

## Recommandations pour réaliser les fiches.

Par Nulentout (Achevé le 28 Juin 2020.)

**M**ise à part cette première page d'explications, toutes les autres *sont prévues pour réaliser des fiches en imprimant les "pages" de ce document par "paire" Recto / Verso* sur du papier de qualité suffisante pour que les éléments situés d'un coté ne soient pas visibles de l'autre. Pour ne pas vous tromper, le tableau de la Fig.1 précise les paires constituant une même "fiche double" car chaque page au format A4 est relatif à deux fiches indépendantes.

**R**éaliser une "paire de fiche" n'est pas spécialement compliqué, toutefois le fait d'avoir à imprimer des deux cotés d'une feuille impose une procédure simple, mais rigoureuse. À titre d'exemple on va traiter le cas des fiches couplées de la FEUILLE ③ :

- 1) Imprimer la **page 6**. (*Repère vertical en gris clair au centre.*)
- 2) Replacer la page sur votre périphérique et imprimer la **page 7**.  
Logiquement, si c'est une imprimante classique, il suffit en principe de replacer la feuille sur le dessus du bac à papier dans la position et l'orientation qu'elle occupe en sortie de la machine.
- 3) Étape non obligatoire, personnellement je protège toutes mes fiches, disposant d'une petite plastifieuse thermique pour P.C.
- 4) Il ne reste plus qu'à séparer les deux fiches en coupant la feuille par le milieu. Puis on découpe tout le tour de la fiche le cadre gris clair "en laissant vivre le trait", cadre qui en délimite la périphérie. Il n'y a pas d'ordre logique dans la succession des fiches car elles ont été rédigées au cours des semaines durant le développement de ce petit projet. **Naturellement, il n'est absolument pas obligatoire d'imprimer toutes les fiches. Vous avez parfaitement le loisir de ne concrétiser que celles qui vous semblent indispensables ...**

**NOTE :** Parfois la façon dont sont organisées les diverses fiches d'un même thème peut sembler curieuse. Elles sont paginées pour permettre au lecteur d'avoir simultanément deux faces relatives à des informations communes. Instructions du gcode et le Pilotage manuel par la ligne USB en sont un exemple typique ...

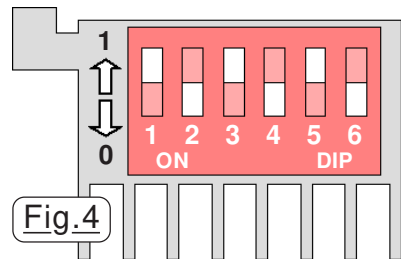
## "Paires de pages" pour réaliser les fiches Recto / verso.

RECTO	VERSO	RECTO	VERSO	RECTO	VERSO
2	3	4	5	6	7
8	9	10	11	12	13
14	15	16	17	18	19
20	21	22	23	24	25
26	27	28	29	30	31

- (2) Pilote de moteurs TB6600. & Moteurs pas à pas 17HS4401.
- (3) Pilote de moteurs TB6600. & LASER de puissance 5500mW.
- (4) Bibliothèque PetitFS.h et PetitSerial.h. & C.I du petit clavier.
- (5) Bibliothèque PetitSerial.h.
- (6) Utiliser LASER GRBL v3.0.4.
- (7) Utiliser LASER GRBL v3.0.4 & Contraintes pour les fichiers gco.
- (8) Carte Arduino NANO. & Intervention sous le plateau ...
- (9) Protocole d'intégration du PUPITRE.
- (10) Instructions gcode étendues.
- (10) Pilotage manuel par la ligne USB du Moniteur.
- (11) Résumé des contraintes pour les fichiers gco.
- (12) Menu PYROGRAVURE. & Menu FICHIERS.
- (13) Le menu des OPTIONS. & Le menu SYSTEME.
- (13) Protocole standard pour effectuer une gravure.
- (14) Le mode MANUEL. & Informations et conseils divers.
- (15) Rangement / Mise en service de la machine.
- (15) Option Mode LASER continu. & Le menu de base.
- (16) Protocole d'alignement du guidage X'X.
- (17) Protocole d'alignement du guidage Y'Y.
- (17) Protocole de mise en place de la courroie crantée.
- (18) à (21) Protocoles de validation du matériel. 1/8 à 8/8
- (22) Câblage des capteurs d'Origine Machine.
- (22) Branchements de la carte PCA9685.
- (22) Utilisation du multiplexeur de servomoteurs.
- (24) et (25) Fiche de câblage des circuits électriques. 1/4 à 4/4
- (26) Ignorer la saisie des Origines Machine :
- (26) Position LASER sur incident imposant un RESET :
- (26) Fiche de câblage des circuits électriques. 5/5
- (27) Le Relais d'alimentation du LASER.
- (27) Dessin du circuit imprimé pour le Relais du LASER.
- (28) BUS série au standard I2C. (1/2) & Encodeur rotatif KY-040.
- (29) Petit module LASER. & BUS série au standard I2C. (2/2)
- (30) TESTEUR DE CONTINUITÉ ET DE LED.
- (30) Vérification du circuit imprimé principal. 2/2
- (31) Vérification du circuit imprimé principal. 1/2

## Pilote de moteurs pas à pas TB6600. 3/3

L'intensité qui traverse les enroulements est directement fonction du couple résistant présent sur l'arbre de sortie. Cette dernière est fonction du courant qui sera géré par le module électronique TB6600. Le tableau de la Fig.4 présente quelques options possibles. Le couple limite correspond au plus grand couple

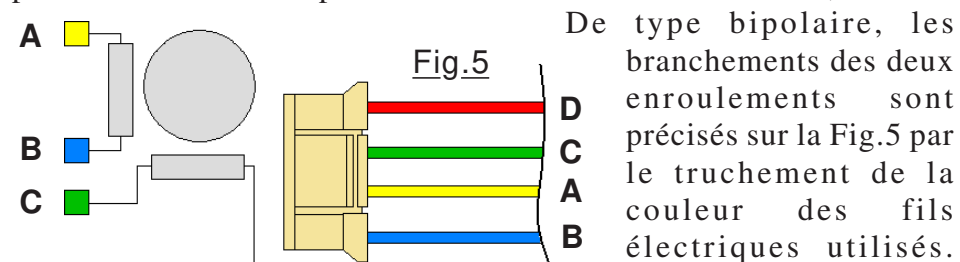


4	5	6	I à vide	Couple limite
0	0	0	120mA	170mA
0	1	0	470mA	640mA
0	0	1	940mA	≈1,3A

possible avant que le moteur "ne cale". L'option retenue correspond à la puissance nominale pour effectuer des déplacements fiables tout en respectant les limitations de courant propres aux moteurs.

## Moteurs pas à pas 17HS4401.

Conçus pour 200 pas par tour en mode biphasé, l'incrémentation fait 1,8° par mouvement élémentaire. Le courant nominal est annoncé pour 1,7A avec une résistance interne des enroulements de 1,5Ω et une inductance de 2,8mH. Le couple de maintien est annoncé pour 40cm^N. Le couple résistant en mode veille est de 2,2cm^N.



De type bipolaire, les branchements des deux enroulements sont précisés sur la Fig.5 par le truchement de la couleur des fils électriques utilisés. Pour adopter le sens de rotation désiré sur la machine il suffira de sélectionner par programme l'état logique de la ligne pilotée par D6 pour le déplacement sur X'X et par D8 pour le mouvement sur Y'Y.

