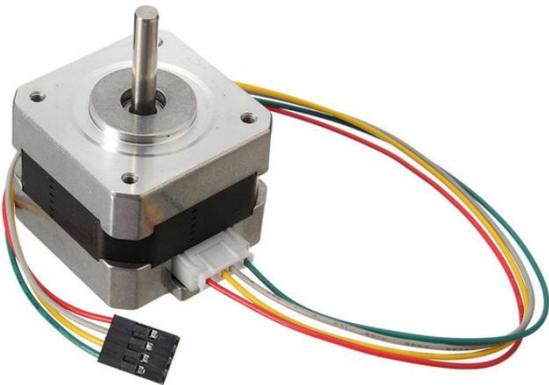
Durée d'un bit : 109μs

Calcul de la durée d'un bit : 1/9600



4.2. Le moteur pas à pas

Nema 17 17HS4401



[Lien de l'image](#)

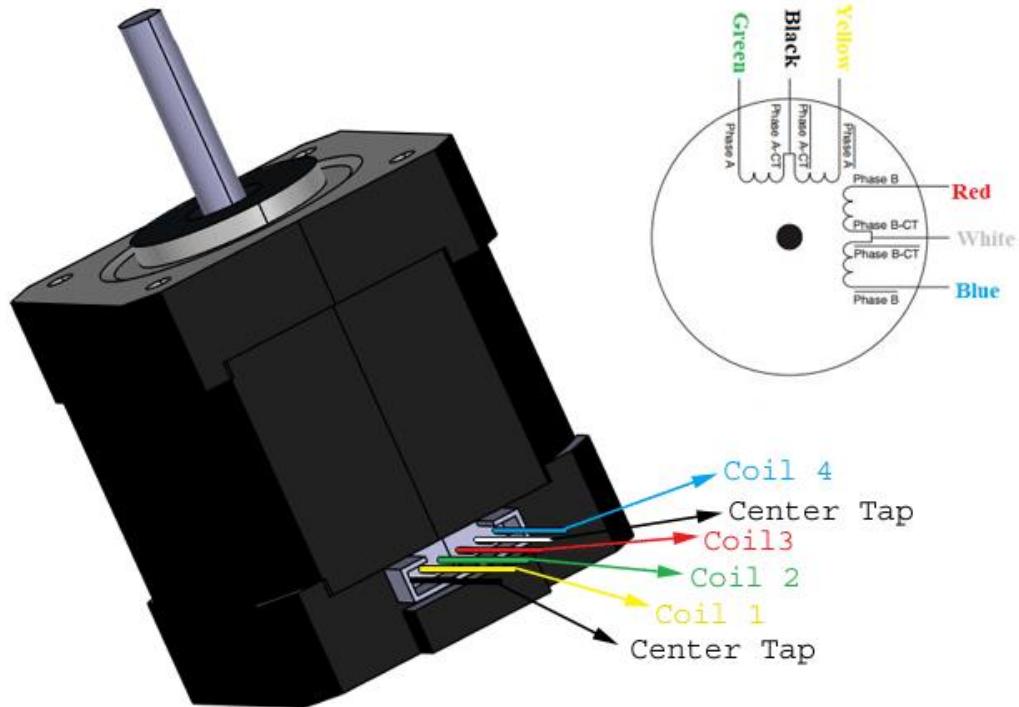
4.2.1. Description

Le moteur pas-à-pas va faire des rotations précises en faisant des pas un par un. Il peut aussi maintenir son emplacement à un angle précis. On peut le commander en envoyant du courant continu dans un certain ordre aux entrées de ce dernier. Pour cela, il faut utiliser un driver moteur, le Pololu A4988 pour mon projet.

4.2.2. Caractéristiques

- Tension d'entrée du moteur : 8V – 35V
- Courant continu par phase : 1.7A
- Couple de maintien : 40 N. cm
- Couple de détente : 2,2 N. cm
- 1,8° par pas (STEP)
- Précision d'angle par pas : +/- 5 %
- 200 pas/révolution (tour complet de 360°), on peut le calculer en divisant 360° par l'angle tourné par pas -> $360^\circ / 1,8^\circ = 200$

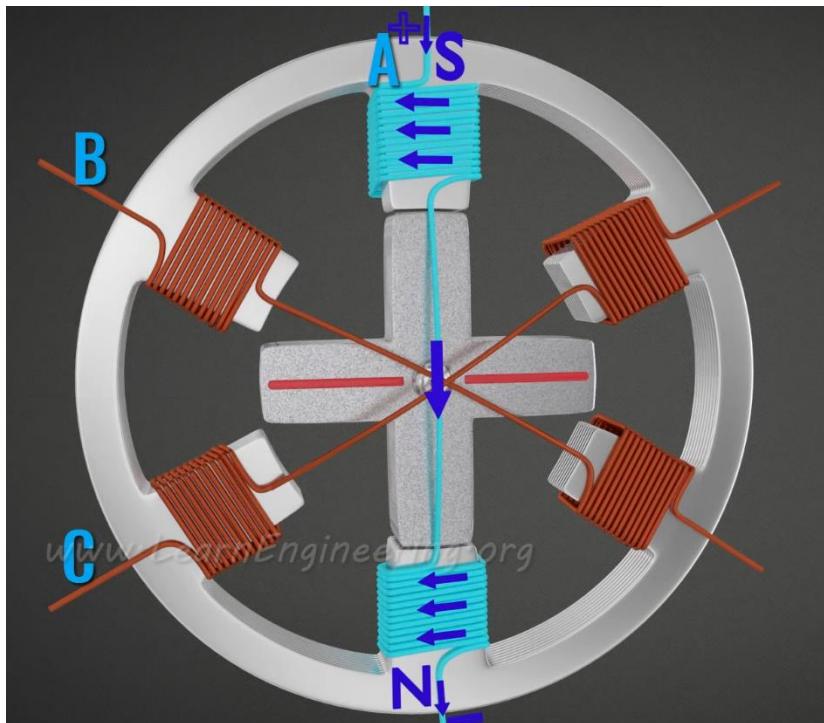
4.2.3. Brochage



[Lien de l'image](#)

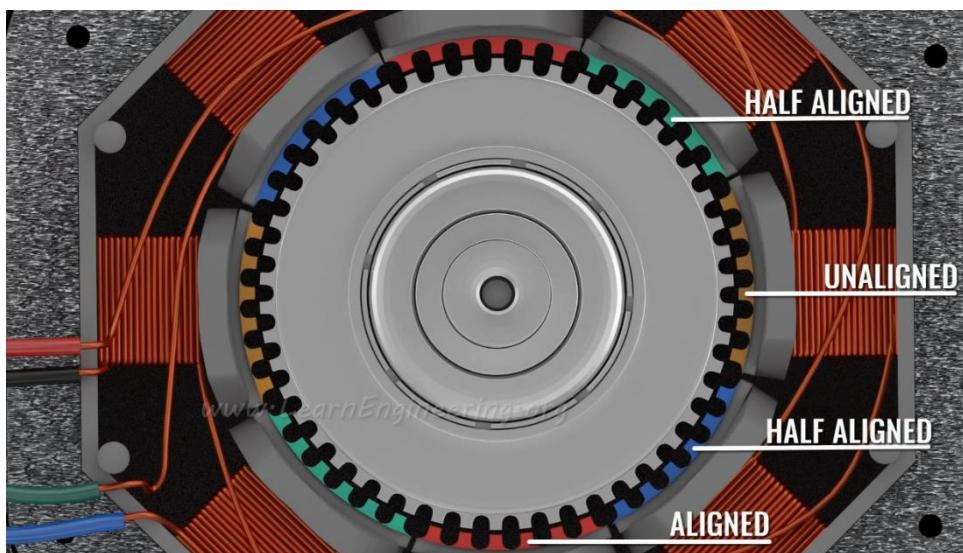
Ce moteur a 6 fils connectés à 2 bobines. Le jaune, noir et vert faisant partie de la première, et le rouge, blanc et bleu faisant partie de la deuxième bobine.

4.2.4. Fonctionnement du moteur



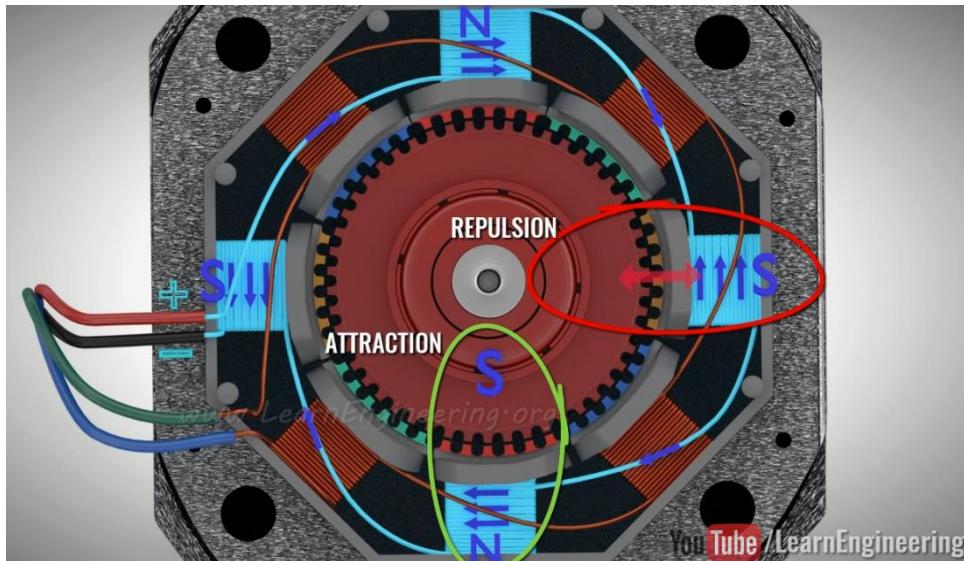
[Lien de l'image](#)

Il y a toujours plus de dents avec bobines que de dents du rotor. Pour tourner, la bobine A va être éteinte pour alimenter les bobines B, cela fera tourner le rotor vers la bobine B, ensuite, la bobine B sera éteinte pour allumer la bobine c et ainsi de suite.



[Lien de l'image](#)

Dans le cas du moteur pas à pas Nema 17, le rotor est composé de deux engrenages avec 50 dents chacun, les deux étant aimantées grâce à un aimant situé dans ces dernières, tandis que le stator procède 48 dans, rendant la majorité des dents pas alignées avec celle du rotor.



Le stator est composé de deux bobines. Quand la bobine A est alimentée, les pôles Nord vont attirer les dents du rotor et les pôles Sud vont repousser ces dernières, alignant les certaines dents du rotor avec celles du stator. Tout cela va engendrer un pas. Comme dans l'exemple précédent, lorsque la bobine B sera alimentée, elle engendrera le déplacement du rotor, permettant ainsi au rotor de faire un deuxième pas.

4.3. Le driver moteur

Pololu A4988



[Lien de l'image](#)

4.3.1. Description

Le driver moteur permet au processeur de commander le moteur pas à pas. C'est un amplificateur de courant. Il transforme les impulsions « step » en signaux de commande pour les bobines du moteur. Il protège le reste du circuit des pointes de courant du moteur à son démarrage, qui est causé par la bobine située à l'intérieur de ce dernier.

4.3.2. Caractéristiques

- Tension de sortie du moteur : 8V – 35V
- Tension d'entrée du driver moteur : 3V – 5,5V
- Courant continu par phase : 1A
- Courant maximal par phase : 2A
- Température maximale : 165 °C

4.3.3. Brochage



[Lien de l'image](#)

Alimentation

VMOT : connectée directement à la partie puissance du moteur (+ 12V).

VDD : connectée à l'alimentation du microprocesseur (+ 5V).

GND x2 : une masse à V_{mot} et une à VDD.

Entrées

ENABLE : C'est une patte active à l'état bas, le microprocesseur doit donc envoyer 0V pour activer le driver moteur. La patte est à 0 par défaut donc, à part si on la met sous tension, le driver moteur est toujours allumé.

STEP : Lorsqu'une impulsion lui est envoyée, le moteur fait un pas.

DIR : Envoi la direction voulue au moteur.

SLEEP : c'est une patte active à l'état bas, elle place le circuit en « sommeil » pour minimiser la consommation d'énergie.

RESET : C'est une patte active à l'état bas, elle réinitialise le circuit. C'est une patte en l'air, car il n'y a pas de résistance pull-Down, il faut donc la brancher à la patte sleep qui en possède une si on ne l'utilise pas.

MS 1, 2, 3 : Ils permettent la sélection de la résolution de pas à faire lors d'une impulsion sur STEP. Ils contiennent des Résistances pull-Down donc, si les pattes sont en l'air, le stepper-moteur n'avancera que pas par pas.

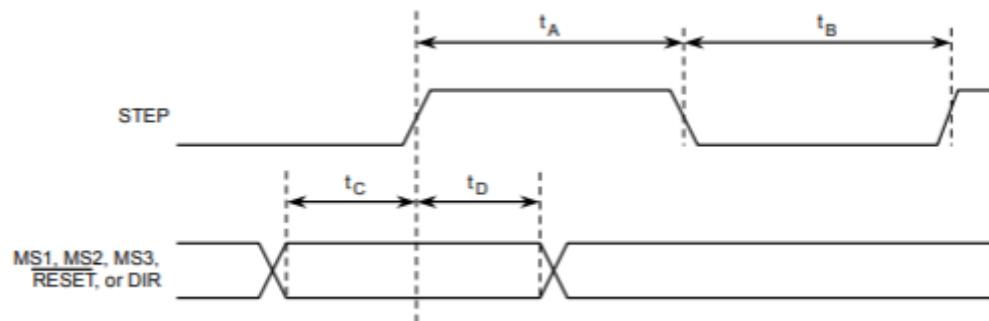
MS1	MS2	MS3	Resolution
LOW	LOW	LOW	Full Step
HIGH	LOW	LOW	Half Step
LOW	HIGH	LOW	Quarter Step
HIGH	HIGH	LOW	Eighth step
HIGH	HIGH	HIGH	Sixteenth Step

[Lien de l'image](#)

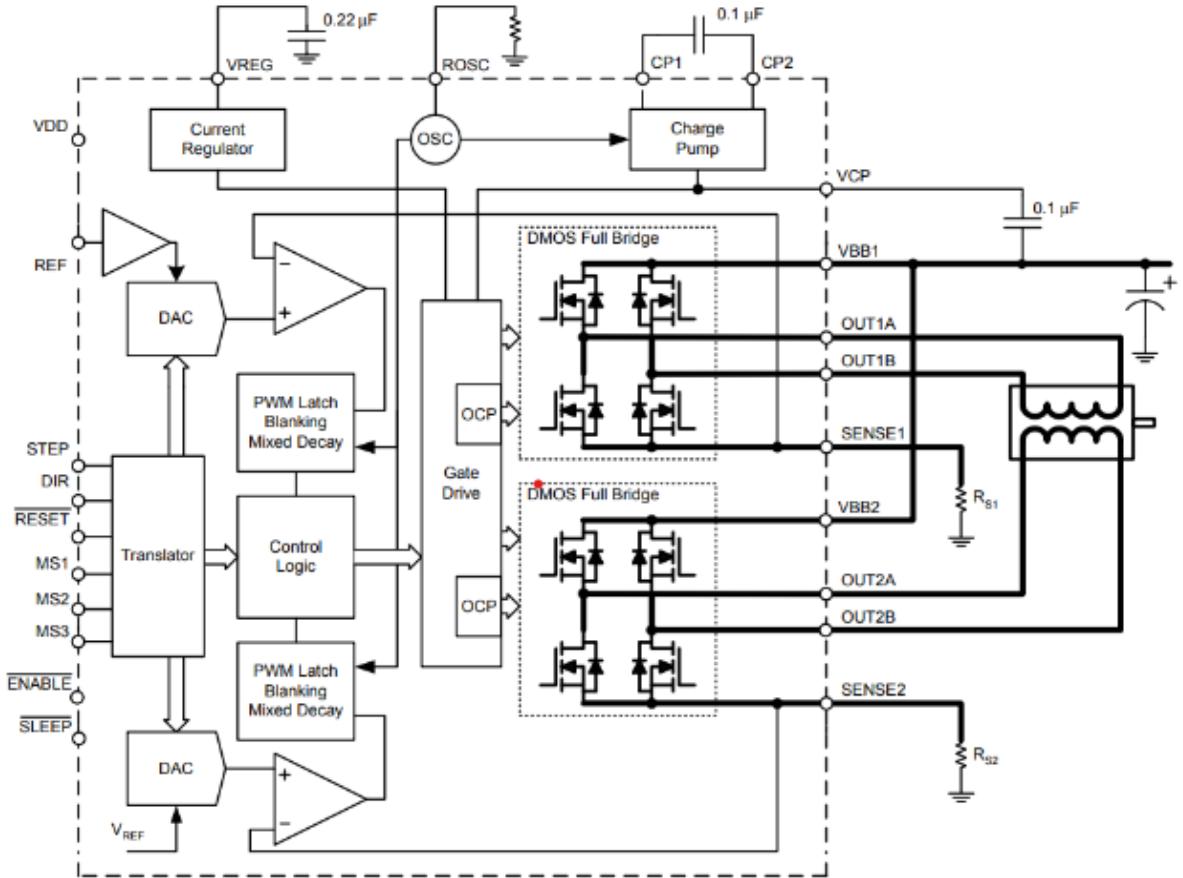
Sorties

1A, 2A, 1B, 2B : Connexions aux deux bobines du stepper-moteur

4.3.4. Fonctionnement du driver moteur



Time Duration	Symbol	Typ.	Unit
STEP minimum, HIGH pulse width	t_A	1	μs
STEP minimum, LOW pulse width	t_B	1	μs
Setup time, input change to STEP	t_C	200	ns
Hold time, input change to STEP	t_D	200	ns



Translator convertis les signaux reçus pour les envoyer dans le circuit.