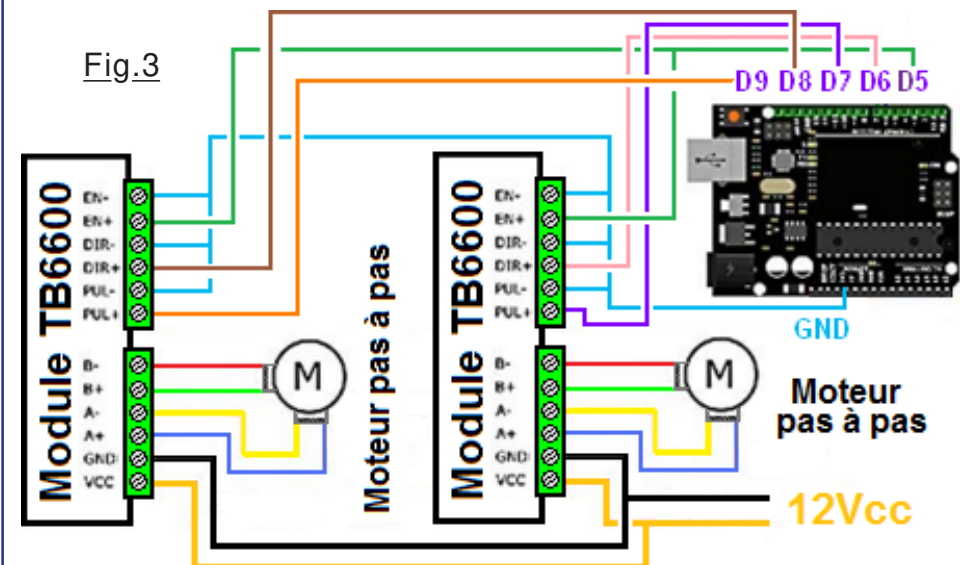
Ces moteurs peuvent être approvisionné à l'unité sur :

<https://www.amazon.fr/dp/B07255B43M/>  
ref=pe\_3044141\_189395771\_TE\_dp\_3

## Pilote de moteurs pas à pas TB6600. 2/3

Montré ci-dessous, le schéma retenu peut être expérimenté avec P02\_Test\_des\_moteurs.ino qui permet de tester le module de commande. L'ensemble est câblé en vue de piloter avec des signaux à logique positive. La sortie de mise en veille est commune aux deux moteurs et branchée sur D5. D6 pilote le sens de rotation sur X'X et D7 les impulsions de génération des pas sur le moteur. De façon analogue D8 pilote le sens de rotation sur Y'Y et D9 les impulsions de validation des pas sur le moteur.

### Branchements entre les deux pilotes et Arduino NANO.



Suite à l'expérience sur la machine complète, une gestion du courant à 1A (1,2A MAXI sur transitoires.) qui fournit un couple largement suffisant tout en provoquant un échauffement dérisoire des moteurs. Les courants mesurés sur le servomoteur sont précisés sur la fiche n°3/3. Le module de pilotage TB6600 ne chauffe absolument pas, il est parfaitement inutile de prévoir une ventilation forcée sur la machine. La poulie présentant un diamètre moyen d'environ 12,73mm, on obtient une définition de 0,025mm par pas à 1600 pas par tour. ( $12,73 * 3,14 / 1600 = 0,025mm$ ) Le tableau situé en bas de la fiche n°1/3 précise dans la ligne mise en évidence par du jaune la combinaison à adopter sur les inverseurs 1, 2 et 3.

## Pilote de moteurs pas à pas TB6600. 1/3

Équipé d'un dissipateur thermique de grande surface ce module électronique est adapté au pilotage de moteurs puissants puisqu'il peut commuter jusqu'à 4A par enroulement. Il intègre un dispositif de huit limitations de courant comprises entre 0,5A à 4A.

Présentée sur la Fig.1 l'interface intègre une isolation par optocoupleurs à commutation rapide avec une forte immunité aux interférences hautes fréquences.



Fig.1

Gère sept différent modes de division pour adapter l'angle de rotation par pas. (Micro Step : 1, 2A, 2B, 4, 8, 16, 32. La résolution pour les demi pas 2A et 2B est la même, mais A représente le bobinage A et B l'enroulement B.) Un dispositif protège contre une inversion de tension en entrée de puissance dont le courant d'entrée peut aller jusqu'à des crêtes de 5A et une tension d'alimentation des moteurs comprise entre 3,3v et 24v. Intègre une protection contre les surchauffes, surintensités et les courts-circuits, la puissance maximale commutée étant de 160W. Le voyant vert indique que le système est sous tension et le voyant bleu indique que le signal est présent et que le pilote fonctionne.

Le fournisseur indique que le voyant vert indique que le système est sous tension et le voyant bleu indique que le signal est présent et que le pilote fonctionne. La fréquence d'horloge peut aller jusqu'à 200kHz avec des durées d'impulsions minimales de 2,2μS. La Fig.2 sert de référence pour préciser le positionnement des inverseurs qui servent à définir l'angle de rotation unitaire. (Attention : ON et OFF semble incorrect sur la sérigraphie.)

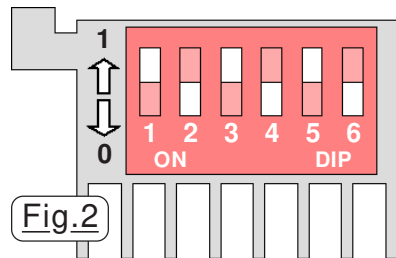


Fig.2

1	2	3	Rotation	Nb pas/tour
0	0	0	NC	Arrêt
0	0	1	x1	200
0	1	0	x1/2	400 2A
1	0	0	x1/2	400 2B
1	0	1	x1/8	1600

## LASER de puissance 5500mW.

Équipé d'un dissipateur thermique ventilé de grande surface ce module est piloté par une électronique en **PWM niveaux logiques TTL à une fréquence de 1000Hz**. La longueur d'onde du faisceau est de 450nm. (Bleu.) La source d'énergie de puissance doit être de 12Vcc. LASER et électronique de pilotage sont refroidis par des ventilateurs dont le souffle est dosé en fonction du dégagement thermique du LASER. Le module est équipé d'une lentille optique à faible divergence en verre protégée par revêtement. Le circuit électronique de commande fonctionne avec rétroaction pour gérer un courant constant. La puissance de sortie maximale est de 5.5W pour une puissance consommée d'environ 26W sur le 12V. (2,2A à pleine puissance.) **L'exposition aux yeux est très dangereuse, imposant le port de lunettes spécifiques de protection.**

**Ne pas utiliser des cibles qui retournent le faisceau vers le haut, pouvant endommager par absorption le LASER.**

L'énergie de puissance 12v peut être fournie sur la petite prise coaxiale repérée 3 sur la Fig.1 ou par le connecteur rouge 1 sachant que la référence GND est commune. L'inverseur 2 permet de couper l'énergie 12v. Quand le LASER fonctionne, la LED rouge du module de pilotage indique que l'ensemble est alimenté en énergie. La LED bleue est un "répétiteur" du signal PWM, c'est à dire que sa luminosité est fonction du rapport cyclique du signal TTL. Montré sur la Fig.2, le branchement sur le connecteur

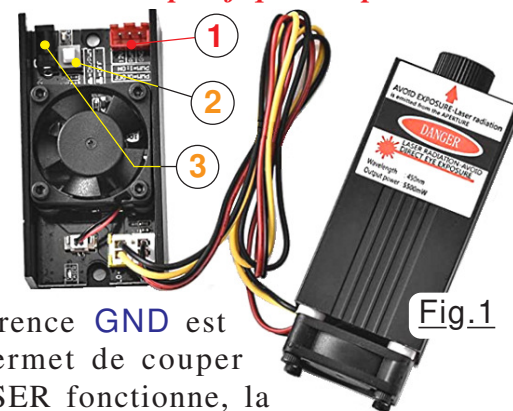


Fig.1

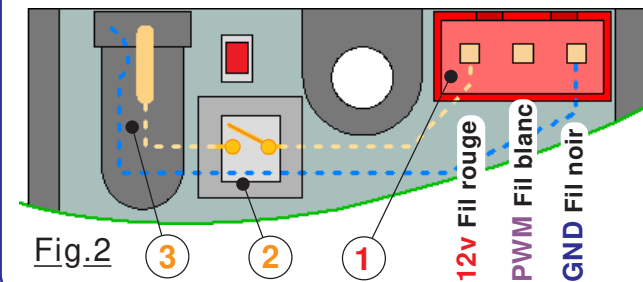


Fig.2

rouge 1 est utilisé sur la machine pour diminuer le nombre de fils électriques dans la ligne qui doit acheminer les signaux et l'énergie sur le chariot X'X.



## Bibliothèque **PetitFS.h** et **PetitSerial.h**.

Incluses dans **PetitFS-Master.h** elles ont pour but de minimiser le code objet quand on utilise *de façon restreinte* un lecteur de carte SD ou la ligne série USB du Moniteur de l'IDE. Il faut savoir que **SD.h** est très performante, mais elle s'octroie environ 50% de l'espace programme d'une carte Arduino, sans compter la taille exigée en mémoire dynamique. Utiliser **PetitSerial.h** fait économiser 936 octets en mémoire de programme et 161 octets en mémoire dynamique si on se contente de lister sur la ligne USB. Seulement 44 octets sont consommés en zone de travail et sur la PILE avec un code de faible taille. (2K-4K octets) Reconnaît les FAT12, FAT16 et FAT32. Travaille sur un seul fichier à la fois.

### Déclarations pour **PetitSerial.h**

```
#include "PetitSerial.h"
PetitSerial PS; // PetitSerial imposé au lieu de Serial standard.
#define Serial PS
Et dans void setup() le classique Serial.begin(115200);
Ne sont utiles que si la ligne USB série du Moniteur est utilisée.
```

### Déclarations pour **PetitFS.h**

```
#include "PetitFS.h"
FATFS fs; // Déclaration des fichiers système.
```

### Protocoles de **PetitFS.h**

**pf\_mount(&fs);**

Fonction qui retourne un code d'erreur non nul si le lecteur n'est pas opérationnel. Exemple d'utilisation :

```
Lecteur_OK = true; if (pf_mount(&fs)) Lecteur_OK = false;
```

**pf\_open(Nom\_Fichier);**

Fonction qui retourne code d'erreur si **Nom\_Fichier** n'est pas présent sur la carte SD insérée dans le lecteur. Exemple d'utilisation :

```
if (pf_open(Nom_Fichier)) "Générer une erreur"; else "utiliser".
```

**ATTENTION** : Le nom du fichier doit respecter les limitations de l'ancien DOS et *ne contenir que des lettres, des chiffres, et le point séparant le type*. Le nom *ne devra pas dépasser huit caractères*. Enfin, **pf\_open** ne distingue pas majuscules et minuscules. Même si le fichier testé contient des lettres minuscules, dans la désignation **Nom\_Fichier** *il ne faudra utiliser que des lettres majuscules*.

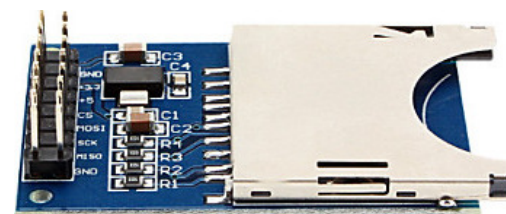
... / ...

## Utilisation du lecteur de carte SD.

Bien que les entrées soient connectées au +3,3v généré en interne sur le petit module, comme des résistances de 10kΩ sont intercalées, on peut directement les brancher sur les sorties d'Arduino dont les tensions peuvent monter jusqu'à +5Vcc. Il n'y a pas de risque pour le matériel, car par nature sur les lignes de type I2C on dialogue par le truchement de "transistors à collecteurs ouverts". La nature des broches étant précisée sur le module, on se contentera de brancher le +5Vcc, CS, MOSI, SCK, MISO conformément aux indications du programme **P04\_TEST\_du\_lecteur\_carte\_SDRAM.ino**.

**ATTENTION** : Parfois la carte étant déjà en place, elle n'est pas reconnue dans certains cas. Pour parer ce problème :

- 1) **GND** de la carte va au repos du B.P. du RESET extérieur.
- 2) Si la carte n'est pas reconnue il suffit de refaire un RESET.



Communiquer avec une carte SD et manipuler un système de fichier n'est pas une opération triviale et englutit beaucoup de ressources, surtout en mémoire de programme et en

RAM dynamique. La bibliothèque par défaut d'Arduino **SD.h** est très performante, mais consomme environ 50% de l'espace programme possible. Aussi, dans la mesure où seule la lecture de fichiers est envisagée pour notre application, on utilisera la bibliothèque nommée **PetitFS-Master.h** qui inclut **PetitFS.h** et **PetitSerial.h**. (Voir la fiche dédiée à cette bibliothèque.) Dans notre application la gestion du port série du Moniteur de l'IDE n'étant pas exploité, la bibliothèque **PetitSerial.h** n'est pas invoquée.

Concrètement, la bibliothèque **PetitFS.h** ne gère que les noms de fichiers ou dossiers au format DOS 8.3, car les cartes sont au format FAT issu du système d'exploitation de Microsoft. Il faudra de ce fait *limiter les noms de fichiers* ou de dossiers *à huit caractères*. Les noms de fichiers sont insensibles à la case, *minuscules et majuscules ne différencient donc pas les fichiers*.

Petit-Master se télécharge sur <https://github.com/greiman/PetitFS>

## Bibliothèque *PetitSerial.h*.

Substituer la bibliothèque native dans l'IDE par *PetitSerial.h* diminue de 161 octets l'occupation dans la mémoire dynamique et économise environ 970 octets dans l'espace réservé au programme.

### ➤ Déclaration de *PetitSerial.h* dans le programme.

Seule la bibliothèque *PetitSerial.h* sera à indiquer dans les directives si un lecteur de carte SD n'est pas présent, *PetitFS.h* n'étant alors pas utilisé. Trois lignes de programme suffisent :

```
#include "PetitSerial.h"
```

```
PetitSerial PS; // Utiliser PetitSerial au lieu du "Serial standard".
```

```
#define Serial PS
```

### ➤ Les instructions de base pour le programme.

Identiques globalement à celles de la bibliothèque native, l'écriture des instructions ne change pas pour lire, écrire ou définir la vitesse de transmission, les effets étant identiques.

*Serial.begin(Vitesse)* : Précise la vitesse exprimée en bauds.

**ATTENTION : Ne pas dépasser la vitesse de 57600 bauds.**

*Serial.print(Donnée)* : Affiche la valeur de la donnée sans passer à la ligne. Les données sont présentées avec les formats habituels tels que DEC, BIN, HEX, OCT, *char(nn)*, *restriction du nombre de chiffres précisés après la virgule pour les nombres* etc.

*Serial.println(Donnée)* : Affiche la valeur de la donnée et force un passage en début de ligne suivante. La *Donnée* est facultative.

*Caractere = Serial.read()* : Effectue une lecture dans le tampon du récepteur série et transfère son code dans l'identificateur *Caractere* qui est une variable de type *char* par exemple. *Si le tampon est vide, retourne la valeur -1.* (Soit aussi le caractère 'y')

**ATTENTION : L'instruction *Serial.available()* n'est plus valide pour tester si un caractère est disponible dans le tampon.**

Pour effectuer la capture d'une saisie sur la voie série, il faut tester le caractère détecté et n'ajouter ce dernier en chaîne de réception que s'il est différent de -1. La fin de saisie sera détectée par le code CR. (13 codé en ASCII sous forme décimale.) Il faut en outre imposer : ☒ Défilement automatique  57600 baud.

... / ...

**NOTE :** Les diverses fonctions présentent une logique particulière. Elles retournent la valeur nulle si le processus est correct et un code d'erreur si un problème se présente. Considéré comme un booléen, ce code vaudra *false* si égal à zéro et *true* si valeur non nulle.

*pf\_read()*;

Fonction qui permet d'extraire un à un les octets *du fichier ouvert*.

Exemple : Listage ligne à ligne sur le moniteur d'un fichier texte :

```
void Liste_le_fichier_index() { // Liste ligne à ligne sur la ligne USB.
uint8_t buf[40]; // Tampon pour l'extraction des octets.
char Tampon[40]; // Tampon de mémorisation de la ligne de texte.
unsigned long pos = 0; // Position courante dans le fichier.
if (pf_open(Nom_Fichier)) Affiche_ERR("Erreur ouverture");
while (1) { // Lecture ligne à ligne du fichier.
    UINT br; // Pointeur de lecture dans le fichier ouvert.
    // On lit le fichier octet par octet avec 1 en deuxième paramètre :
    if (pf_read(buf, 1, &br)) Affiche_ERR("Erreur lecture");
    if (br == 0) break; char carac = (char)buf[0];
    // Ci-dessus : Pour la lecture, on stocke dans la variable "carac".
    // Si on est sur une fin d'une ligne, affiche la chaîne construite :
    if(carac == '\n') {
        Tampon[pos] = '\0'; // On ajoute " \0 " pour marquer la fin de chaîne.
        // On réinitialise la position à 0 pour lire une nouvelle ligne :
        pos = 0;
        Serial.println(Tampon); // Affichage de la ligne extraite.
    } else { // Si ce n'est pas la fin de ligne, on continue à lire le Tampon :
        Tampon[pos] = (char)buf[0]; pos++;
    }
    Serial.println(); Serial.print("Lecture complete.");
}
```

*pf\_write()*;

Fonction qui permet d'écrire *dans le fichier ouvert*.