DAC (digital-to-analog converter) convertis un signal continu en signal analogique.

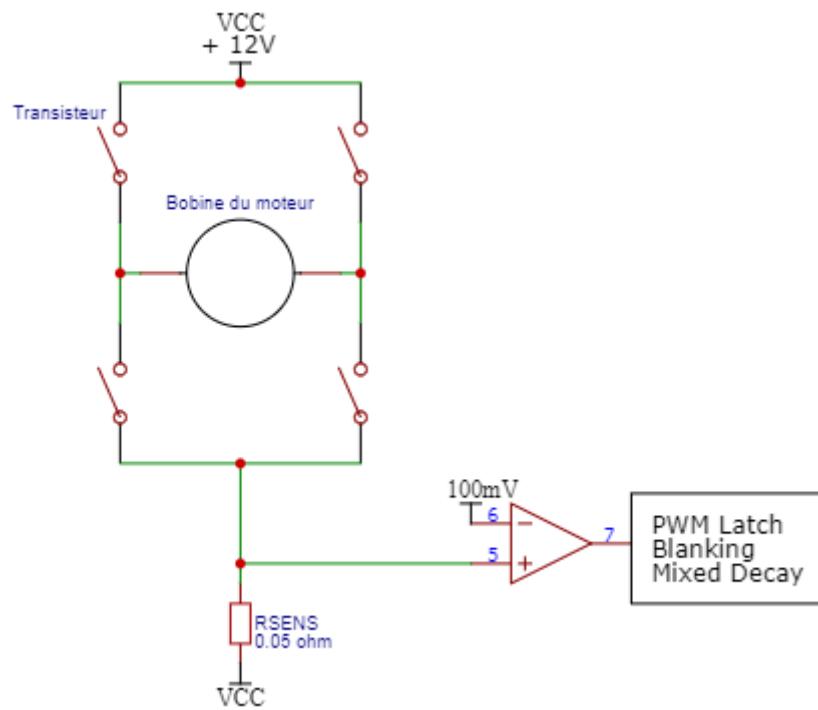
Gate Drive amplifie la puissance d'un contrôleur à faible puissance pour sortir à signal à haute tension pour des transistors à haute puissance comme les Power MOS-FET

OCP (Over Current Protection) protègent contre les potentiels effets dangereux de l'augmentation de tension venant du gate drive.

OSC charge et décharge le condensateur de Charge Pump.

Charge Pump fournit une tension de 2x12V pour les moteurs en charge le condensateur et le mettant en série avec l'alimentation par la suite. Les 24 V sont nécessaires pour piloter les grilles du MOS-FET.

4.3.5. Pont en H avec la résistance de mesure du courant



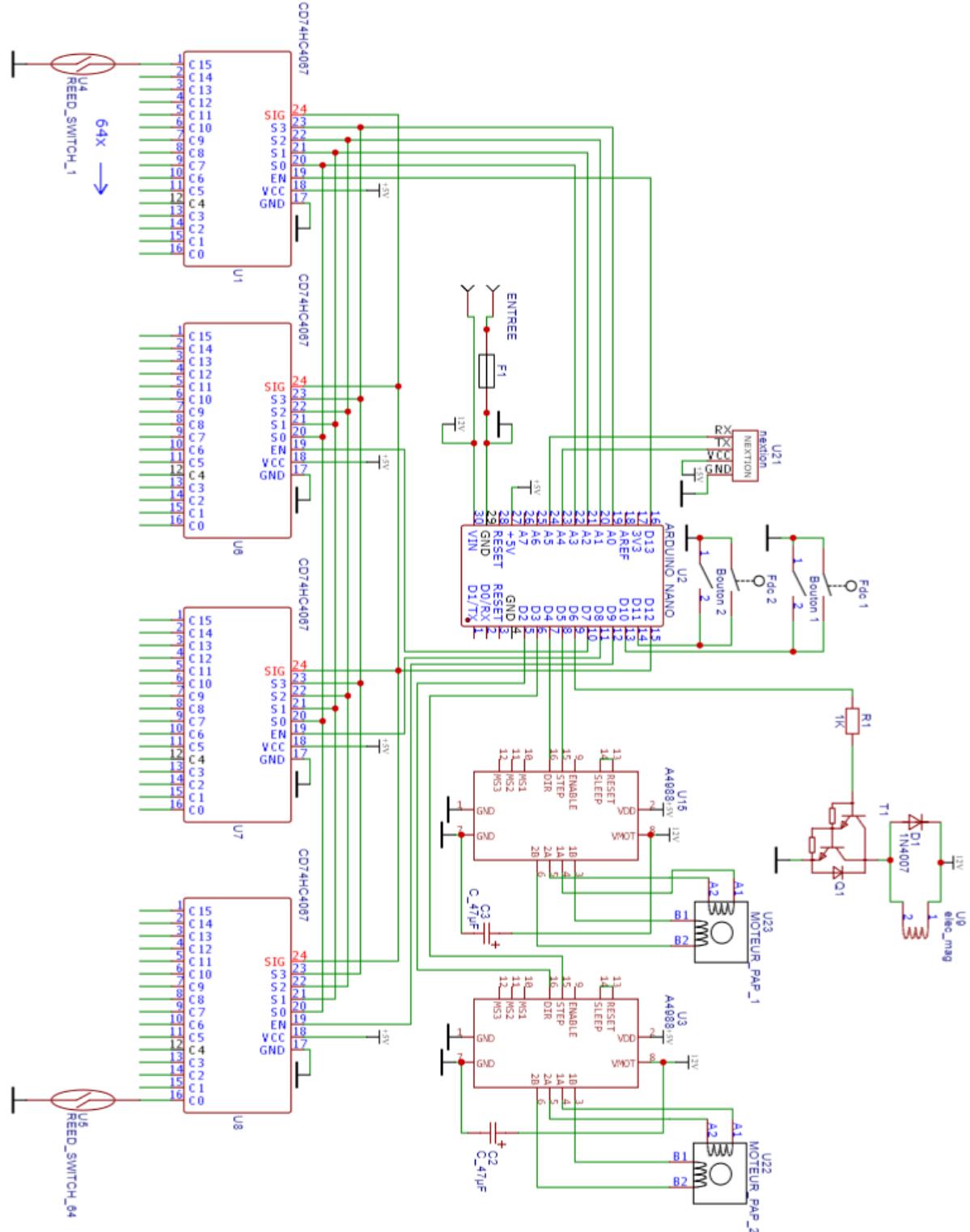
Dans cet exemple, la limitation de courant est à 2 A.

$$U = R_{\text{Sens}} * I_{\text{moteur}} = 0.05 * 2 = 100\text{mV}$$

R_{Sens} est une résistance de mesure du courant du moteur. Lorsque la tension à ses bornes dépasse une certaine valeur (100 mV), le comparateur bascule en envoyant 1 au PWM Latch Blanking Mixed Decay, qui déclenchera une procédure qui arrêtera le moteur.

5.L'étude détaillée des sous-ensembles

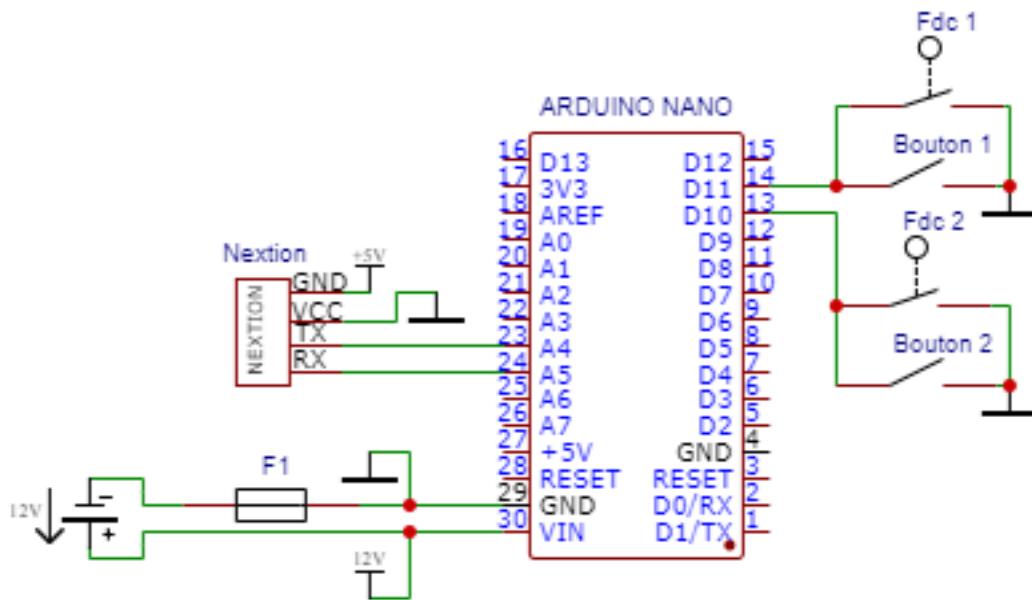
5.1 Schéma de principe



5.2 Interface utilisateur

5.2.1. L'écran graphique Nextion

Ces pattes Rx Tx sont connectées aux pattes A4 et A5 du microprocesseur. Comme le moteur communique en UART, les pattes connectées entre les composants sont censées être connectées en Rx Tx, ce qui n'est pas le cas pour moi. À cause de cela, je dois simuler des pattes Rx Tx dans le code de mon projet, afin de communiquer correctement avec l'écran. Voir le point « 4. Principes mis en jeu » pour voir l'explication, programmation et les mesures de l'écran.



5.2.2. caractéristiques de l'Arduino Nano :

- Microcontrôleur ATmega328
- 32 kB dont 2 kB utilisés par le bootloader
- 2 kB de SRAM interne
- 1kB EEPROM
- 32 x 8 Registres
- Compteur en temps réel avec oscillateur à part
- 6 chaînes PWM
- USART programmable

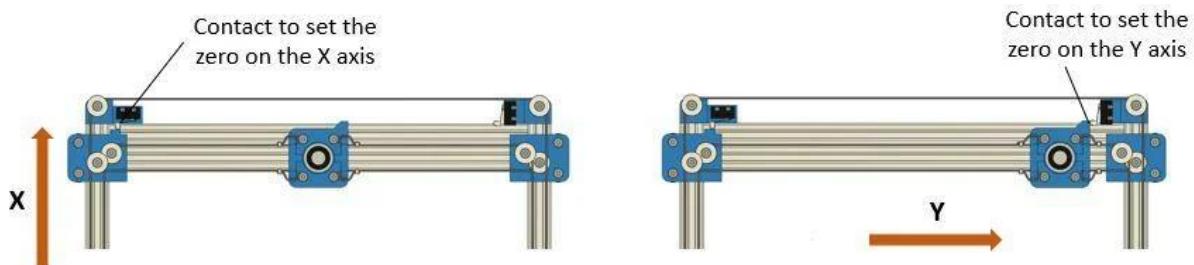
Alimentation

- Connexion USB type C
- Alimentation externe irrégulière entre 6-20V (pin 30)
- Alimentation régulière de 5V (pin 27)

E/S

- 22 Numériques
- 8 Analogiques
- 6 Sorties PWM

5.2.3. Les fins de course



[Lien de l'image](#)

Les fins de course sont utilisées pour la remise à zéro de la position de l'électro-aimant. Pour cela, ils sont placés aux extrémités des axes x et y. afin de faire la remise à 0, je fais avancer l'électro-aimant vers la fin des deux axes x et y.

5.2.4. Le fusible

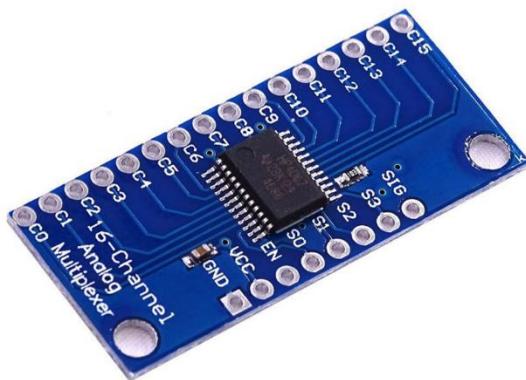
Mon projet consomme en moyenne 800mA, j'ai donc choisi un Fusible de 1A temporisé, qui tolère les pointes de surintensité, car j'utilise des moteurs qui créent des pointes de courant lors de leur alimentation à cause des bobines à l'intérieur de ces-derniers.

5.2.5. Programme de test pour les fins de course

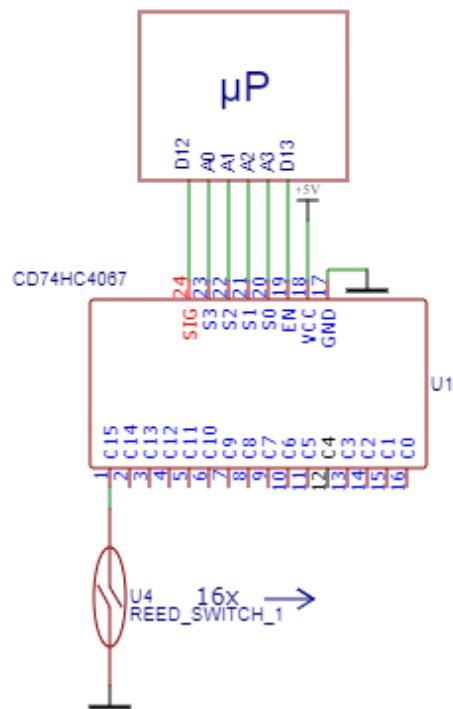
```
#define FCA 11//fin de course A
#define FCB 10//fin de course B
void setup() {
    pinMode(FCA, INPUT_PULLUP);
    pinMode(FCB, INPUT_PULLUP);
}

void loop() {
    if (!digitalRead(FCA)) {
        Serial.println("La fin de course A a été actionnée");
    }
    if (!digitalRead(FCB)) {
        Serial.println("La fin de course B a été actionnée");
    }
}
```

5.3. Le multiplexeur



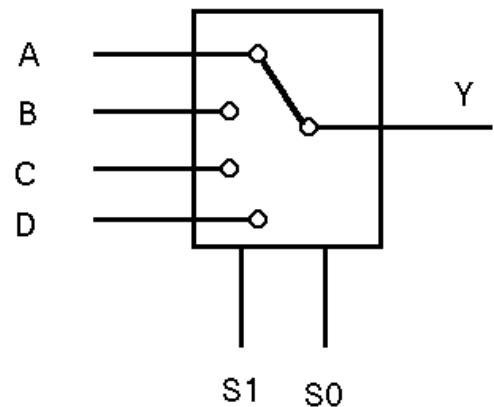
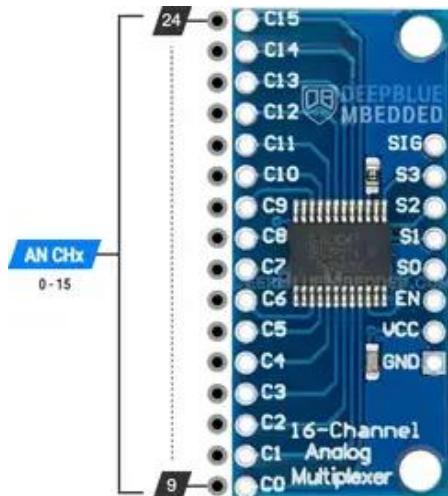
[Lien de l'image](#)



5.3.1. Description

Le multiplexeur va lire les Reed switch chacun connecté à une des entrées (C0 à C15) et renvoyer la donnée au microprocesseur depuis le pin SIG (signal) pour savoir si le pion se trouve ou pas sur une position spécifique. Pour ce faire, le microprocesseur va servir de compteur pour lire les 16 entrées et ça très rapidement.

Le MUX/DEMUX CD74HC4067 a deux manières de fonctionner : en tant que multiplexeur ou démultiplexeur. Dans le cas de mon projet, je l'utilise en tant que multiplexeur. Afin de l'utiliser le CD74HC4067 en tant que multiplexeur, il faut déclarer la patte SIG en tant qu'entrée et envoyer une donnée sur les entrées C0-C15.



[Lien de l'image](#)

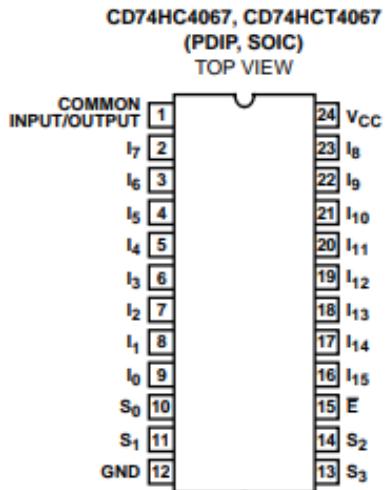
CD74HC4067 Module Pinout

En utilisant une des entrées S0-S4, on choisit les pattes à lire. La donnée envoyée sur la patte sélectionnée va être envoyée sur la patte SIG. La patte E. est active à l'état bas, il faut donc la brancher à la masse pour utiliser le multiplexeur. Il fonctionne à 5V.