- Impossible de créer le fichier ou d'en agrandir la taille.
- Impossible de mettre à jour l'horodatage.
- L'opération d'écriture peut démarrer / s'arrêter sur la limite du secteur uniquement et l'attribut en lecture seule du fichier ne peut pas bloquer l'opération d'écriture.

*pf\_lseek ofs*); Déplacer le pointeur de lecture / écriture.

*pf\_opendir()*; Fonction qui ouvre un répertoire.

*pf\_readdir()*; Fonction qui lit un élément de répertoire.

Description sur : [http://elm-chan.org/fsw/ff/00index\\_p.html](http://elm-chan.org/fsw/ff/00index_p.html)

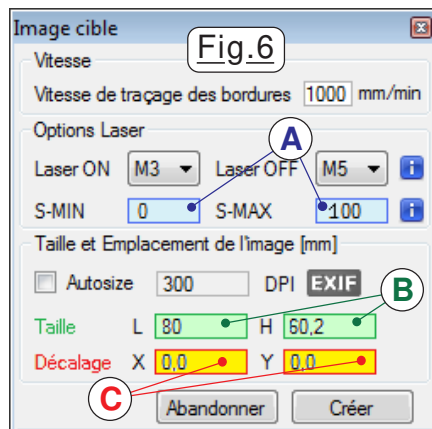
## Utiliser **LASER GRBL v3.0.4.** 3/4

### ➤ Les options pour générer le code gco.

Cliquer sur **Suivant** ouvre la fenêtre contextuelle de la Fig.6 sur laquelle on doit impérativement **forcer des valeurs nulles en C pour avoir l'origine en bas à gauche de l'image** comme c'est le cas sur notre machine pilotée par Arduino. Par ailleurs, le programme d'exploitation sur l'ATmega328 utilise directement les paramètres qui suivent les instructions "S" pour moduler la gestion de la puissance du LASER. Imposer en **A** les valeurs 0 et 100 auront pour effet d'aboutir à une modulation comprise entre 0% et 100%. En **B** il faudra obligatoirement définir les dimensions de l'image, avec possibilité d'en augmenter ou d'en diminuer la grandeur image, sans possibilité d'en changer les proportions. (*Ajuster Systématiquement ces dimensions est impératif, car au chargement d'un fichier, les valeurs précédentes sont conservées.*) **Vérifier que L soit inférieure à 228 mm et H inférieures à 202 mm**, qui sont les valeurs des amplitudes des mouvements mécaniques logiciels sur la machine.

Protocole pour obtenir un fichier gcode à partir d'une image :

- 1) Activer **LASER GRBL** > Fichier > Ouvrir un fichier >
- 2) Sélectionner l'image. > Ouvrir > Choisir l'option de balayage. >
- 3) Ajuster **Lumière, Contraste** ... Orienter, Symétriser etc >
- 4) Dans **Qualité** définir la précision désirée **en imposant 2.000**. >
- 6) **Suivant** > Dans la fenêtre contextuelle de la Fig.6 **imposer en C des valeurs nulles ou positives** pour le **Décalage** de façon à ne générer que des déplacements avec des coordonnées supérieures à zéro.
- 7) Case **Autorise décochée**, adopter en **B** les dimensions désirées pour celles de la gravure LASER : **Impératif à préciser**.
- 9) **Créer** > Fichiers > **Sauver le programme** > Définir le chemin et le nom du fichier. > Sauvegardé en attribut **nc**, il faut renommer en type **gco**. Si résultat incorrect : **[F12]** deux fois pour recommencer.



Page 6

## Utiliser **LASER GRBL v3.0.4.** 2/4

### ➤ Les options de balayage.

Paramètre important, **Qualité 2,000 Nb Lig./mm** conditionne la finesse du tracé, et influence directement la durée de réalisation de la pyrogravure. **Impératif : En imposant 2 la précision sera de 0,5mm.**

- Pour l'option ☒ **Tracé Ligne par Ligne** il sera impératif de privilégier **Horizontal** qui minimise les déplacements sur Y. **Surtout ne pas sélectionner Vertical qui engendre dans le code gco des lignes commençant par Y qui seront refusée par le logiciel d'Arduino.** Cette option accepte **NB** cochée ou décochée. (Voir Fig.4)
- L'option ☒ **1bit NB Pointillisme** qui procède par un fonctionnement "tout ou rien" du LASER génère une image formée de points. Comme on peut l'observer sur la Fig.3 proposée ci-contre, les effets de dégradés sont obtenus par la densité variable des points étalés le long de la ligne pyrogravée.
- Comme montré sur la Fig.5 l'option ☒ **Vectorise !** aboutit à un effet de "contournage". La puissance du LASER sera modulée même si la case **NB** est cochée et son curseur poussé tout à droite.



Fig.3

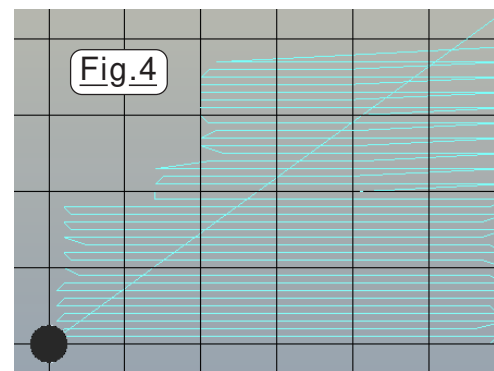


Fig.4

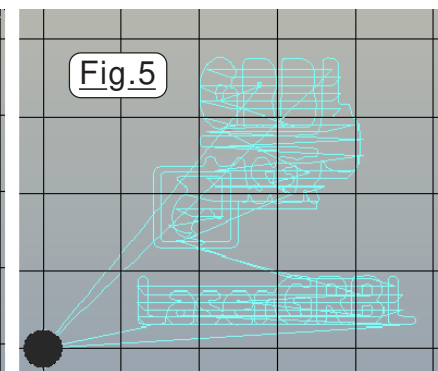


Fig.5

- Ne pas utiliser l'option ☒ **Centerline** qui sur la version installée du logiciel engendre un texte d'alerte **Unknown error (0xd8)**.

**NOTE :** La fenêtre contextuelle de l'option **C** sur la Fig.2 ne s'ouvre que si le fichier image chargé est un dessin réalisé en couleur.

Tutoriel en VF : <https://www.youtube.com/watch?v=-h9sA9x5fbE>



## Utiliser **LASER GRBL v3.0.4**. 1/4

N'utilisant pas le logiciel **LASER GRBL** pour piloter directement une machine par la ligne série USB, mais uniquement pour générer un fichier **gcode** qui sera sauvegardé sur une carte SD, seules quelques fonctions de base seront pertinentes.



### ➤ Paramétrage initial.

Après avoir utilisé **Créer** la seule façon de reprendre les ajustements s'ils ne sont pas satisfaisants consiste à recharger l'image. Par défaut c'est la commande **[CTRL] R** qui est attribuée. Située juste au dessus du **B.S**, la commande **[F12]** me semble plus conviviale et facile à retenir. Procédure pour imposer ce choix :

- 1) Commande **Grbl** > **Hotkeys** > Indexer **ConnectDisconnect** >
- 2) Par exemple proposer "c" pour libérer **[F12]** >
- 3) Indexer la ligne **ReopenFile** > **[F12]** > **Save** pour valider.

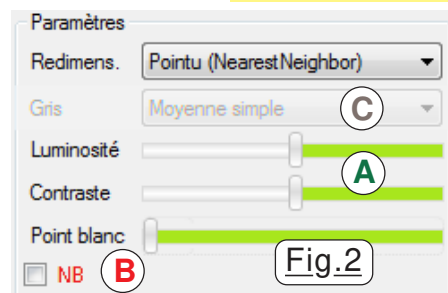
### ➤ Traitements de l'image.

Classiques, les commandes visualisées sur la Fig.1 permettent en **2** et **3** d'effectuer des rotations de 90°, et en **4** et **5** les symétries habituelles. L'icone **6** sert à réaliser l'extraction d'une zone qui deviendra l'image complète alors que **7** inverse "les couleurs". Attention, à tout moment la commande **1** fait revenir à l'image initiale et perdre toutes les modifications antérieures. Il importe de noter que l'onglet de visualisation **Original** subit également les transformations apportées en mode **Aperçu**.



Fig.1

### ➤ Gestion de la pyrogravure.



Les paramètres montrés en Fig.2 n'influencent que le mode **Aperçu**. En **A** on agit sur les seuils du traitement. Si la case **NB** en **B** est cochée le LASER est piloté en "tout ou rien" alors que décochée la puissance est modulée. On peut à convenance avoir une visualisation du balayage effectué sur l'image en cochant l'option ☒ **Aperçu des lignes**.

## Utiliser **LASER GRBL v3.0.4**. 4/4

Quelques valeurs et quelques conseils issus de l'expérience peuvent aider à choisir des paramètres pour transposer une image en un fichier **gco** pour la pyrograveuse. La même image peut aboutir à des effets totalement différents en fonction des paramètres imposés à **LASER GRBL**. Exemple typique :

### ➤ Générer le plus gros fichier gco possible.

Obtenir le fichier contenant un maximum de lignes impose en préambule une image noir et blanc avec inversion de la modulation du LASER à chaque mouvement de ce dernier, et une image la plus grande potentiellement soit 202mm x 228mm. Les plus petits déplacements sur chaque axe sont de 0,5mm. L'image sera donc composée d'un damier de 404 x 456 petits carrés. Cette image nommée **Image 202 x 228.bmp** est disponible dans **<ARCHIVES>**. Paramétrages à adopter :

Redimens : **Pointu (Nearest Neighbor)**      Qualité **2,000** Nb Lig./mm  
**Luminosité** : 160. (Curseur à droite.)      ☒ **NB**  
**Contraste** : 160. (Curseur à droite.)  
**Pointblanc** : 0. (Curseur à gauche.)      @  
**Taille** : 202.0 x 228.0

Avec ces paramètres, on obtient un fichier de 92.064 lignes avec changement de puissance LASER à chaque mouvement. Nommer ce fichier **Img9.gco** et ajouter les deux lignes de commentaire :

```
; Img9 MAX 202 x 228.  
; 92066 lignes.
```

### ➤ Paramètres pour une image en dégradés de gris.

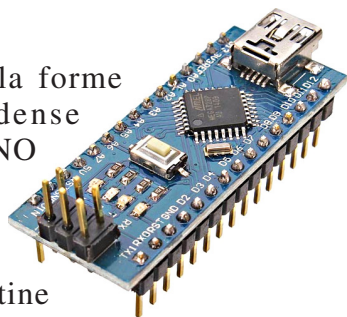
Extrait du **LOGO gris.bmp** les paramètres qui conduisent au résultat @ sont les suivants :

Redimens : **Pointu (Nearest Neighbor)**      Qualité **2,000** Nb Lig./mm  
**Luminosité** : 100. (Curseur au centre.)      ☐ **NB**  
**Contraste** : 100. (Curseur au centre.)      @  
**Pointblanc** : 0. (Curseur à gauche.)  
**Taille** : 10.0 x 10.0 (On aboutit à un fichier de 352 lignes)

@ : Si la case ☐ **Aperçu des lignes** est décochée, quand on passe de l'onglet **Aperçu** à **Original**, visuellement il n'y a aucune différence.

## Carte Arduino NANO.

L'arduino NANO se présente sous la forme d'une minuscule carte qui condense l'intégralité des fonctions d'une Arduino UNO tout en ne mesurant que 1,9 cm x 4,5 cm. La NANO utilise l'ATmega328 en version CMS. Les broches d'utilisation sont séparées pour pouvoir la placer sur une platine d'essais classique. Arduino NANO peut être alimentée soit par le connecteur Mini-USB soit en externe avec +6V à +20V non régulé sur la broche **VIN**. **ATTENTION : Pas de VIN simultanément avec la liaison Mini-USB ou la carte électronique sera détruite.** Alimentée par le connecteur Mini-USB la carte fournit  $\approx 4,2\text{V}$  sur la broche **5V** pour alimenter des modules périphériques. Cette broche peut également être alimentée en +5Vcc simultanément avec la prise Mini-USB. On peut ainsi alimenter la carte par la broche **5V**, sur son électronique d'application, tout en branchant en parallèle la ligne USB pour programmer sur site et dialoguer avec le Moniteur de l'IDE.



14 broches binaires. (Dont 6 fonctionnant en PWM.)

8 broches d'entrées Analogiques dont 6 pouvant fonctionner en E/S. Courant maximal par broche de sortie : 40 mA. (Total MAX : 100mA)

L'ATmega328 a 32 Ko, (Avec 2 KB utilisé pour le bootloader).

L'ATmega328 a 2 Ko de SRAM et 1 Ko de mémoire EEPROM.

Par rapport à la carte Arduino UNO la NANO présente **deux entrées Analogiques supplémentaires A6 et A7**. Elles ne peuvent pas être utilisées en E/S binaires, mais uniquement en entrées analogiques et ne disposent pas de résistances PUL-UP internes. Inutile de les déclarer en entrée, on les utilise directement avec la syntaxe standard `analogRead(20)` et `analogRead(21)`.

De nombreux clones chinois de ces cartes existent. Certains sont basés sur des circuits intégrés tels que l'ATmega328P CH340G qui ne sont pas reconnus directement sous Windows sans installer un pilote spécifique. Pour mes cartes le driver adapté a été trouvé sur : <http://www.mediafire.com/download/pjqn88uc64acpgz/ch341ser.zip> L'exécutable une fois activé la ligne USB fonctionne normalement et les NANO se programment sans problème à condition :

- De sélectionner le type **Arduino Nano** avec **Outils ...**
- D'activer le bon port pour la ligne USB.

- 8) Brancher le fil rouge avec un connecteur à trois broches sur la sortie **S11** du multiplexeur. (*S11 ou toute autre sortie libre.*)
- 9) Connecter la ligne d'éclairage à deux fils verts et jaunes.
- 10) Soulever à convenance le PUPITRE pour pouvoir aisément insérer le connecteur double sur le lecteur de carte SD. Le toron part vers le haut, calé contre le commutateur des calibres.
- 11) Brancher le CLAVIER, (*Limande grise à quatre fils : Voir Fig.1*) puis l'immobiliser provisoirement sur le deux colonnes qui supportent le plateau de la machine.
- 12) Débrancher la fiche d'énergie 12v arrivant du bloc secteur.
- 13) Glisser sous tous les torons de fils la ligne grise à cinq fils qui va sur la platine des shunts. Rebrancher la fiche 12v.
- 14) Déplacer manuellement le chariot Y'Y sur toute la course possible et vérifier le dégagement total sans aucun obstacle.

## Intervention sous le plateau ou sur le PUPITRE.

**Commencer par interdire toute activation possible du LASER de puissance**, son allumage intempestif pourrait endommager des éléments se trouvant sur le statif de la machine.

- 1) Retirer les quatre écrous qui immobilisent le plateau sur les colonnes et déposer la rallonge de ce dernier située à l'arrière.
- 2) Soulever légèrement et libérer sur le PUPITRE les éléments : **Codeur rotatif incrémental / Afficheur OLED / CLAVIER. @**
- 3) En avançant légèrement l'ensemble enlever le boulon central qui immobilise le PUPITRE sur le plateau de la machine.
- 4) Déposer entièrement le plateau et replacer le pupitre sur les deux colonnes situées à l'avant de la machine. L'immobiliser.
- 5) Si l'intervention l'exige, rebrancher les trois éléments @.
- 6) Si intervention sur le PUPITRE :
  - Débrancher toutes les autres liaisons.
  - Déposer le pupitre avec protection idoine sur la face coté B.P.
- 7) Replacer les écrous sur les rondelles plates des colonnes arrières.
- 8) Intervention terminée :
  - Si déposé : Réinstaller le PUPITRE,
  - Rebrancher éventuellement toutes les lignes du PUPITRE,
  - Valider le fonctionnement de tous les éléments du PUPITRE,
  - Tout ce qui a été modifié sous le plateau est réinstallé,
  - Replacer le plateau et sa rallonge sur les quatre colonnes.