5.3.2. Les dimensions

12 mm x 40 mm x1.3mm

5.3.3. Le brochage



VCC : connecté au +5V du µP

GND : connecté à la masse du µP

Entrées

Entablé C'est une patte à l'état bas ce qui veut dire qu'il faut envoyer un 0 pour l'activer
I0 – I15 ces pattes lisent les capteurs et envoient la donnée reçue sur la patte SIG
S0 – S3 ces pattes choisissent quelles entrées lire (de I0 à I15).

On va utiliser le µP comme compteur pour définir quelle patte lire.

Sorties

SIG : cette patte envoie la donnée lue sur une des 15 entrées I0-I15 au µP

TRUTH TABLE

S0	S1	S2	S3	\bar{E}	SELECTED CHANNEL
X	X	X	X	1	None
0	0	0	0	0	0
1	0	0	0	0	1
0	1	0	0	0	2
1	1	0	0	0	3
0	0	1	0	0	4
1	0	1	0	0	5
0	1	1	0	0	6
1	1	1	0	0	7
0	0	0	1	0	8
1	0	0	1	0	9
0	1	0	1	0	10
1	1	0	1	0	11
0	0	1	1	0	12
1	0	1	1	0	13
0	1	1	1	0	14
1	1	1	1	0	15

NOTE:

H = High Level

L = Low Level

X = Don't Care

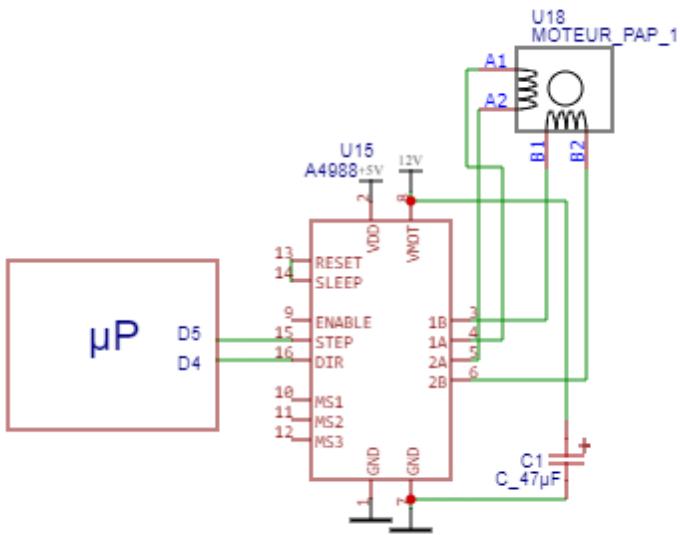
5.3.4. Programme de test

```
#define SIG 12
#define EN1 13
int SIGVAL;

void setup() {
    pinMode(SIG, INPUT_PULLUP);
    pinMode(EN1, OUTPUT);
    digitalWrite(EN1, 0);
    Serial.begin(9600);
    while (!Serial) {}
    DDRC |= 0B00001111;

    for (int i = 0; i < 16; i++) {
        PORTC = (PORTC & 0B11110000) | (i & 0B00001111); //envoi
        des données
        SIGVAL = digitalRead(SIG); //binaires sur le PORTC en
        parallèle
        Serial.print(SIGVAL);
        Serial.print(<< n° >>);
        Serial.println(i);
        delay(500);
    }
}
```

5.4. Les moteurs pas-à-pas



Voir les principes mis en jeu pour avoir l'explication complète du driver moteur et moteur pas à pas.

5.4.1. Rôle du condensateur C1

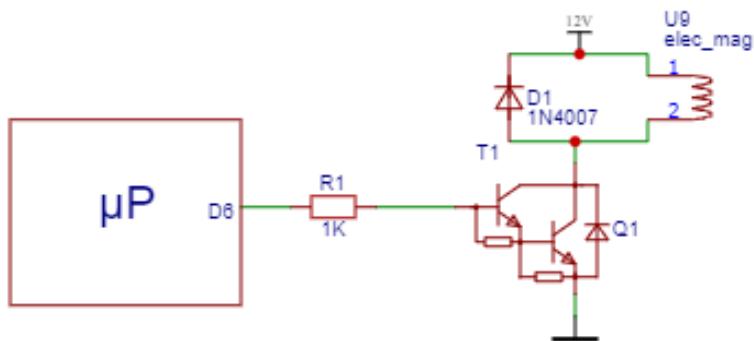
Il fait office de «réserve» de courant pour le moteur pas à pas. En effet, ce dernier consomme beaucoup de courant surtout le moteur, qui devient circuit fermé lors de son démarrage.

5.4.3. Programme de test

```
#define STEP 5
#define DIR 4
void setup() {
    pinMode(STEP, OUTPUT);
    pinMode(DIR, OUTPUT);
}

void loop() {
    digitalWrite(DIR, 1);
    for (int x = 0; x < 200; x++) {
        digitalWrite(STEP, 1);
        delayMicroseconds(1200);
        digitalWrite(STEP, 0);
        delayMicroseconds(1200);
    }
}
```

5.5. L'électro-aimant



[Lien de l'image](#)

5.5.1. Description

L'électro-aimant va s'activer pour, avec l'aide des moteurs, déplacer les pièces sur la planche de jeu. Il peut se déplacer sur deux axes, horizontalement et verticalement. Il est relié au microprocesseur par un transistor TIP120, une diode et une résistance.

5.5.2. Les dimensions

2,5 cm de diamètre et 2 cm de hauteur.

5.5.3. La fonction des signaux

Alimentation :

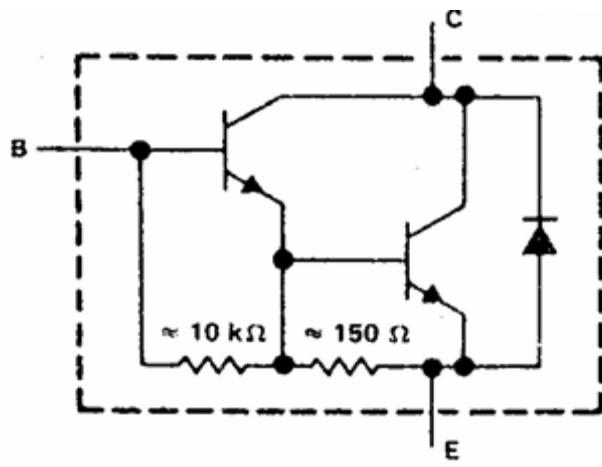
VCC : alimentation (+ 12V).

GND : Connectée à la masse du circuit.

5.5.4. Caractéristiques

- Tension : 12V DC
- Force de maintien : 5 kg/50N
- Température de fonctionnement maximale : 103 °C
- Taille : 25x25x20 mm
- Longueur du câble : 200 mm
- Consommation : +- 500 mA

5.5.5. Le TIP 120

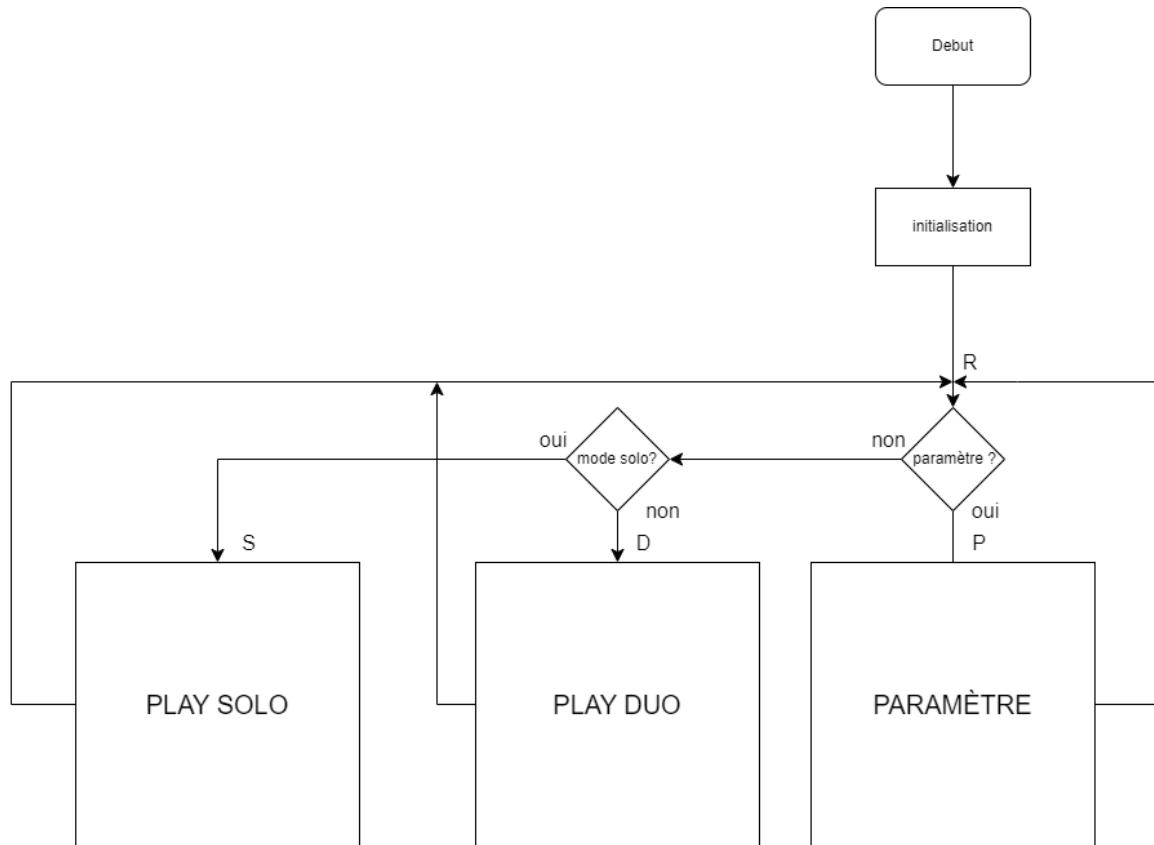


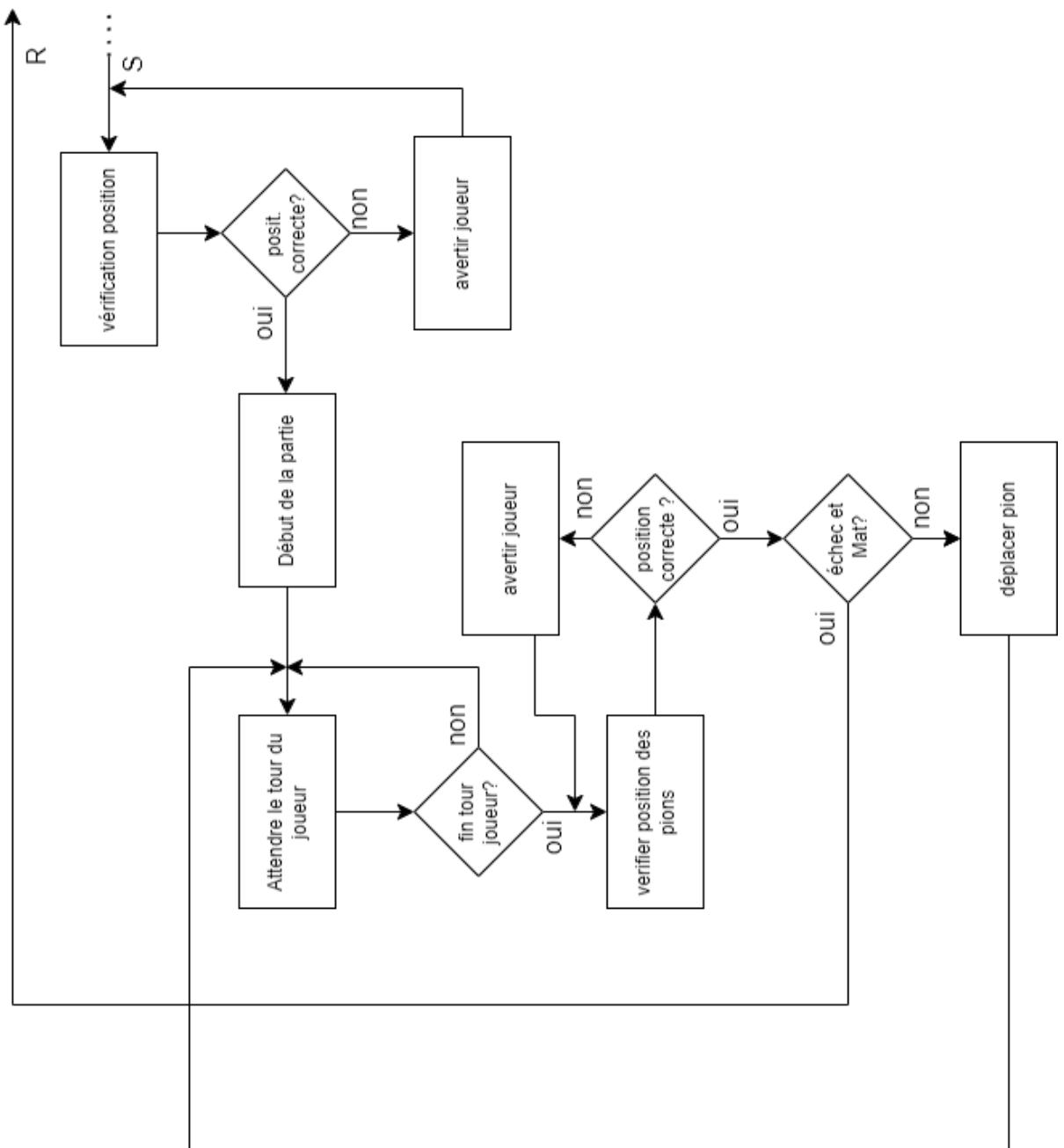
Lien de l'image

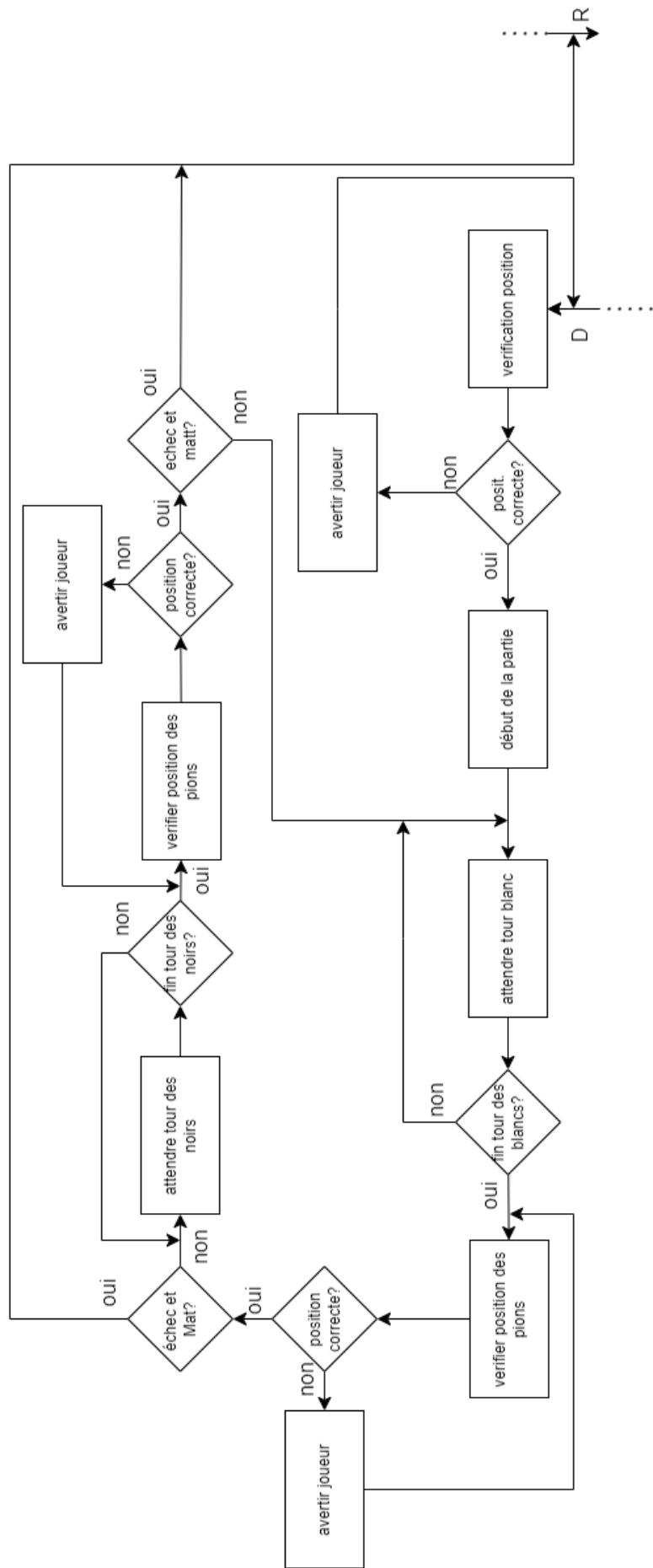
Le TIP 120 est un transistor Darlington, je l'utilise pour augmenter le courant dans l'électro-aimant. Il est composé de deux transistors NPN, 2 résistances et une diode de roue libre. Cette disposition de transistors permet d'avoir beaucoup de courant dans I_C , pouvant alimenter des composants qui en demandent plus que ce que l'alimentation peut fournir. La température risque de vite monter vu qu'il y a beaucoup de courant. Cependant, pour son utilisation dans mon projet, la température ne devrait pas trop augmenter même si j'ai prévu un emplacement pour un refroidisseur en plaçant le transistor sur le bord du PCB.

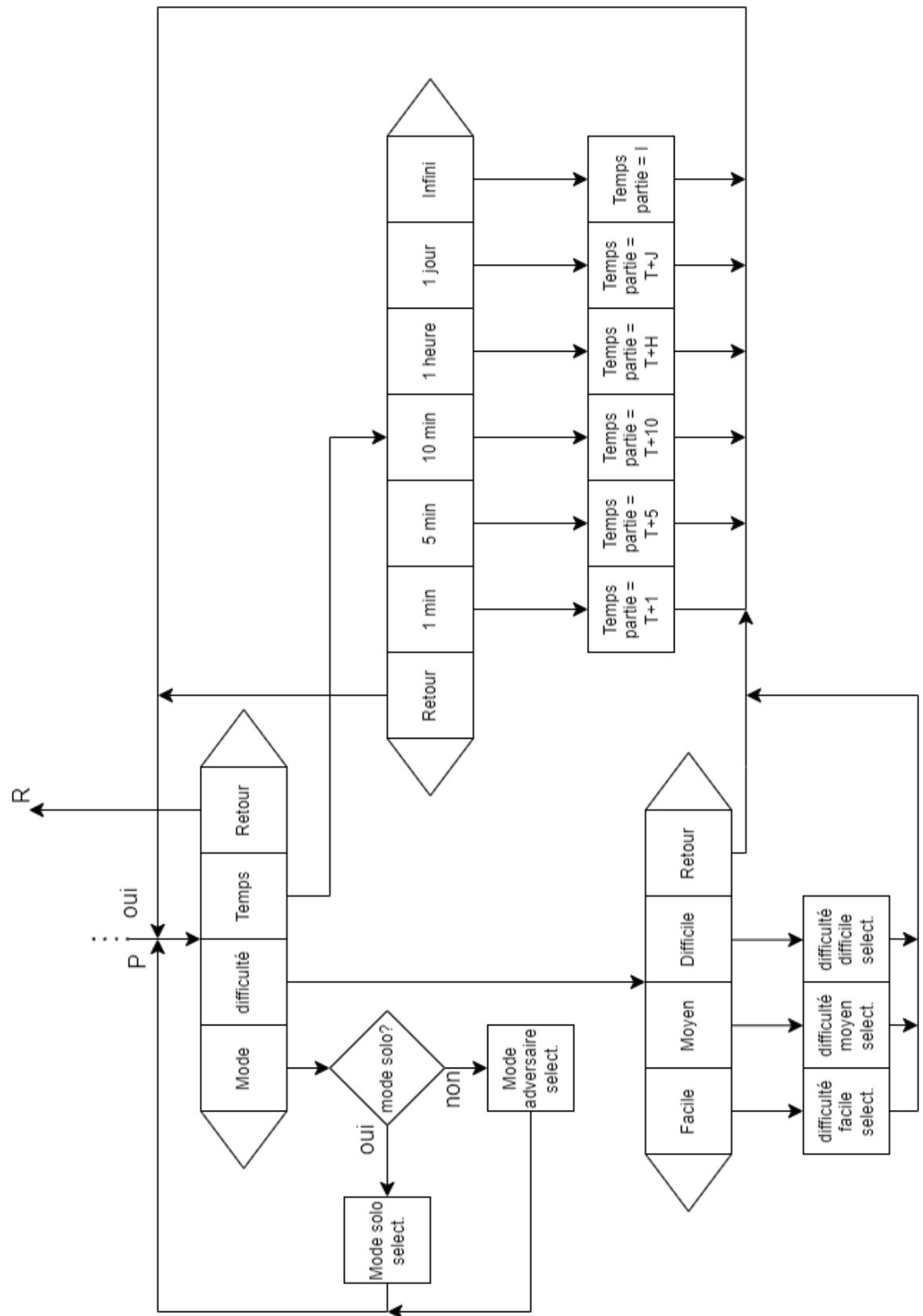
6. La programmation

6.1. L'ordinogramme général









6.2. Le programme

Comme le programme de mon projet est conséquent (environ 600 lignes), j'ai décidé de faire un lien GitHub, permettant la lecture de ce dernier plus simple. Je vous invite donc à consulter mon programme sur le lien suivant : <https://github.com/DawidSac24/AutoMate>

7.La fabrication

7.1. Liste des pièces requises

7.1.1. Pièces mécaniques

- 4 x Rails en V 20x20mm longueur :
 - o 395 mm (côté fin de course)
 - o 375 mm x2 (sur les côtés)
 - o 350 mm (celui du milieu)
 - o 315 mm (côté moteurs)
- Boitier en bois contreplaqué :
 - o 2 x Face 470 x 470 x 10 mm
 - o 1 x Face 450 x 85 x 10 mm
 - o 1 x Face 470 x 85 x 10 mm
- 10 x supports d'angles à 90°
- 2 x Poulie GT2, 20 dents
- 8 x Poulie GT2 sans dents
- 1 x ceinture GT2 de 3,5 m
- 1 x pièces de jeu d'échec – Staunton 3
- 32 x aimants – diamètre 8 x 3 mm

7.1.2. Pièces électroniques

- 1 x Arduino Nano
- 2 x Moteur pas à pas – Adafruit – Nema 17 200 pas/tr, 12V 350 mA
- 2 x Driveurs moteur pas à pas – Pololu – A4988
- 1 x Écran graphique Nextion
- 1 x Électro-aimant – force de maintien de 5Kg
- 1 x Diode de roue libre – 1N4007
- 1 x Transistor Darlington TIP120
- 2 x Fins de courses
- 64 x Reed Switch – 2 x 14.5 mm
- 1 x résistance de 1KΩ
- 4 x Multiplexeur – CD74HC4067

- 1 x connecteur femelle - DC Jack
- Barrettes sécables femelles

7.1.3. Visserie

- 8 x Nini roue en V
- 4 x Entretoise en aluminium - ID : 5 mm - Hauteur : 6 mm
- 4 x entretoise excentrique - M5
- 4 x hexagonaux. Écrou de blocage - M5
- 4 x hexagonaux fins. Écrou - M5
- 8 x Mini Cale de Précision - OD : 8 mm - ID : 5 mm - Epaisseur : 1 mm
- 26 x écrous en T - M5
- 14 x hexagonaux. Vis à tête cylindrique à six pans creux - M5 x 8 mm
- 12 x hexagonaux. Vis à tête cylindrique à six pans creux - M5 x 12 mm
- 2 x hexagonaux. Vis à tête cylindrique à six pans creux - M5 x 35 mm
- 2 x hexagonaux. Vis à tête cylindrique à six pans creux - M5 x 30 mm
- 2 x hexagonaux. Vis à tête cylindrique à six pans creux - M5 x 25 mm
- 8 x hexagonaux. Vis à tête cylindrique à six pans creux - M5 x 15 mm
- 4 x vis à tête fendue - M2 x 15 mm
- 4 x écrou - M2
- 8 x hexagonaux. Vis à tête creuse - M3 x 10 mm
- 1 x hexagonal. Vis à tête creuse - M4 x 15 mm
- 4 x vis à tête fraisée fendue - M5 x 20 mm
- 4 x écrou - M5
- 128 x vis en laiton – M2 x 12mm

7.2. Modèles en 3D

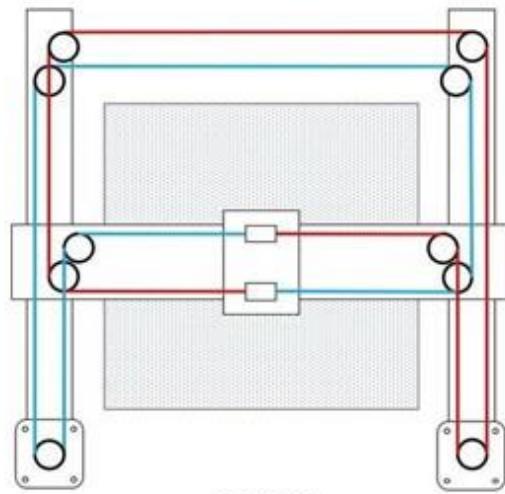
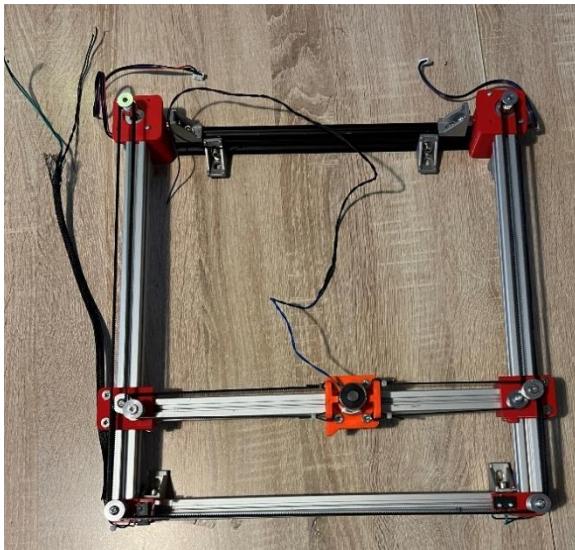
Téléchargez les modèles 3D pour l'impression

A cause de la complexité de mon projet et avec l'accord de mon promoteur, j'ai pu utiliser certaines pièces en 3D venant d'Inscrutable.

7.3. Assamblage



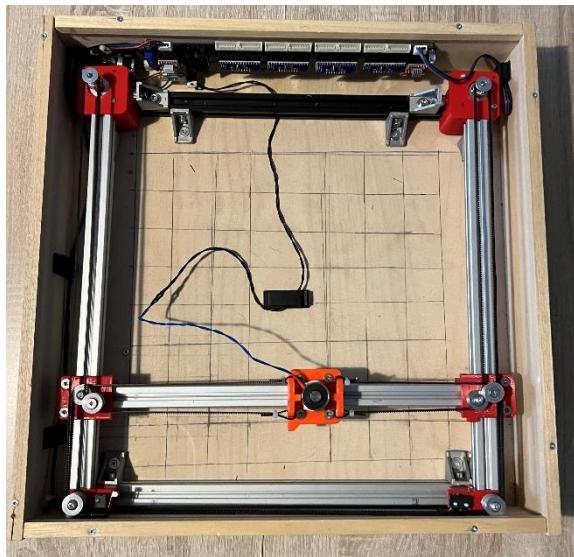
Coupez les plaques en bois avec les bonnes dimensions en utilisent les mises en plan (voir annexes) et assemblez-les avec des vices à bois.



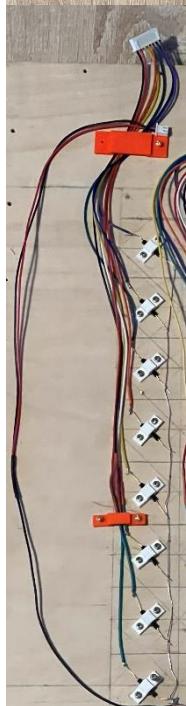
Ensuite, fixez le PCB avec des entretoises. Assemblez les rails entre eux avec des supports d'angles à 90 degrés. Placez et fixez les pièces imprimées en 3D et placez le rail central.

Ensuite, fixez l'électro-aimant sur son support et vissez les roues de ce-dernier. Fixez les roues à poulies sur leurs supports dédiés. N'oubliez pas de fixer et souder les fins de courses.

Préparez un câblage propre avec des gaines avant la fixation des rails.



Enfin, vous pourrez placer et fixer les rails assemblés dans le boîtier avec des supports d'angles à 90 degrés.



Pour la face du dessus, commencez par fixer les capteurs à aimants avec leurs supports imprimés en 3D avec des vices en laiton. Ensuite, soudez la masse de ces-derniers avec un fil mono-brun entièrement dénudé. Soudez la masse à un connecteur que vous placerez par la suite sur le PCB. Utilisez les supports à fils imprimés en 3D pour fixer le câblage et soudez le connecteur à 8 fils aux extrémités des capteurs à aimants. Reproduisez la procédure pour chaque colonne de l'échiquier.

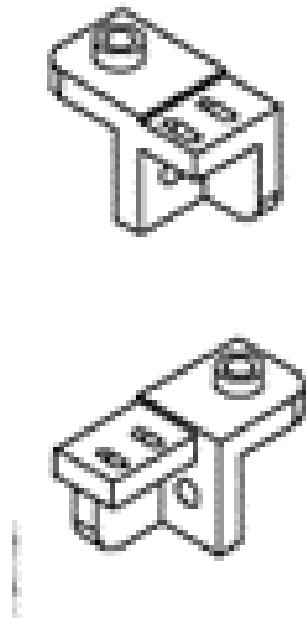
8.La mise au point

8.1. Les problèmes rencontrés et les solutions

I. La fixation des rails en V

Pour fixer les rails en V, je me suis inspiré de celle du projet de jeux d'échecs sur Instructable. Cependant, j'ai décidé d'utiliser un rail en moins (celui entre les deux moteurs) en pensant que cela ne poserait pas de problèmes. Après la fixation de ce dernier, je me suis rendu compte que les rails étaient difficiles à fixer droit et ils n'étaient pas assez stables lors du déplacement de l'électro-aimant. Pour résoudre ce problème, j'ai simplement acheté les rails manquants.

II. Le manque de place dans le boitier

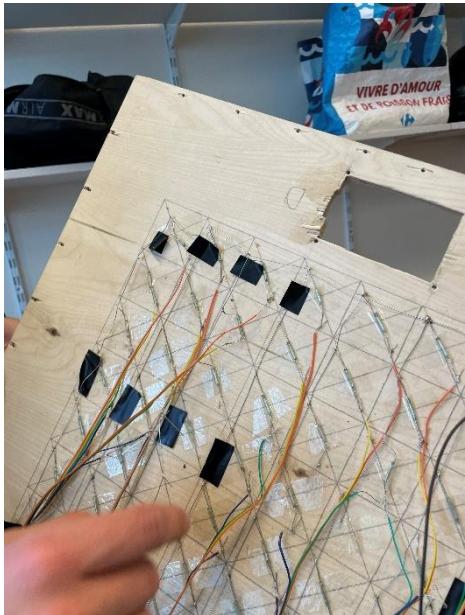


Lors de la conception du boitier, j'ai essayé d'optimiser mon espace utilisé dans le boitier, pour que ce dernier n'est pas trop grand et pour avoir les cases des pions les plus grands possibles, afin d'être sûr que les pions puissent avoir suffisamment d'espace à contourner d'autres pions. Cependant, je n'ai pas prévu l'espace que prenaient mes supports à fin de course. J'ai dû donc en créer une nouvelle, prenant un peu d'espace sur les rails. De plus, j'ai aussi manqué d'espace pour un emplacement des pions éliminés, l'électro-aimant ne pourra donc pas placer les pions sur les côtés lui-même. Pour résoudre ce problème, je devrais réduire la taille des cases, ce qui pourrait gêner le déplacement des pions lorsqu'ils les contournent.

III. La hauteur du boitier

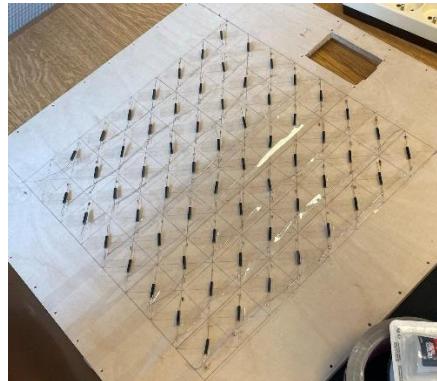


J'ai opté pour du contre-plaquée pour le boitier de mon projet. Après fixation, je me suis rendu compte que les pièces n'étaient pas découpées correctement, rendant la face du dessus plus bas à un endroit qu'à un autre. À cause de cela, l'électro-aimant percutait les capteurs à aimants et ainsi que les fils auxquels ils étaient soudés. Afin d'assurer le déplacement de l'électro-aimant, j'ai surélevé la face du dessus avec des « lattes » en bois de 12x12 mm.



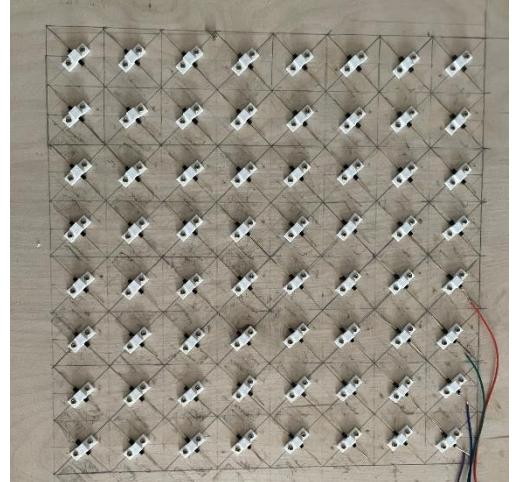
IV. les capteurs à aimants

En premier lieu, j'avais utilisé des capteurs en verre. Cependant, ces derniers n'étaient pas solides et ne détectaient pas suffisamment les aimants à travers la face du dessus. J'ai donc commandé des capteurs à aimants en plastiques, plus sensibles aux aimants et moins fragiles.