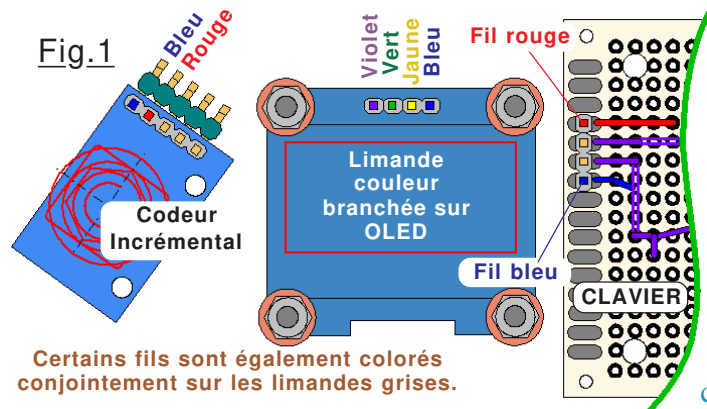
**ATTENTION au pincement des fils entre plateau et rondelle plate.**



## Protocole d'intégration du PUPITRE.

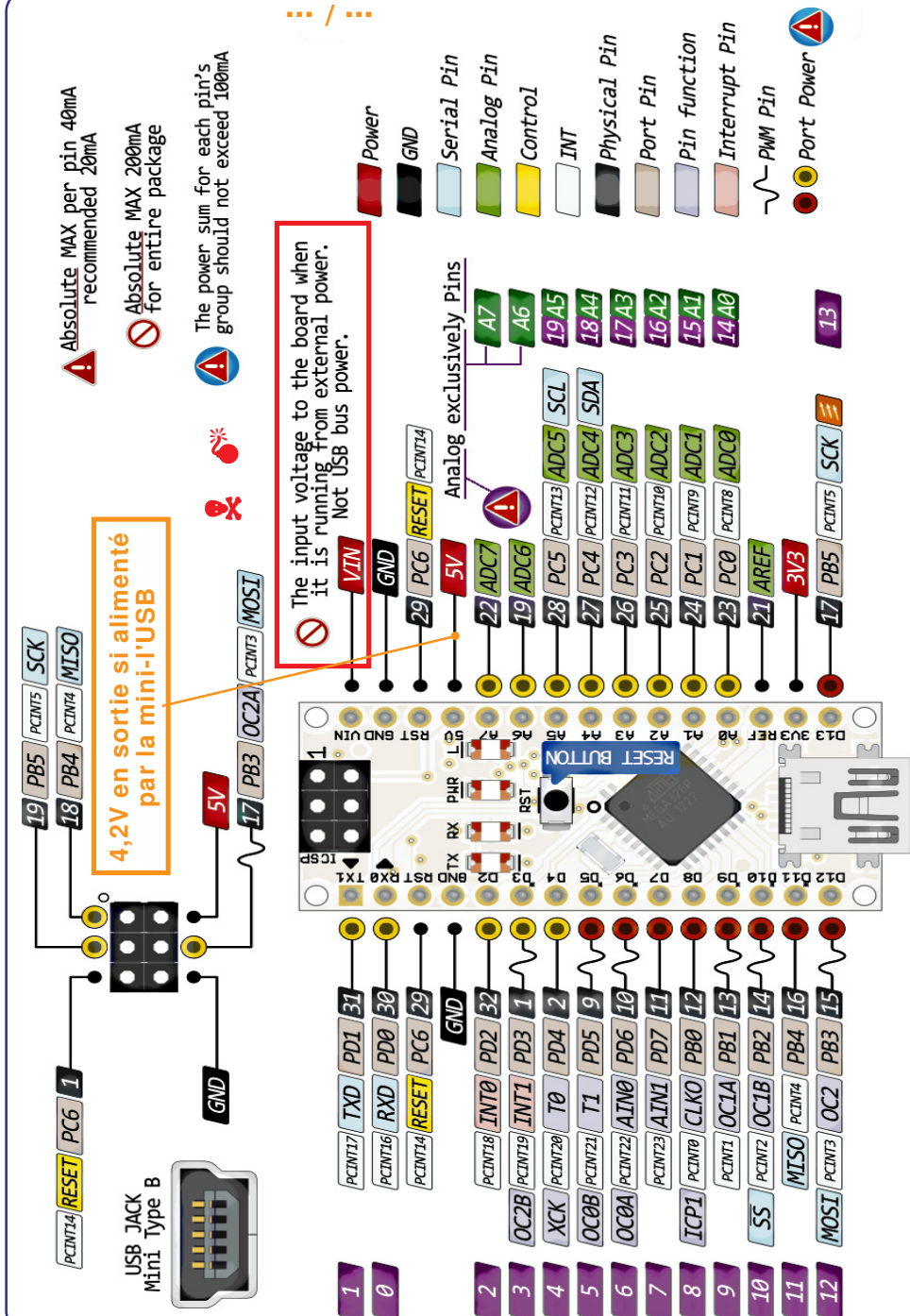
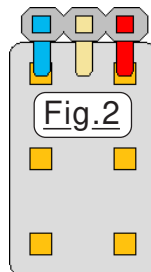
Déposé "au sol" le long du statif sur une protection souple, la procédure tient compte de la longueur des torons de liaison et de l'accessibilité des divers connecteurs que ce soit sur le pupitre ou sur la machine. Cette procédure constitue également la validation électrique et électronique du PUPITRE. Au préalable on a vérifié la coupure de **GND** par le bouton de RESET sur la ligne qui va au lecteur de carte SD. Le petit tableau des LEDs est branché. Enfin, les écrous des deux colonnes avant qui supporte le plateau sont enlevés pour pouvoir y placer provisoirement le PUPITRE.

- 1) Le bloc alimentation secteur est débranché. (*Pas d'énergie.*)
- 2) La SÉCURITÉ ÉNERGIE est verrouillée. (*Pas de puissance.*)
- 3) **La liaison alimentation 5Vcc est débranchée sur la carte multiplexeur PCA9685 à 16 canaux et sur la carte Arduino NANO.**
- 4) Brancher le codeur incrémental. (*Ligne à 5 fils et limande grise. Voir Fig.1*) **Inverser ce connecteur détruira le régulateur 5V !**
- 5) Connecter l'afficheur OLED. (*Ligne à multicolore à 4 fils.*)



Les différents connecteurs HE14 ont leurs broches colorées pour en préciser l'orientation. La Fig.1 donnée ci-contre précise les couleurs de repérage pour certaines broches.

- 6) Enfiler le HE14 à huit broches coudées sur le sélecteur des calibres de mesurage. (*Limande grise qui va vers le répéteur à LEDs.*) Insérer les huit broches résiste un peu. Il importe donc de soulager le connecteur HE14 coudé en le soutenant du côté opposé.
- 7) Brancher le connecteur HE14 à trois broches, ligne grise à deux fils qui part vers l'arrière sur l'inverseur "multimètre". (*Voir la Fig.2 donnée ci-contre*)



## Instructions gcode étendues.

Simplifier au maximum l'analyseur syntaxique en optimisant le **gcode** issu de **LASER GRBL** n'interdit pas de prévoir diverses instructions complémentaires bien utiles pour la mise au point de la machine, ou surtout pour créer manuellement des "programmes image" **gcode** spécifiques avec un quelconque éditeur de texte et les déverminer. En particulier des images de calibrage pour la machine par exemple. Aussi, l'analyseur syntaxique a été complété pour accepter, outre la liste des codes déjà explicités, les éventuelles instructions complémentaires suivantes :

- **Snn** : Modifier librement la puissance LASER.
- **G1 Xnn Ynn** : **G1** avec paramètre **Y** et sans paramètre **S**.
- **G1 Xnn Ynn Snn** : **G1** complet avec les trois paramètres.
- **G0 Xnn Ynn** : Les ligne **G0** sans le **S0** sont autorisée.

En particulier, les deux instructions de type **G1** engendrent des déplacements avec LASER activé dans des directions vectorielles, c'est à dire en ligne droite pour des mouvements "diagonaux".

### ► Les codes sans déplacement.

- **M0** : Éteint l'écran par programme.
  - **M1** : Rétablit l'écran par programme.
- Il est ainsi commode lors de la validation d'un programme image ou de tests divers sur la machine, de n'afficher que les instructions **gco** pertinentes durant l'opération de gravure.
- **M2** : PAUSE. Quand ce code est rencontré, LASER éteint, l'écran affiche **PAUSE en ligne nnn**. Le CLAVIER attend une touche quelconque pour déclencher la reprise, sa LED verte clignote. On peut en intercaler librement dans tout le fichier image.
  - **M4 nn** : Modifie automatiquement la valeur de la variable nommée **Puissance\_MAX\_sur\_le\_LASER**. Bien faire ATTENTION à ce que la valeur soit dans les limites [**1** à **100**] car cette données n'est pas vérifiée. En sortie de pyrogravure elle sera conservée jusqu'à un RESET ou changée dans le menu **OPTIONS**.
  - **M6 nnnnn** : Change la valeur du **Délai** la donnée étant exprimée en millisecondes. ATTENTION : La valeur doit rester dans les limites [**10** à **9990**] car cette données n'est pas vérifiée. (Si on dépasse, l'affichage sera incorrect.) En sortie de pyrogravure elle sera conservée jusqu'à un RESET ou changée dans le menu **OPTIONS**.