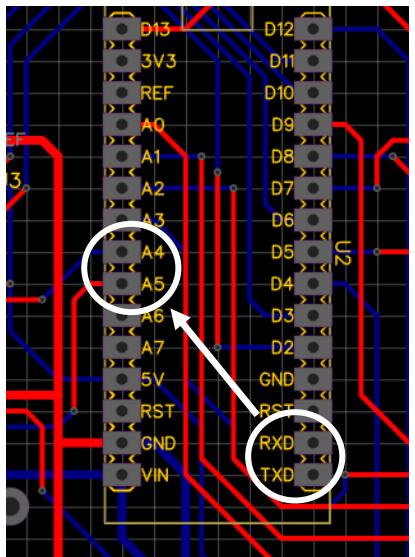
V. les supports des capteurs à aimants

Pour la fixation des capteurs à aimants, j'ai d'abord utilisé du scotch pour pouvoir facilement changer les composants en cas de panne éventuelle. Cependant, je me suis aperçu que cette solution était très peu fiable. J'ai donc opté pour des supports à imprimer en 3 d, fixés par 2 visse de chaque côté.



VI. Les vices des supports des capteurs à aimants

Pour la fixation des supports des capteurs à aimants, j'ai utilisé des vices en métal. Cependant, les aimants des pièces d'échecs étaient trop attirés par les vices. Par conséquent, les vices ne pouvaient pas être déplacés par l'électro-aimant. J'ai donc décidé de commander des vices en laiton, afin de ne plus bloquer le déplacement des pions.



VI. la conception du PCB

Lors de la conception de mon PCB, j'ai mal connecté le Rx et Tx de l'écran Nextion au microprocesseur. Comme l'écran communique en UART, j'ai dû utiliser la librairie [SoftwareSerial](#) pour faire communiquer les deux modules entre eux. Suite à ça, je ne pouvais plus utiliser la librairie de l'écran Nextion, ce qui m'a obligé à communiquer avec le module manuellement. Après réflexion, je me suis rendu compte que je pouvais plus facilement aménager mon code. Au final, sans cette erreur je ne me serais pas rendu compte du fonctionnement de la librairie [Nextion](#) ce qui m'a permis d'avoir un meilleur code qu'avec cette dernière.

8.2. Les améliorations à réaliser

During the realization of my project, I had many ideas for my project, which would have allowed me to improve it. Unfortunately, I did not have enough time to implement them. That's why I decided to present them below:

- Je pourrais utiliser une ESP-32 pour pouvoir communiquer avec le projet de mon camarade, lui aussi, faisant un jeu d'échecs automatisé. Ça permettrait à deux personnes de jouer sur une vraie planche d'échecs à distance. De plus, grâce à l'ESP-32, je pourrais me connecter à un site de jeux d'échecs en ligne, afin de jouer contre des joueurs en ligne ou des intelligences artificielles proposées sur certains sites.
- J'aurais pu ajouter deux fins de courses supplémentaires pour protéger le système de courroie lors du déplacement de l'électro-aimant. Ils seraient actionnés lorsque l'aimant atteint un des bords du rail de milieu, ce qui déclencherait une remise à 0 de sa position. Ensuite, je pourrais ajouter un système de double vérification, en plaçant un ampèremètre qui déclencherait une remise à 0 lorsque l'électro-aimant se bloquerait sur un des rails.

9. La conclusion

Durent la réalisation d'Auto-mate, j'ai pu apprendre beaucoup de choses. J'ai commis beaucoup d'erreurs et j'ai pu trouver des solutions à ces-dernières, ce qui m'a beaucoup enrichi. J'ai amélioré mes compétences en codage, acquis des connaissances en électronique supplémentaires à celles vues durant mes deux dernières années d'études. Je n'ai malheureusement pas pu faire tout ce que je voulais même si je suis très content de mon avancement ainsi que du résultat final de mon projet.

10. Bibliographie et webographie

- Source principale pour l'inspiration et la conception du projet :

Automated-Chessboard. [29.05.2024]. <https://www.instructables.com/Automated-Chessboard/>

- Source utilisée pour les principes mis en jeu :

Definition de tft. [29.05.2024]. <https://www.lcd-compare.com/definition-de-tft.htm>

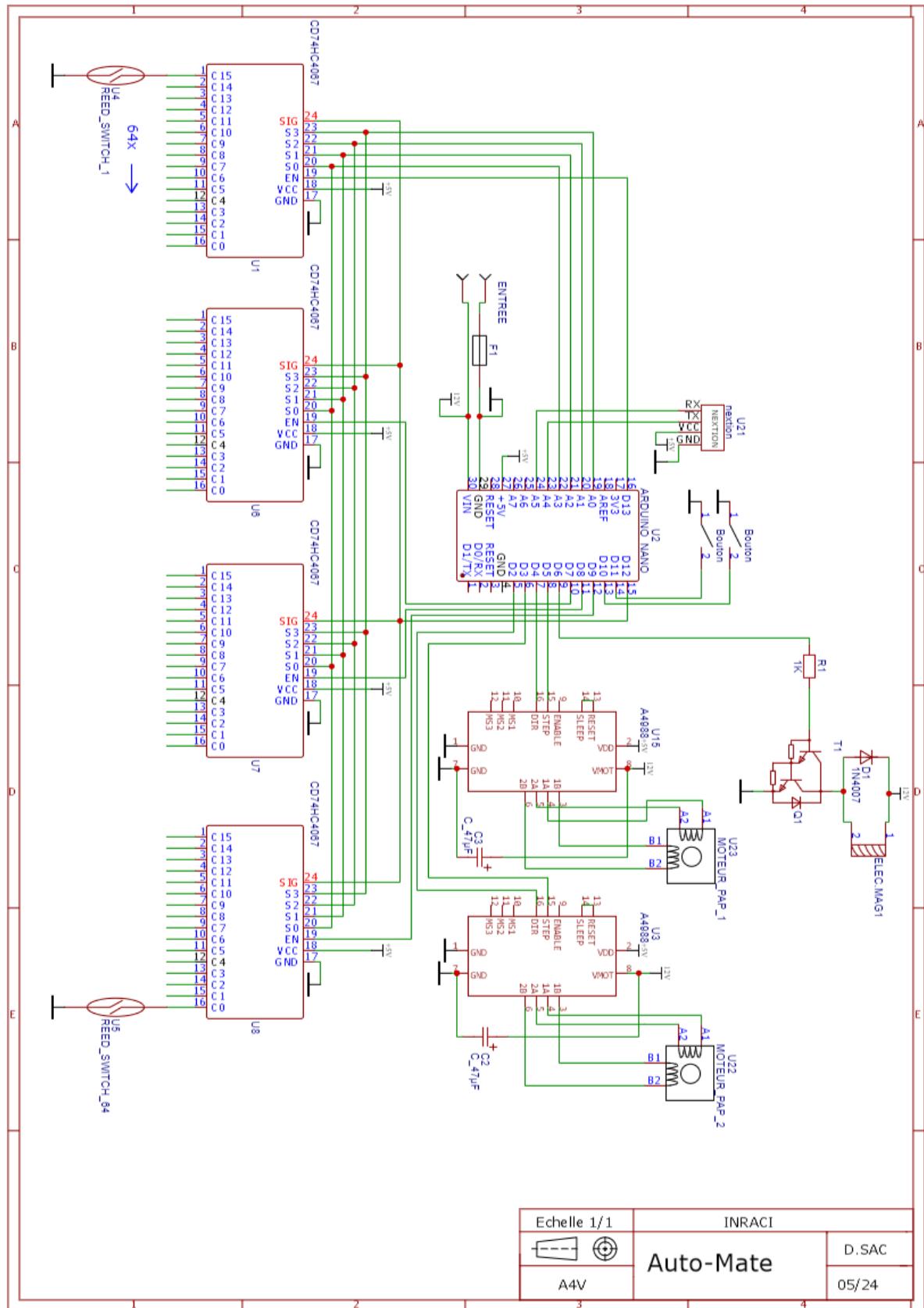
resistif ou capacitif quel-est le bon ecran tactile. [29.05.2024].

<https://riverdi.com/fr/blog/resistif-ou-capacitif-quel-est-le-bon-écran-tactile>

Pour le reste des explications, j'ai utilisé les datasheets des composants utilisés ainsi que les cours et explications en classe de Mr. Mazzeo et Mr. Block.

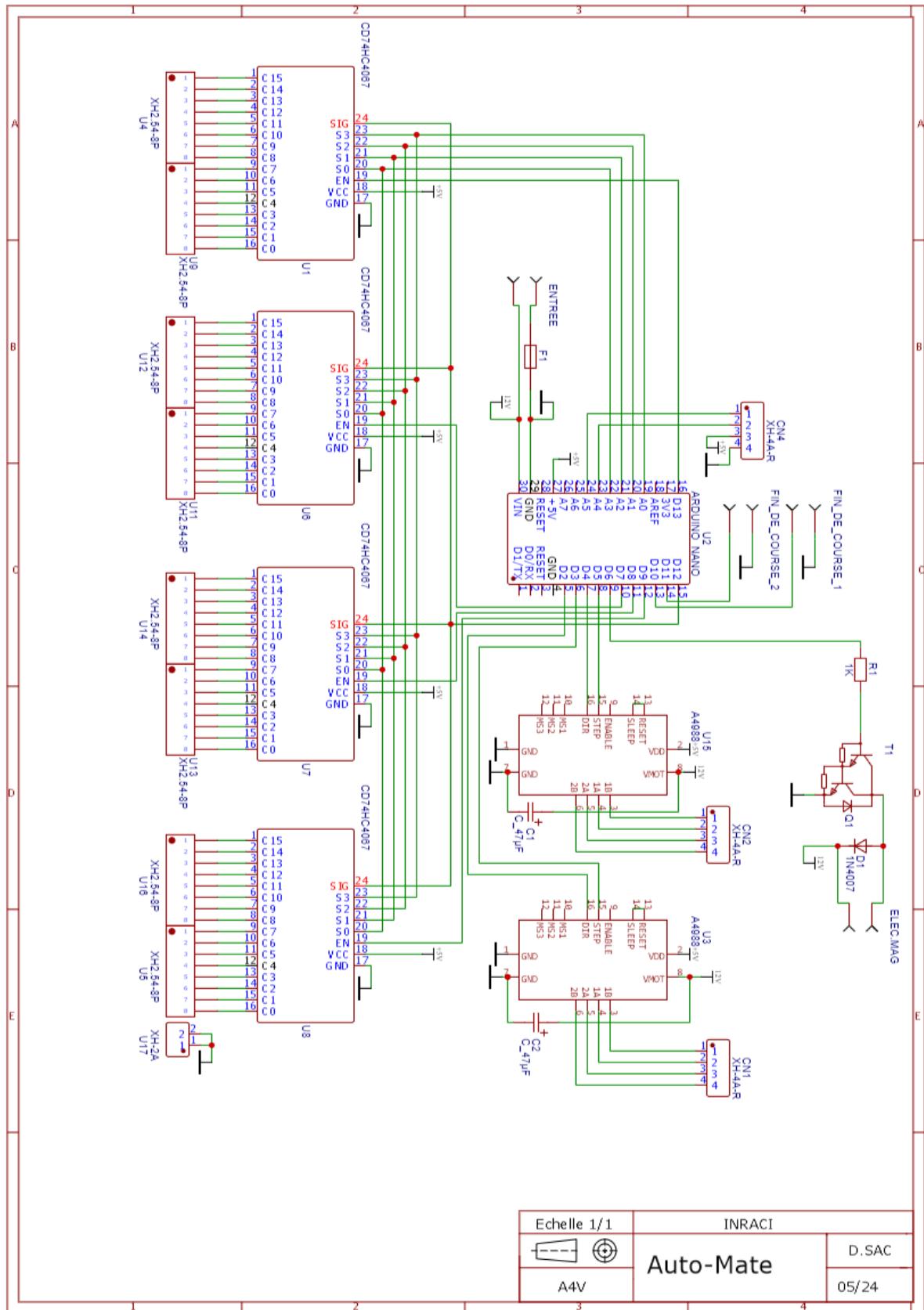
11. Les annexes

I. Le schéma de principe sans connecteur

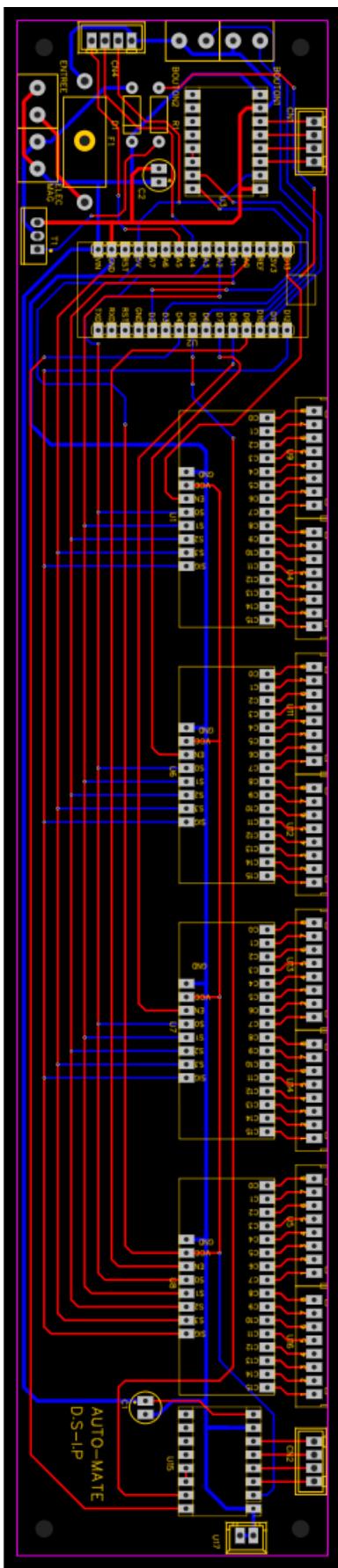


Echelle 1/1	INRACI	D.SAC
Auto-Mate		
A4V	05/24	

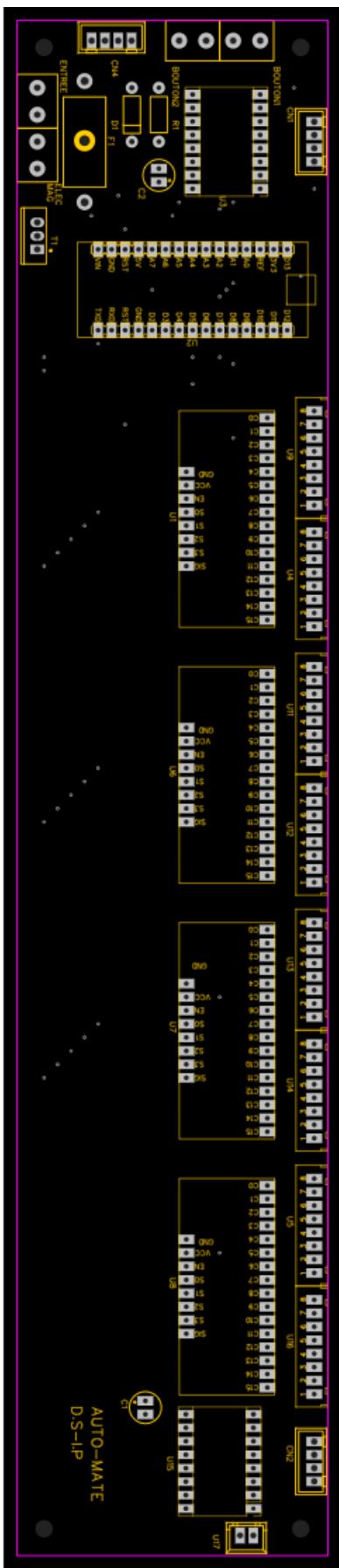
II. Le schéma de principe avec connecteur



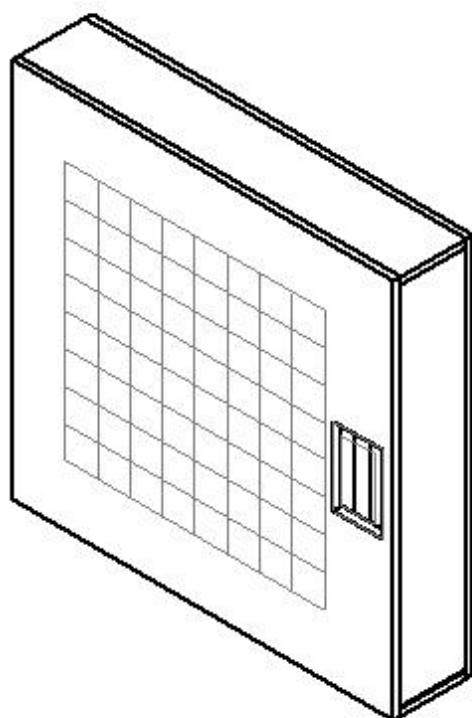
III. PCB



IV. PCB sérigraphie



VII. Vue 3D



ECHELLE 1/5

Inraci

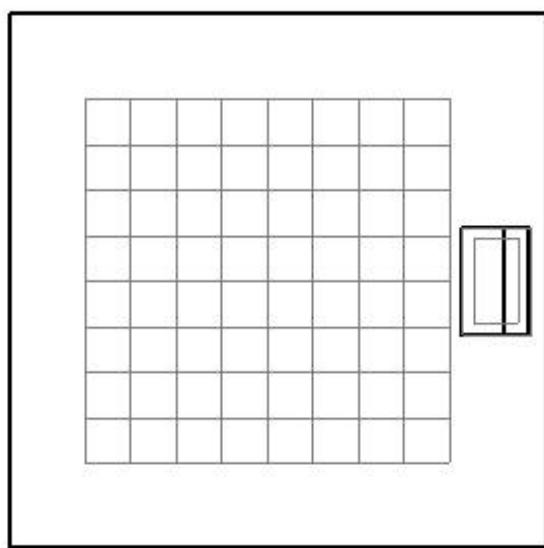


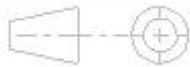
A4

Vue d'ensamble

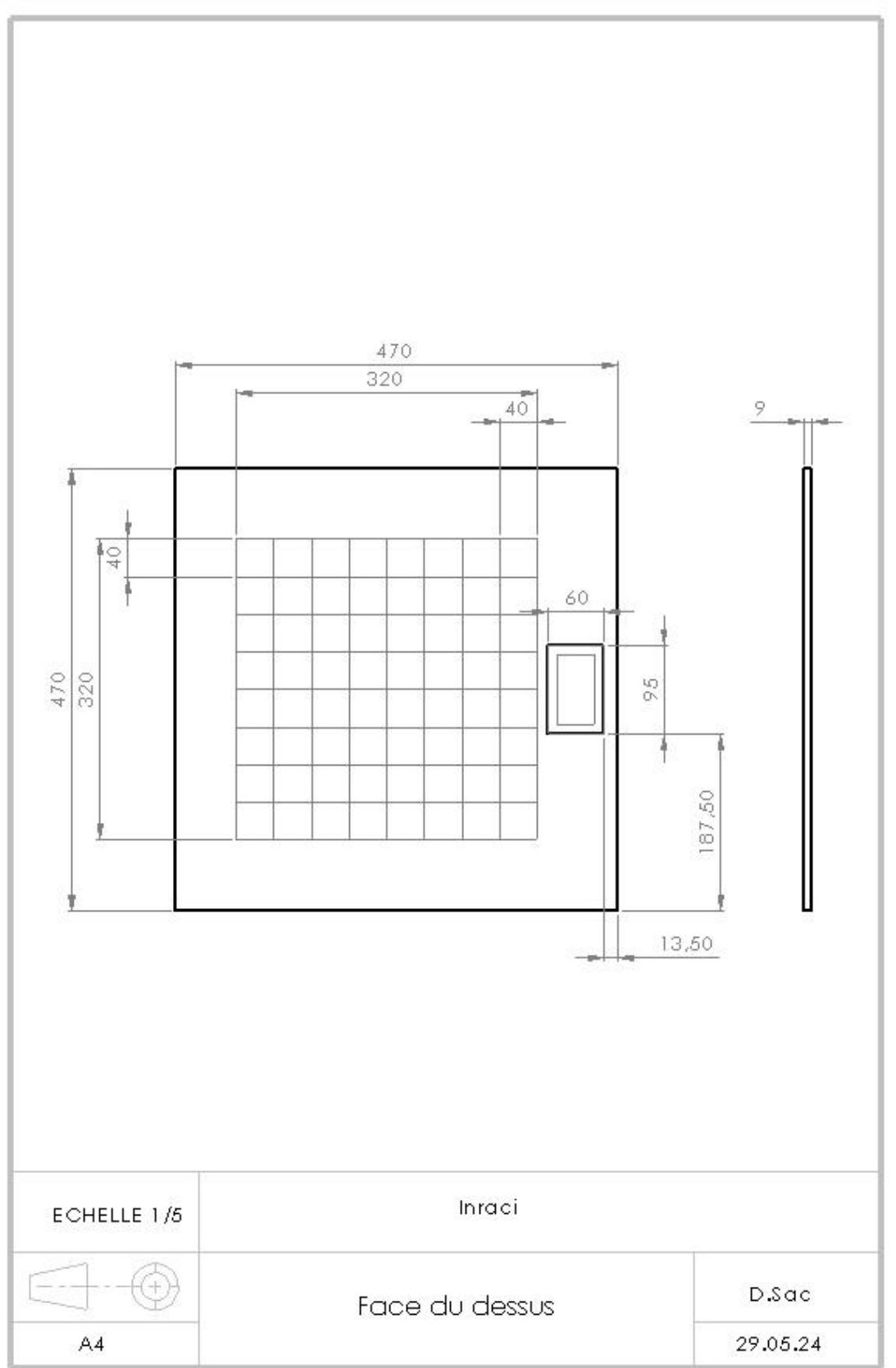
D.Sac

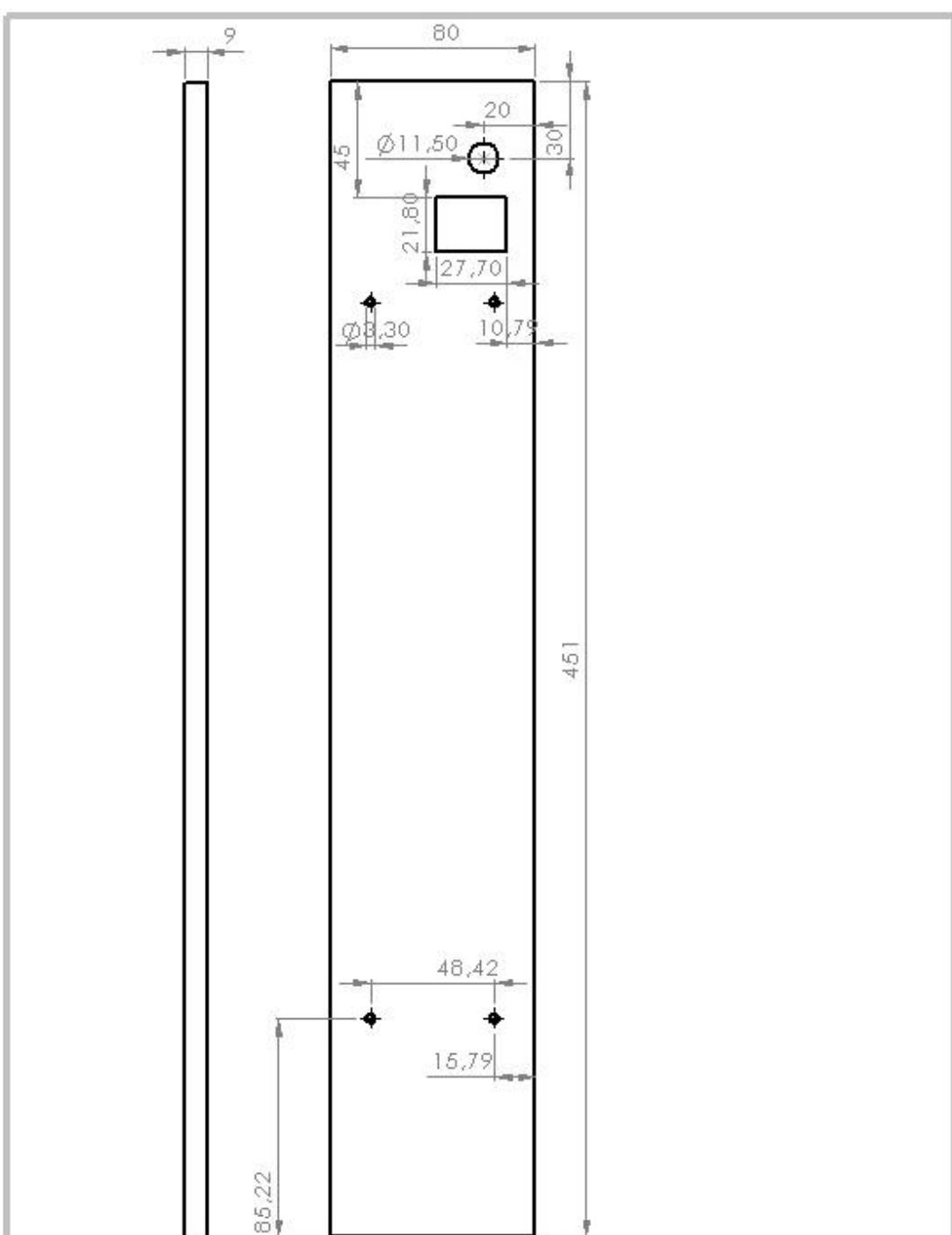
29.05.24



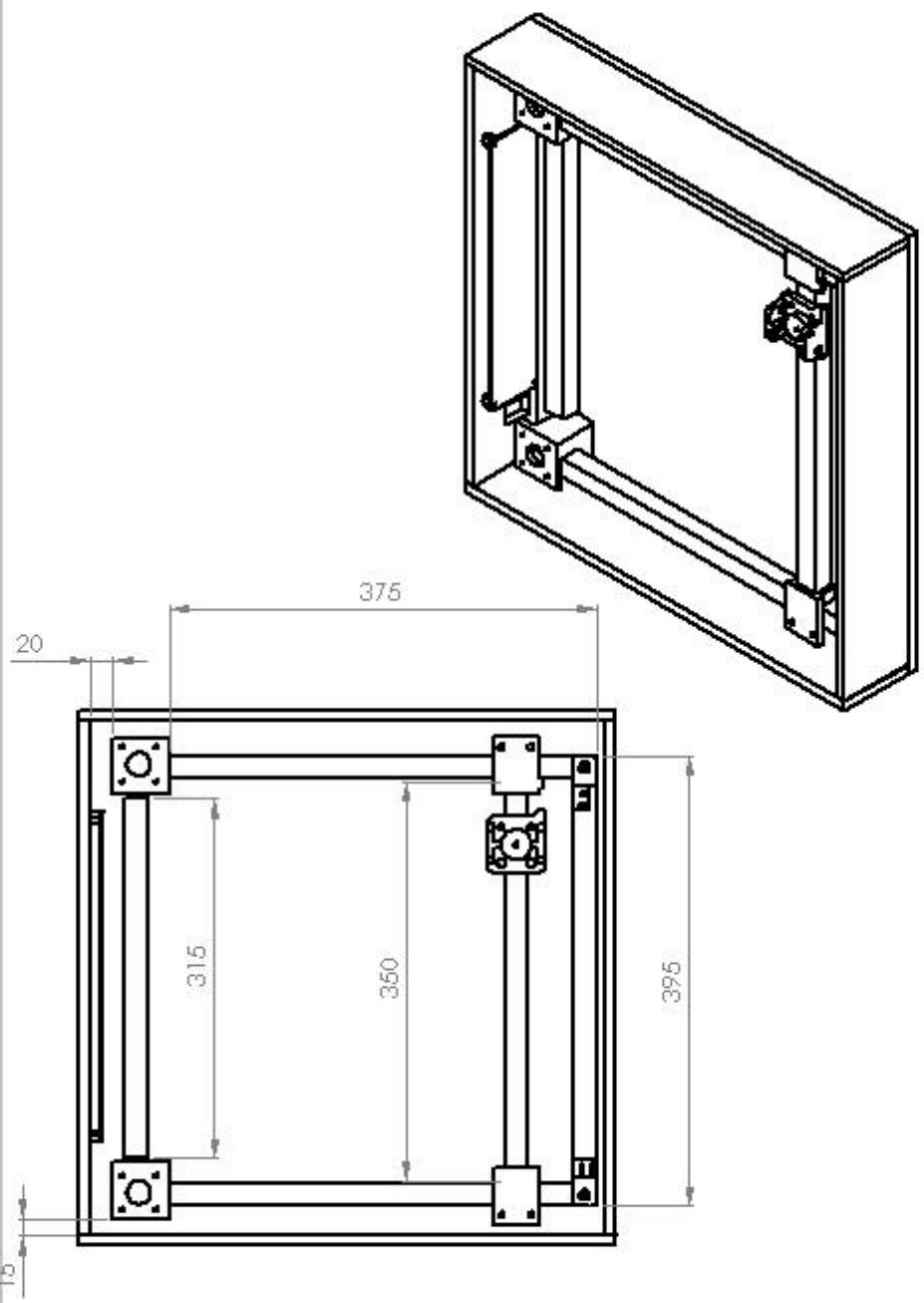
ECHELLE 1/5	Inraci	
	Vue d'ensamble	D.Sac
A4		29.05.24

VIII. Mise en plan

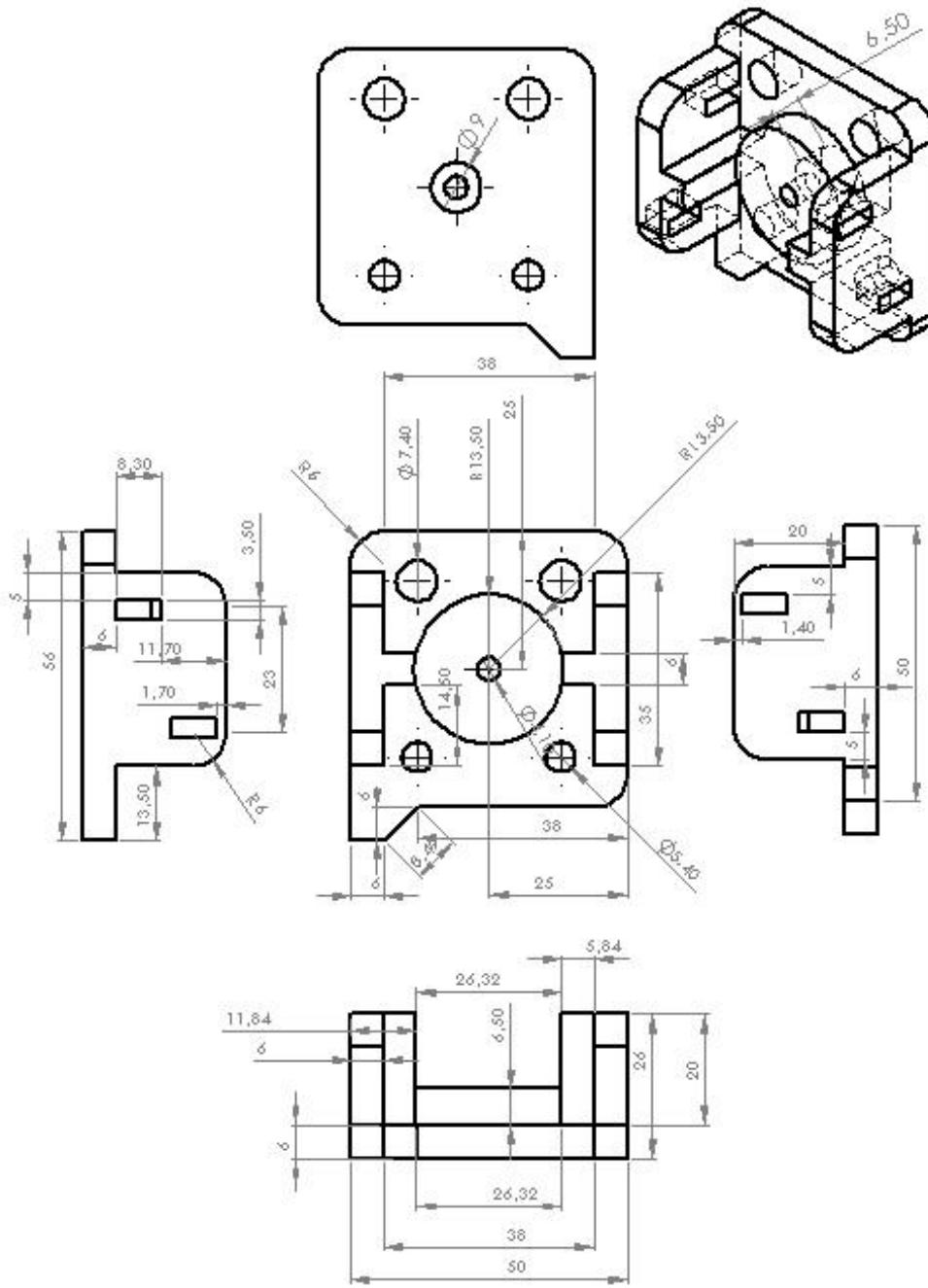




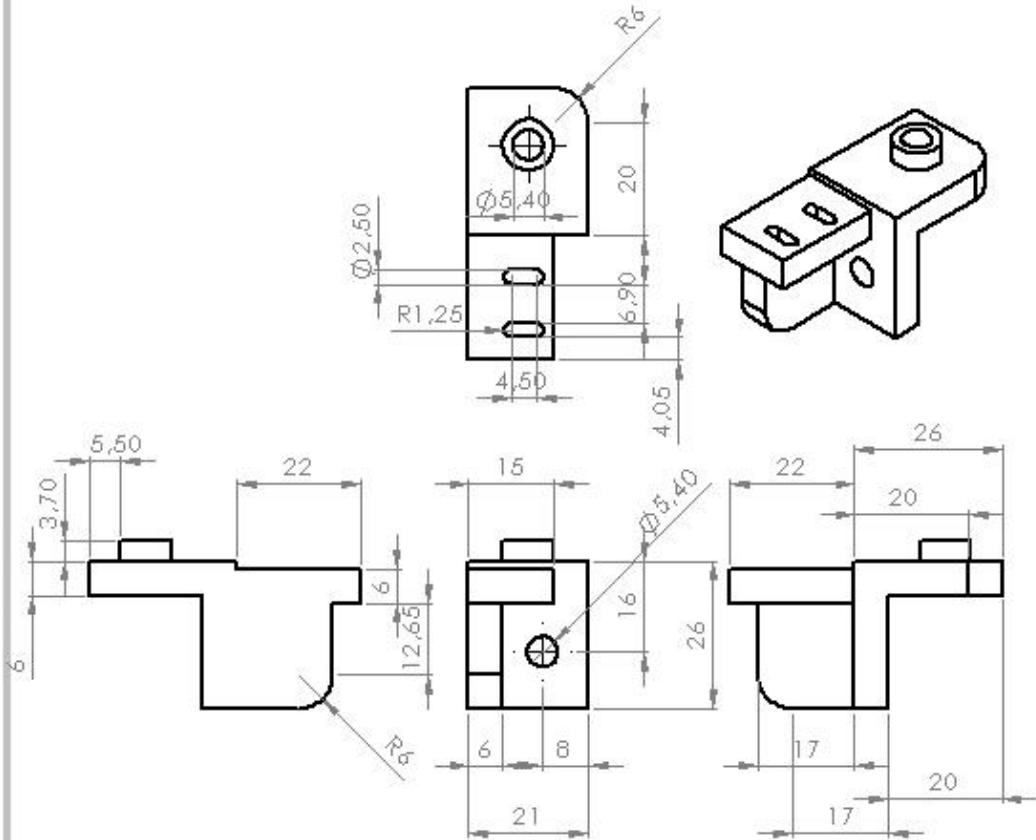
ECHELLE 1/2	Inraci	
	Face du côté de l'alimentation et du PCB	D.Sac
A4		29.05.24