## Pilotage manuel par la ligne USB du Moniteur.

Valider les routines telles que l'analyseur syntaxique ou celles qui effectuent les actions de pyrogravure impose de respecter scrupuleusement la structure d'un fichier tel qu'il sera créé avec **LASER GRBL** et se conformer tout particulièrement aux valeurs des divers paramètres précisés dans la fiche **Utiliser LASER GRBL v3.0.4**. La chronologie valide d'une séquence de pilotage USB est proposée dans l'exemple typique donné ci-dessous :

```
G90
G0 X50 Y200
M3 S0
S0
F1000
; -----
X60 S50
X70 S1
M1
G0 X10 Y190 S0
M0
G1 X50 S30
; Texte quelconque
(Ligne vide)
G0 X150 Y170 S0
G0 X85 Y173 S25 (Code G0 : Valeur de S ignorée et % = 0)
G1 X1 S1
M2
M433 (*)
G1 X3 Y3
G0 X2 Y190 S5
M6 1234 (*)
S 73
S 0
S 66
G1 X5 Y194 S0 (Code G1 : Valeur de S ignorée et % conservé)
; -----
M5
G0 X0 Y0
```

- En rouge **les instructions impératives**.
- En vert **le corps de l'image**.
- En gris des instructions **gco** possibles générées par **LASER GRBL** mais qui sont ignorées par le programme **P20**.
- **nn** dans la plage 0 à 100 pour **S**.
- **nn** dans la plage 0 à 228.0 pour **X**.
- **nn** dans la plage 0 à 202.0 pour **Y**.
- **nn** dans la plage 0 à 100 pour **M4**.
- **nn** dans la plage 10 à 9990 pour **M6**.
- Pour **X** et **Y** une décimale n'est pas obligatoire et sera **.0** ou **.5** si précisée.

Cet exemple de pilotage par la ligne USB du Moniteur de l'**IDE** utilise toute la combinatoire des codes **gco** valides. Vérifier que l'état qui en résulte sur la machine est cohérent. **Attention : Il n'y a pas de vérification du contenu des commandes, des lignes incorrectes peuvent bloquer le programme sur Arduino.**

(\*) Valeur à vérifier dans **OPTIONS** après être sorti du dialogue **USB**.

## Pilotage manuel par la ligne USB du Moniteur.

Effectuer la validation du logiciel s'avère très indigeste si la seule façon d'envoyer les consignes consiste à créer manuellement des "Images" spécifiques en code **gco** que l'on doit pour chaque tentative modifier et réinscrire sur la carte SD. Le Mode MANUEL quand à lui n'est pas du tout pertinent pour développer ou modifier du logiciel. De loin, la technique la plus adaptée consiste à envoyer ligne à ligne des instructions en **gco** directement à partir du Moniteur de l'IDE et de voir comment réagit le programme.

Comme il n'est pas possible par manque de place d'ajouter cette fonction dans **P20\_Programme\_exploitation.ino**, un programme spécial nommé **P30\_pilotage\_par\_USB.ino** doit être téléversé sur la carte NANO. Ce n'est pas pénalisant en ce sens que ce programme ne sera utilisé qu'en cours de développement. Dès que les routines de base sont au point, on les retranscrit dans le programme **P20** que l'on téléverse à nouveau sur Arduino. Pas besoin de modifier le contenu de l'EEPROM pour ce pilotage par la voie série.

Manuellement on risque fort de frapper au clavier du P.C. des **consignes non valides entraînant un comportement imprévu du programme**. Il suffit d'oublier un espace, de saisir une mauvaise lettre etc. Aussi, il est fortement recommandé de **passer en SÉCURITÉ et couper la puissance 12v**. Quand le programme fonctionne correctement, il est alors possible de **rétablir la puissance, mais alors couper le LASER ou mettre des lunettes de protection spéciales**.

- 1) Si **SÉCURITÉ** masquer les capteurs avec du carton par exemple, pour faire croire que les deux origines machines sont effectives.
- 2) Dans le menu de base activer **OPTIONS** ajuster à convenance le pourcentage LASER et le **Délai** de ralentissement.
- 3) Dans le menu de base invoquer l'item **Moniteur USB** :
  - Le programme affiche **Capture Origines** et allume la LED verte.
  - Le Moniteur affiche **Proposer des consignes USB valides** :
  - L'écran OLED affiche **X=0.0 / Y = 202.0 / P = 0** pour 100.

Le fonctionnement du dialogue série ne sera correct que si la vitesse sur le Moniteur est imposée à **11520 baud**, que valider génère un **Retour chariot** et que la case **Défilement automatique** soit cochée.

- 4) Verrouiller [MAJ] sur le clavier du P.C. (*Simple conseil.*)
- 5) Envoyer dans l'ordre strict décrit au verso des commandes valides.

... / ...

## Résumé des contraintes pour les fichiers gco.

Strictement impératives, certaines contraintes doivent absolument être respectées ou une réaction non prévue du logiciel peut engendrer une divergence du comportement de la machine. C'est particulièrement vrai si les fichiers sont modifiés manuellement.

- Les premières contraintes à respecter concernent la génération du fichier **gco** avec **LASER GRBL** en respectant les paramètres précisés dans les fiches **Utiliser LASER GRBL v3.0.4**.
- Le programme **LASER GRBL** ne génère aucun commentaire, on peut toutefois en ajouter librement, ils seront ignorés sauf si ils sont placés sur les deux premières lignes.



**ATTENTION** : Si on ajoute des lignes manuellement dans le code **gco** il ne faut pas que les lignes dépassent 25 caractères où le tableau **Existe[]** sera "écrasé" en mémoire Arduino.

- Non obligatoire la première ligne sera considérée comme un "TITRE" qui sera affiché dans le menu **FICHIER** par le B.P. **Y-↓**.
- Non obligatoire la ligne commentaire n°2 contiendra un texte court du genre **" ; 19 lignes."** qui précise la taille du fichier qui sera également affichée dans le menu **FICHIER** par le B.P. **Y- ↓**. (*Utilisable si commence par ' ; ' suivi d'un nombre de lignes.*)

### **➤ Instructions générées par LASER GRBL.**

Les seules instructions générées en respectant les critères imposés sur la fiche **Utiliser LASER GRBL v3.0.4**. 2/4 sont :

**G90** (*Utiliser le positionnement absolu : Obligatoire.*)

**G0 Xnn Ynn** (*Déplacement sur l'origine de l'image : Obligatoire.*)

Seule ligne de type **G0** (*Et celle à la fin*) qui ne contient pas **Snn**.

**M3 S0** (*LASER ON à 0% : M3 Obligatoire mais pas S0.*)

**S0**

**F1000.** (*Ligne ignorée sur la graveuse.*)

**Xnnn.n Snn** (*Commence par X, obligatoirement suivi de S*)

**G0 Xnnn.n Ynnn.n S0**

**G1 Xnnn.n Sn**

**M5** (*LASER ON à 0% : Obligatoire.*)

**G0 X0 Y0** (*Retour rapide en configuration dégagée.*)



.n est soit le chiffre 0, soit 5.

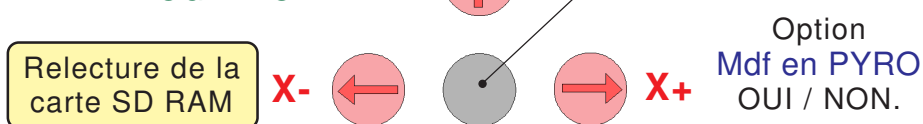


## Menu PYROGRAVURE.

Affiche la liste des fichiers valides établie lors d'une lecture de la carte SD RAM. Sur RESET ou à la mise sous tension de la une lecture initiale de la carte SD est effectuée.

- Le B.P. central du codeur rotatif engendre le retour au **Menu de base**. Si la PYROGRAVURE d'un fichier est en cours, **engendre la sortie anticipée** de cette dernière si on confirme avec **Y+ ↑**.
- La rotation du codeur incrémental change l'indexation du fichier pointé dans la liste des images potentielles, avec un effet de "recirculation" pour les limites écran **Img0** et **Img9**.