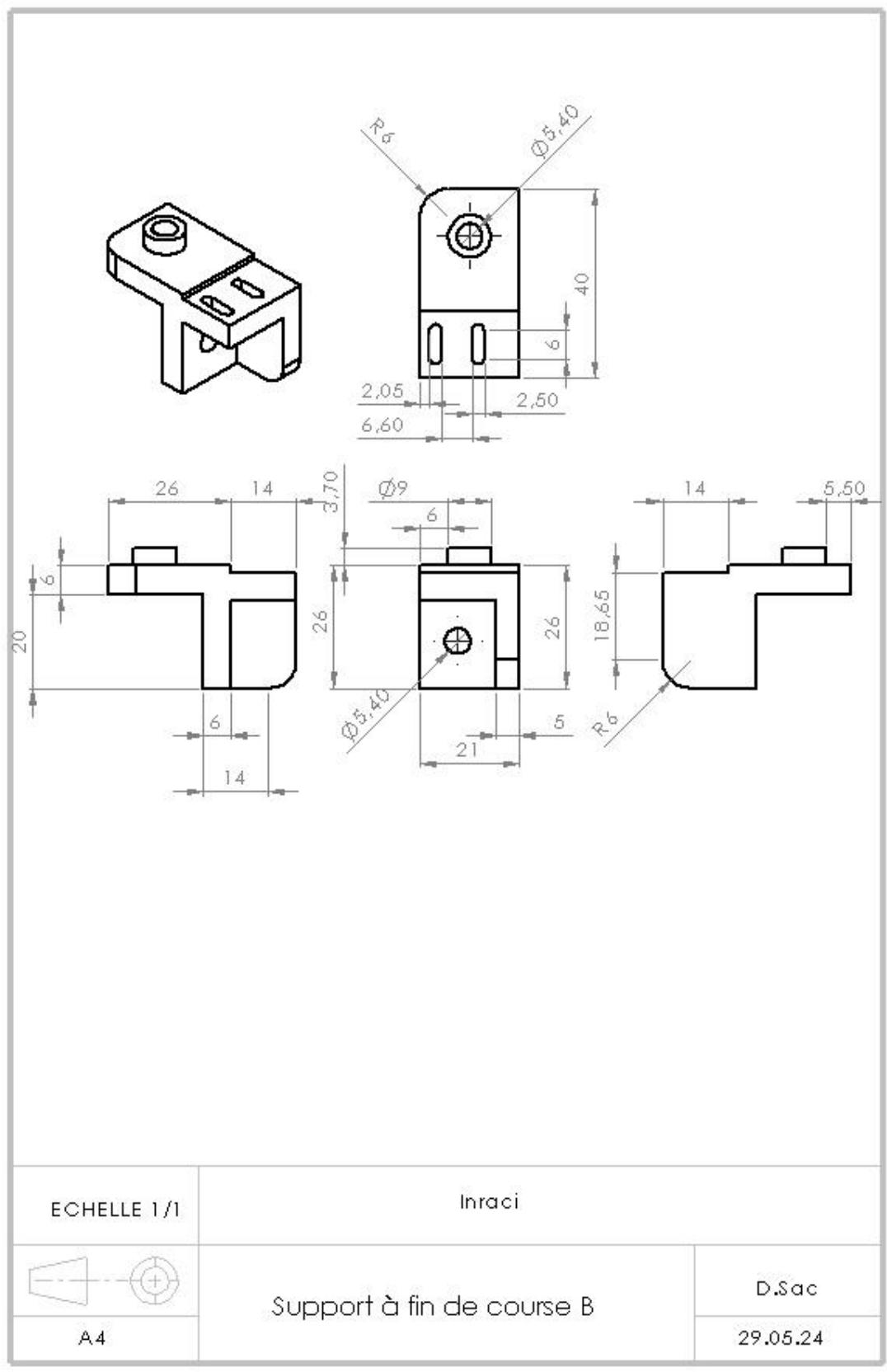
ECHELLE 1/1	Inraci	D.Sac
	Vue de l'intérieur du boîtier	29.05.24
A4		

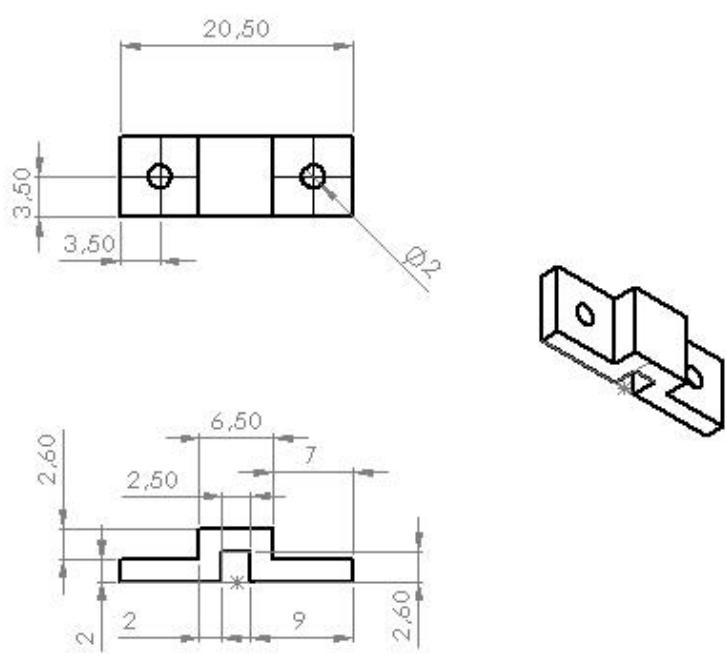


ECHELLE 1/1	Inraci	D.SAC
A4	Support à aimant	29.05.24

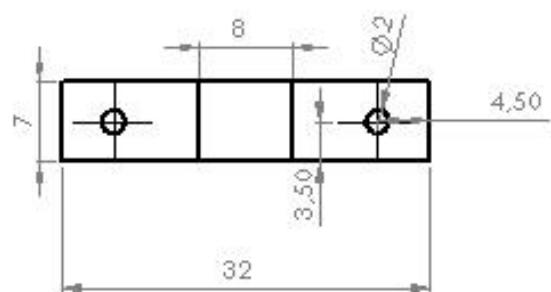
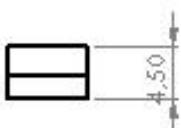
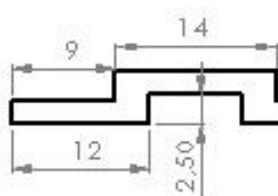
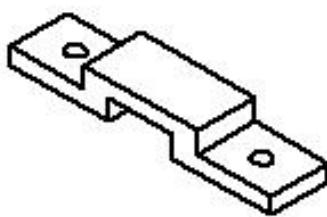


ECHELLE 1 /1	Inraci	D.Sac
	Support à fin de course A	
A4		29.05.24

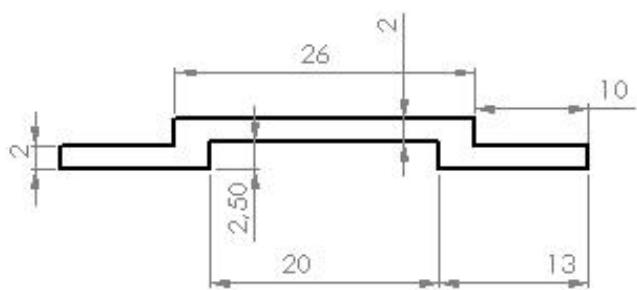
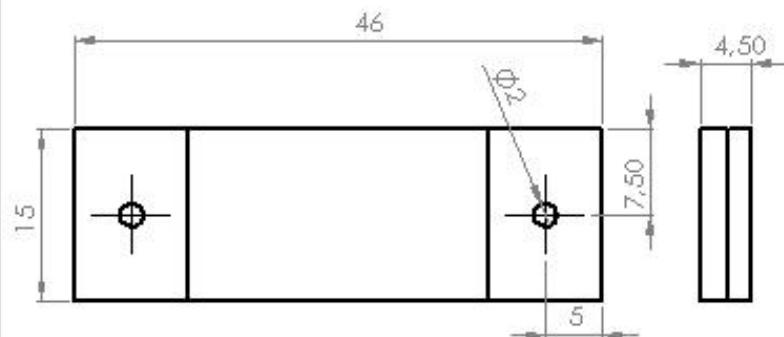
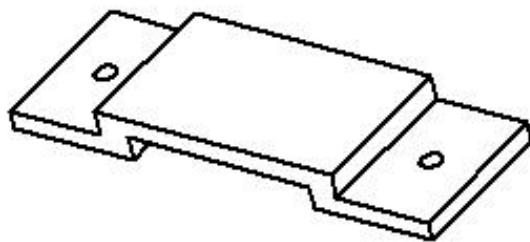




ECHELLE 1/1	Inraci	
	Support à capteur à aimant	D.Sac
A4		29.05.24



ECHELLE 1/1	Inraci	
	petit support des fils	D.Sac
A4		31.05.24



ECHELLE 1/1

Inraci



A4

Grand support pour fils

I.PALAHIUK

31.05.24

IX. Fiches techniques des composants peu courants

Le driver moteur Pololu A4988



A4988

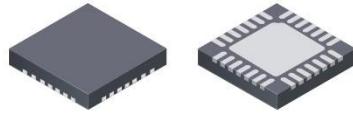
DMOS Microstepping Driver with Translator And Overcurrent Protection

Features and Benefits

- Low $R_{DS(ON)}$ outputs
- Automatic current decay mode detection/selection
- Mixed and Slow current decay modes
- Synchronous rectification for low power dissipation
- Internal UVLO
- Crossover-current protection
- 3.3 and 5 V compatible logic supply
- Thermal shutdown circuitry
- Short-to-ground protection
- Shorted load protection
- Five selectable step modes: full, $1/2$, $1/4$, $1/8$, and $1/16$

Package:

28-contact QFN
with exposed thermal pad
5 mm × 5 mm × 0.90 mm
(ET package)



Approximate size

Description

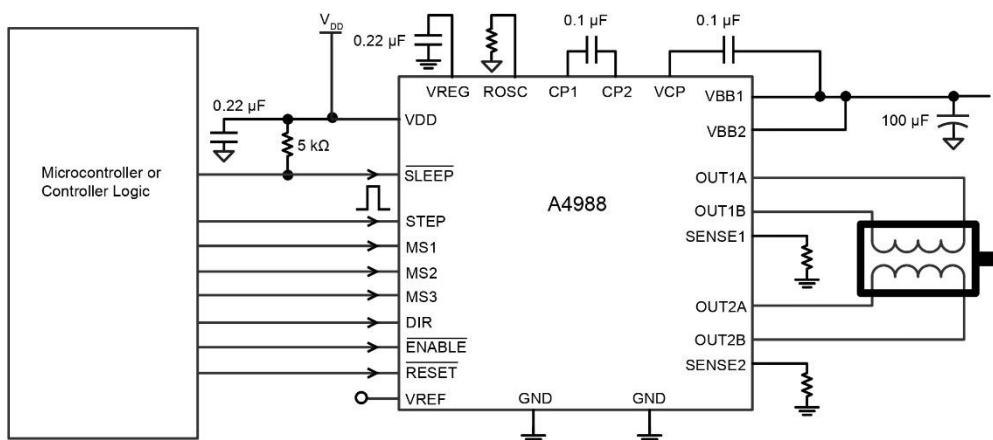
The A4988 is a complete microstepping motor driver with built-in translator for easy operation. It is designed to operate bipolar stepper motors in full-, half-, quarter-, eighth-, and sixteenth-step modes, with an output drive capacity of up to 35 V and ± 2 A. The A4988 includes a fixed off-time current regulator which has the ability to operate in Slow or Mixed decay modes.

The translator is the key to the easy implementation of the A4988. Simply inputting one pulse on the STEP input drives the motor one microstep. There are no phase sequence tables, high frequency control lines, or complex interfaces to program. The A4988 interface is an ideal fit for applications where a complex microprocessor is unavailable or is overburdened.

During stepping operation, the chopping control in the A4988 automatically selects the current decay mode, Slow or Mixed. In Mixed decay mode, the device is set initially to a fast decay for a proportion of the fixed off-time, then to a slow decay for the remainder of the off-time. Mixed decay current control results in reduced audible motor noise, increased step accuracy, and reduced power dissipation.

Continued on the next page...

Typical Application Diagram

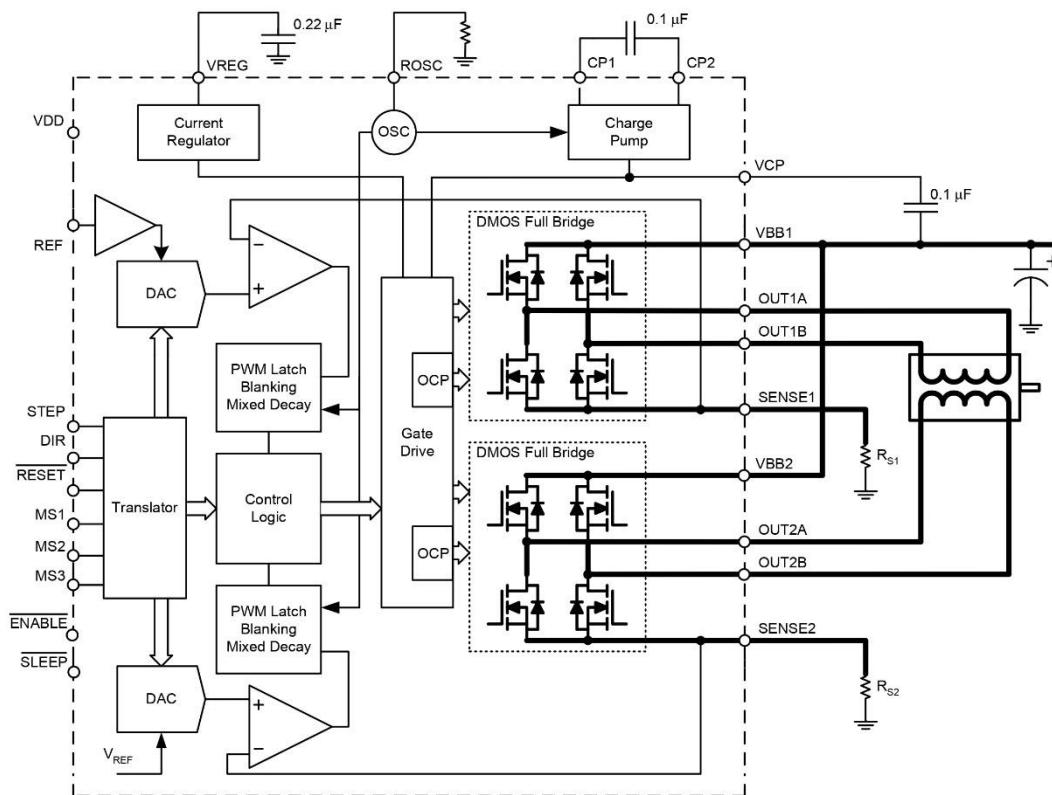


4988-DS, Rev. 5

A4988

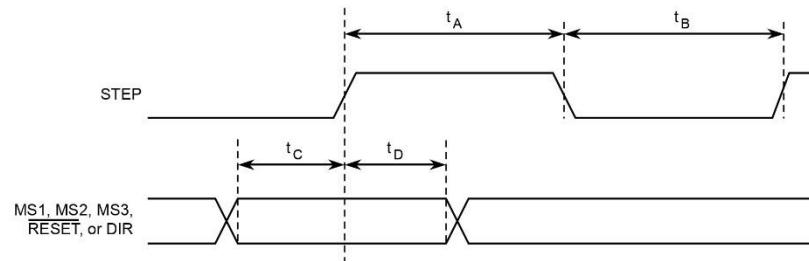
**DMOS Microstepping Driver with Translator
And Overcurrent Protection**

Functional Block Diagram



Allegro MicroSystems, LLC
115 Northeast Cutoff
Worcester, Massachusetts 01615-0036 U.S.A.
1.508.853.5000; www.allegromicro.com

3



Time Duration	Symbol	Typ.	Unit
STEP minimum, HIGH pulse width	t_A	1	μs
STEP minimum, LOW pulse width	t_B	1	μs
Setup time, input change to STEP	t_C	200	ns
Hold time, input change to STEP	t_D	200	ns

Figure 1: Logic Interface Timing Diagram

Table 1: Microstepping Resolution Truth Table

MS1	MS2	MS3	Microstep Resolution	Excitation Mode
L	L	L	Full Step	2 Phase
H	L	L	Half Step	1-2 Phase
L	H	L	Quarter Step	W1-2 Phase
H	H	L	Eighth Step	2W1-2 Phase
H	H	H	Sixteenth Step	4W1-2 Phase



Allegro MicroSystems, LLC
115 Northeast Cutoff
Worcester, Massachusetts 01615-0036 U.S.A.
1.508.853.5000; www.allegromicro.com

Functional Description

Device Operation. The A4988 is a complete microstepping motor driver with a built-in translator for easy operation with minimal control lines. It is designed to operate bipolar stepper motors in full-, half-, quarter-, eighth, and sixteenth-step modes. The currents in each of the two output full-bridges and all of the N-channel DMOS FETs are regulated with fixed off-time PWM (pulse width modulated) control circuitry. At each step, the current for each full-bridge is set by the value of its external current-sense resistor (R_{S1} and R_{S2}), a reference voltage (V_{REF}), and the output voltage of its DAC (which in turn is controlled by the output of the translator).

At power-on or reset, the translator sets the DACs and the phase current polarity to the initial Home state (shown in Figures 9 through 13), and the current regulator to Mixed Decay Mode for both phases. When a step command signal occurs on the STEP input, the translator automatically sequences the DACs to the next level and current polarity. (See Table 2 for the current-level sequence.) The microstep resolution is set by the combined effect of the MSx inputs, as shown in Table 1.

When stepping, if the new output levels of the DACs are lower than their previous output levels, then the decay mode for the active full-bridge is set to Mixed. If the new output levels of the DACs are higher than or equal to their previous levels, then the decay mode for the active full-bridge is set to Slow. This automatic current decay selection improves microstepping performance by reducing the distortion of the current waveform that results from the back EMF of the motor.

Microstep Select (MSx). The microstep resolution is set by the voltage on logic inputs MSx, as shown in Table 1. The MS1 and MS3 pins have a 100 k Ω pull-down resistance, and the MS2 pin has a 50 k Ω pull-down resistance. When changing the step mode the change does not take effect until the next STEP rising edge.

If the step mode is changed without a translator reset, and absolute position must be maintained, it is important to change the step mode at a step position that is common to both step modes in order to avoid missing steps. When the device is powered down, or reset due to TSD or an over current event the translator is set to

the home position which is by default common to all step modes.

Mixed Decay Operation. The bridge operates in Mixed decay mode, at power-on and reset, and during normal running according to the ROSC configuration and the step sequence, as shown in Figures 9 through 13. During Mixed decay, when the trip point is reached, the A4988 initially goes into a fast decay mode for 31.25% of the off-time, t_{OFF} . After that, it switches to Slow decay mode for the remainder of t_{OFF} . A timing diagram for this feature appears on the next page.

Typically, mixed decay is only necessary when the current in the winding is going from a higher value to a lower value as determined by the state of the translator. For most loads automatically-selected mixed decay is convenient because it minimizes ripple when the current is rising and prevents missed steps when the current is falling. For some applications where microstepping at very low speeds is necessary, the lack of back EMF in the winding causes the current to increase in the load quickly, resulting in missed steps. This is shown in Figure 2. By pulling the ROSC pin to ground, mixed decay is set to be active 100% of the time, for both rising and falling currents, and prevents missed steps as shown in Figure 3. If this is not an issue, it is recommended that automatically-selected mixed decay be used, because it will produce reduced ripple currents. Refer to the Fixed Off-Time section for details.

Low Current Microstepping. Intended for applications where the minimum on-time prevents the output current from regulating to the programmed current level at low current steps. To prevent this, the device can be set to operate in Mixed decay mode on both rising and falling portions of the current waveform. This feature is implemented by shorting the ROSC pin to ground. In this state, the off-time is internally set to 30 μ s.

Reset Input (RESET). The RESET input sets the translator to a predefined Home state (shown in Figures 9 through 13), and turns off all of the FET outputs. All STEP inputs are ignored until the RESET input is set to high.

Step Input (STEP). A low-to-high transition on the STEP



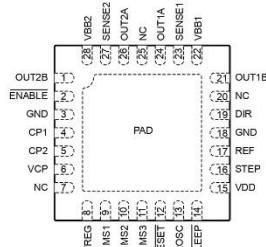
Allegro MicroSystems, LLC
115 Northeast Cutoff
Worcester, Massachusetts 01615-0036 U.S.A.
1.508.853.5000; www.allegromicro.com

7

A4988

***DMOS Microstepping Driver with Translator
And Overcurrent Protection***

Pin-out Diagram



Terminal List Table

Name	Number	Description
CP1	4	Charge pump capacitor terminal
CP2	5	Charge pump capacitor terminal
VCP	6	Reservoir capacitor terminal
VREG	8	Regulator decoupling terminal
MS1	9	Logic input
MS2	10	Logic input
MS3	11	Logic input
<u>RESET</u>	12	Logic input
ROSC	13	Timing set
SLEEP	14	Logic input
VDD	15	Logic supply
STEP	16	Logic input
REF	17	G _m reference voltage input
GND	3, 18	Ground*
DIR	19	Logic input
OUT1B	21	DMOS Full Bridge 1 Output B
VBB1	22	Load supply
SENSE1	23	Sense resistor terminal for Bridge 1
OUT1A	24	DMOS Full Bridge 1 Output A
OUT2A	26	DMOS Full Bridge 2 Output A
SENSE2	27	Sense resistor terminal for Bridge 2
VBB2	28	Load supply
OUT2B	1	DMOS Full Bridge 2 Output B
ENABLE	2	Logic input
NC	7, 20, 25	No connection
PAD	—	Exposed pad for enhanced thermal dissipation*

*The GND pins must be tied together externally by connecting to the PAD ground plane under the device.



Allegro MicroSystems, LLC
115 Northeast Cutoff
Worcester, Massachusetts 01615-0036 U.S.A.
1.508.853.5000; www.allegromicro.com

Le transistor TIP 120



PLASTIC MEDIUM-POWER

COPPLEMENTARY SILICON TRANSISTORS

...designed for general-purpose amplifier and low speed switching applications

FEATURES:

- * Collector-Emitter Sustaining Voltage -
 $V_{CEO(SUS)} = 60$ V (Min) - TIP120,TIP125
 $= 80$ V (Min) - TIP121,TIP126
 $= 100$ V (Min) - TIP122,TIP127

- * Collector-Emitter Saturation Voltage

$$V_{CE(sat)} = 2.0 \text{ V (Max.)} @ I_C = 3.0 \text{ A}$$

- * Monolithic Construction with Built-in Base-Emitter Shunt Resistor

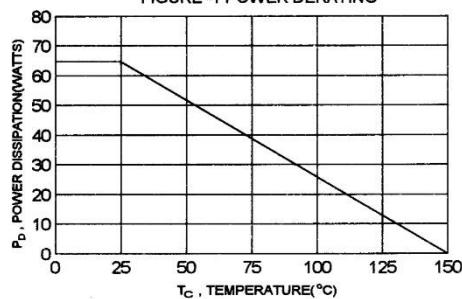
MAXIMUM RATINGS

Characteristic	Symbol	TIP120 TIP125	TIP121 TIP126	TIP122 TIP127	Unit
Collector-Emitter Voltage	V_{CEO}	60	80	100	V
Collector-Base Voltage	V_{CBO}	60	80	100	V
Emitter-Base Voltage	V_{EBO}		5.0		V
Collector Current-Continuous -Peak	I_C I_{CM}		5.0		A
Base Current	I_B		120		mA
Total Power Dissipation @ $T_C = 25^\circ\text{C}$ Derate above 25°C	P_D		65 0.52		W W/ $^\circ\text{C}$
Operating and Storage Junction Temperature Range	T_J, T_{STG}		- 65 to +150		$^\circ\text{C}$

THERMAL CHARACTERISTICS

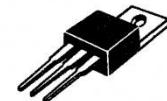
Characteristic	Symbol	Max	Unit
Thermal Resistance Junction to Case	R_{eJC}	1.92	$^\circ\text{C/W}$

FIGURE -1 POWER DERATING

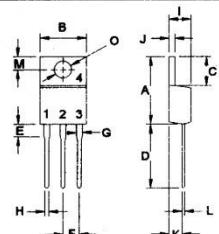


NPN	PNP
TIP120	TIP125
TIP121	TIP126
TIP122	TIP127

5.0 AMPERE
DARLINGTON
COMPLEMENTARY SILICON
POWER TRANSISTORS
60-100 VOLTS
65 WATTS



TO-220



PIN 1.BASE
2.COLLECTOR
3.EMITTER
4.COLLECTOR(CASE)

DIM	MILLIMETERS	
	MIN	MAX
A	14.68	15.31
B	9.78	10.42
C	5.01	6.52
D	13.06	14.62
E	3.57	4.07
F	2.42	3.66
G	1.12	1.36
H	0.72	0.96
I	4.22	4.98
J	1.14	1.38
K	2.20	2.97
L	0.33	0.55
M	2.48	2.98
O	3.70	3.90

TIP120, TIP121, TIP122 NPN / TIP125, TIP126, TIP127 PNP

ELECTRICAL CHARACTERISTICS ($T_c = 25^\circ\text{C}$ unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
----------------	--------	-----	-----	------

OFF CHARACTERISTICS

Collector - Emitter Sustaining Voltage (1) ($I_c = 30 \text{ mA}, I_B = 0$)	TIP120,TIP125 TIP121,TIP126 TIP122,TIP127	$V_{CEO(sus)}$	60 80 100	V
Collector Cutoff Current ($V_{CE} = 30 \text{ V}, I_E = 0$) ($V_{CE} = 40 \text{ V}, I_E = 0$) ($V_{CE} = 50 \text{ V}, I_E = 0$)	TIP120,TIP125 TIP121,TIP126 TIP122,TIP127	I_{CEO}		mA
Collector Cutoff Current ($V_{CB} = 60 \text{ V}, I_E = 0$) ($V_{CB} = 80 \text{ V}, I_E = 0$) ($V_{CB} = 100 \text{ V}, I_E = 0$)	TIP120,TIP125 TIP121,TIP126 TIP122,TIP127	I_{CBO}		mA
Emitter Cutoff Current ($V_{EB} = 5.0 \text{ V}, I_C = 0$)		I_{EBO}	0.2 0.2 0.2	mA

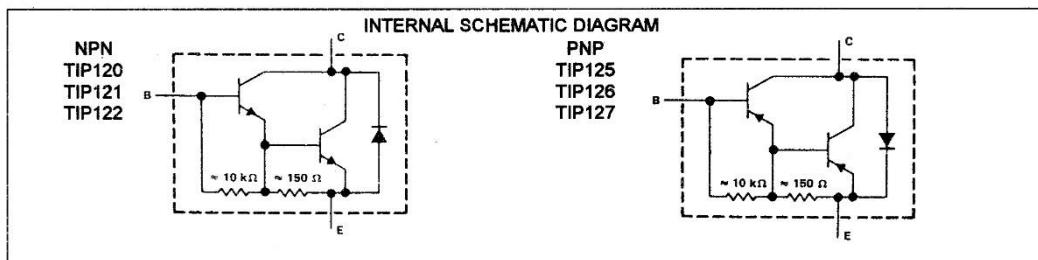
ON CHARACTERISTICS (1)

DC Current Gain ($I_C = 0.5 \text{ A}, V_{CE} = 3.0 \text{ V}$) ($I_C = 3.0 \text{ A}, V_{CE} = 3.0 \text{ V}$)	h_{FE}	1000 1000		
Collector-Emitter Saturation Voltage ($I_C = 3.0 \text{ A}, I_B = 12 \text{ mA}$) ($I_C = 5.0 \text{ A}, I_B = 20 \text{ mA}$)	$V_{CE(sat)}$		2.0 4.0	V
Base-Emitter On Voltage ($I_C = 3.0 \text{ A}, V_{CE} = 3.0 \text{ V}$)	$V_{BE(on)}$		2.5	V

DYNAMIC CHARACTERISTICS

Small-Signal Current Gain ($I_C = 3.0 \text{ A}, V_{CE} = 4.0 \text{ V}, f = 1.0 \text{ MHz}$)	h_{fe}	4.0		
Output Capacitance ($V_{CB} = 10 \text{ V}, I_E = 0, f = 0.1 \text{ MHz}$) TIP120,TIP121,TIP122 TIP125,TIP126,TIP127	C_{ob}		300 250	pF

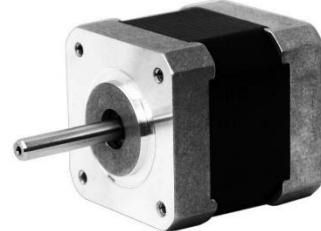
(1) Pulse Test: Pulse width = 300 us , Duty Cycle $\leq 2.0\%$



Le moteur pas à pas

1.8° 42MM High Torque Hybrid Stepping Motor

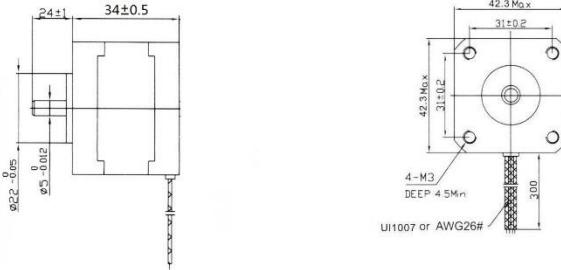
Item	Specifications
Step Angle	1.8°
Step Angle Accuracy	±5% (full step, no load)
Resistance Accuracy	±10%
Inductance Accuracy	±20%
Temperatru Rise	80°CMax. (rated current, 2 phase on)
Ambient Temperatuar	-20°C~+50°C
Insulation Resistance	100MΩMin., 500VDC
Dielectric Strength	500VAC/ for one minute
Shaft Radial Play	0.02Max. (450 g-load)
Shaft Axial Play	0.08Max. (450 g-load)
Max. radial force	28N (20mm from the flange)
Max. axial force	10N



● 42MM Hybrid Stepping Motor Specifications

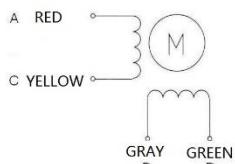
Model No	Rated Voltage	Current /Phase	Resistance /Phase	Inductance /Phase	Holding Torque	# of Leads	Rotor Inertia	Weinght	Detent Torque	Length
XY42STH34-0354A	V	A	Ω	mH	Kg-cm		g-cm²	kg	g-cm	mm
XY42STH34-0354A	12	0.35	34	33	1.6	4	35	0.22	120	34

● Dimension

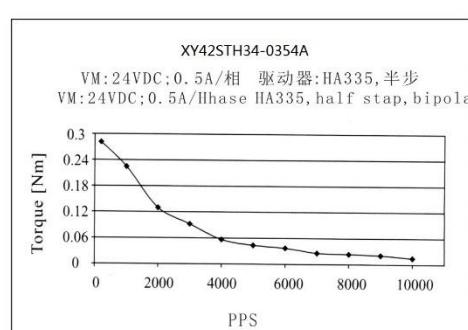


● Wiring Diagram

4 LEADS



● Pull out Torque Curve



X. Sources du projet

Fiches techniques des composants

<https://docs.arduino.cc/resources/datasheets/A000005-datasheet.pdf>

https://www.pololu.com/file/download/a4988_DMOS_microstepping_driver_with_translator.pdf?file_id=0J450

<https://pages.pbclinear.com/rs/909-BFY-775/images/Data-Sheet-Stepper-Motor-Support.pdf>

https://www.ti.com/lit/ds/symlink/cd74hc4067.pdf?ts=1717223357397&ref_url=https%253A%252F%252Fwww.google.com%252F

https://www.mouser.com/datasheet/2/187/honeywell_hwscs04648-1-1735504.pdf

https://nextion.ca/datasheets/nx3224t024_011/

https://www.alldatasheet.com/view.jsp?Searchword=Datasheet%20tip120&gad_source=1&gclid=CjwKCAjwjeuyBhBuEiwAJ3vuoeOOkJIWU-3P_XCdjEkRbp5iweYyHIQZ3TE4ZteqdSRLWCSbX3d6hBoCxE8QAvD_BwE

Source principale du projet

<https://www.instructables.com/Automated-Chessboard/>