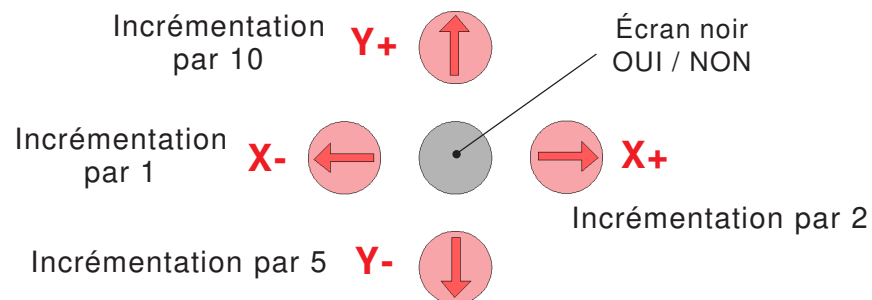
Déclenche une PYROGRAVURE **Y+ ↑** Écran noir OUI / NON



Caractéristiques de l'image indexée. (1) **Y- ↓** (1) : Gravure :  $\pm nnn$  précise l'évaluation statistique très approximative de la durée que prendra la pyrogravure avec l'écran noir.

### ► Durant la PYROGRAVURE.

- Le B.P. central du codeur rotatif déclenche la sortie anticipée si on confirme avec **Y+ ↑**. **Il faut appuyer sur le bouton central du codeur incrémental jusqu'à sa prise en compte.** (La LED clavier jaune s'allume.) En situation d'urgence effectuer un RESET.
- Si dans le menu **OPTIONS** on a validé l'item **Mdf en PYRO** il sera possible de **Modifier** la puissance du LASER durant la pyrogravure. La rotation du codeur augmentera ou diminuera d'une unité la valeur de la Puissance MAX par défaut, le CLAVIER modifie l'incrément :

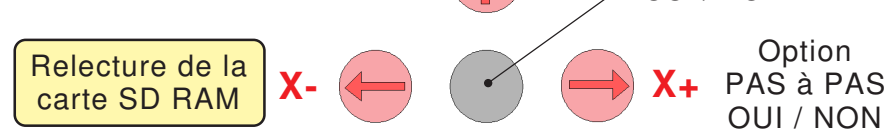


## Menu FICHIERS.

Affiche la liste des fichiers valides établie lors d'une lecture de la carte SD RAM. Sur RESET ou à la mise sous tension de la une lecture initiale de la carte SD est effectuée.

- Le B.P. central du codeur rotatif engendre le retour au **Menu de base**. Si l'analyse syntaxique d'un fichier est en cours, **(Et écran noir.) permet la sortie anticipée** de cette dernière si on confirme avec **Y+ ↑**.
- La rotation du codeur incrémental change l'indexation du fichier pointé dans la liste des images potentielles, avec un effet de "recirculation" pour les limites **Img0** et **Img9**.

Analyse syntaxique du fichier indexé **Y+ ↑** Écran noir OUI / NON









Caractéristiques de l'image indexée. (1) **Y- ↓** (1) : Tmp Analyse précise une évaluation de la durée que prendra une vérification complète du fichier l'écran étant noir.

- Comportement sur sortie anticipée demandée :
  - \* **FIN de procédure ?** > Si B.P. **Y+ ↑** affiche **Syntaxe correcte** (Information vraie uniquement pour la partie du fichier vérifiée.) et attend un B.P. pour retour au menu **FICHIER**.
  - \* Si réponse négative, affiche "L'état machine" et attend une touche pour passer à la suite. On sait ainsi où en est l'analyse du fichier. L'analyse reprend, mais l'écran reste affiché pour que l'opérateur puisse le lire. Le B.P. noir deux fois permet de reprendre le mode rapide écran noir et PAS à PAS suspendu.
- Durant l'analyse d'un fichier ou la relecture de la carte SD le codeur incrémental peut décaler l'indexation d'une ou deux positions. (Effet peu pénalisant puisqu'il est visuel car affiché sur OLED.)

**Durant l'analyse syntaxique d'un fichier**, le B.P. noir alterne entre le mode rapide "écran noir" et le fonctionnement avec affichage de "L'état machine" à chaque ligne de code. En mode rapide, si l'option PAS à PAS est activée, elle est suspendue jusqu'au rétablissement de l'affichage avec la touche noire.



### Le menu des OPTIONS.

**R**elatif aux opérations de pyrogravure il est prévu pour définir préalablement la puissance maximale qui sera possible sur le LASER et le Délai de maintien en position à chaque instruction de mouvement lors d'un pas de 0,5mm. En particulier les valeurs mémorisées seront prises en compte par le Mode MANUEL et dans le menu SYSTEME par l'item Mode LASER continu. Noter que ces *valeurs modifiées* par M4 ou M6 seront également *mémorisées*.

- Le codeur rotatif  modifie l'option indexée.
- La Touche Y+  sélectionne la modification du % LASER.
- Le B.P. noir force LASER 50 / 100 et Délai : 5.00s.
- La Touche Y-  sélectionne la modification du Délai.
- Cliquer sur X-  définit un Incrément / Décrément de valeur 1.
- Cliquer sur X+  impose un Incrément / Décrément de valeur 10.
- Le bouton central du codeur rotatif  ramène au **Menu de base**.

### Le menu SYSTEME.






**C**ontenant les fonctions qui facilitent la mise au point ou la maintenance de la machine et du logiciel, on trouve surtout l'item Mode LASER continu parfois utile en pyrogravure pour déterminer les Délais d'exposition pour les matériaux résistants.






- Le codeur incrémental  permet de sélectionner l'item dans la liste.
- Le bouton central du codeur rotatif  active l'option indexée.
- **Dégagement plateau** : Après capture de l'Origine Machine, La pyrograveuse est "dégagée", puis il y a retour au menu SYSTEME.
- **Mode LASER continu** : Voir la fiche dédiée. La sortie de ce mode d'utilisation de la pyrograveuse ramène au menu SYSTEME.
- **U sur capteur ORG** : Affiche en boucle les valeurs numérisées par les capteurs d'Origine Machine X'X et Y'Y et précisent la conclusion déduite par le logiciel en fonction des seuils de détermination.
- **Test capteur ORG** : Vérifie le bon fonctionnement des deux capteurs d'Origine Machine X'X et Y'Y et indiquent la conclusion sur la LED tricolore. Puis il y a retour au **Menu de base**.
  - \* LED **vert cyan** : Les deux capteurs sont opérationnels.
  - \* LED **violet** : L'un au moins des deux capteurs est soit en panne, soit masqué car la machine n'est pas en configuration dégagée.

Si sur un RESET ou à la mise sous tension la **LED triple s'illumine en violet**, il importe de **dégager manuellement la machine en utilisant l'option Dégagement plateau du menu SYSTEME**.



### Protocole standard pour effectuer une gravure.

**B**ien que toutes les étapes ordonnées conseillées dans cette procédure ne soient pas indispensables, c'est la méthode la plus rationnelle pour engager une opération de pyrogravure.

- Commencer dans le menu **OPTIONS** par vérifier la valeur de la *puissance maximale sur le LASER* et celle de la *temporisation à chaque position* ces deux paramètres étant fondamentaux. (1)
- Activer le menu **PYROGRAVURE**.
- Insérer dans le lecteur une carte SD contenant l'image désirée.
- Cliquer sur X-  pour établir la liste des fichiers image détectés.
- Codeur incrémental  pour sélectionner l'image dans la liste.
- Touche Y-  pour vérifier que c'est le modèle désiré, puis cliquer sur une touche quelconque du clavier pour sortir de l'écran d'informations.
- Si le nombre de lignes est important et que ce fichier n'a jamais été vérifié il est *recommandé* d'effectuer une *analyse syntaxique*,  la durée estimée du traitement étant précisé en bas de l'écran. 

- \* Sortir de **PYROGRAVURE** avec le B.P. central du **C.I.** .
- \* Activer le menu **FICHIER** avec  et .
- \* Codeur incrémental  pour sélectionner l'image dans la liste.
- \* Cliquer sur Y+  pour déclencher l'analyse syntaxique.
- \* À convenance, bouton noir du CLAVIER pour éteindre l'écran.
- \* Si "OK" *Reprendre au tout début le protocole de pyrogravure.*

**C.I.** désigne le **C**odeur **I**ncrémental.


- Cliquer sur Y+  pour commencer le processus de pyrogravure, la machine positionne le LASER sur l'origine image pour placer la cible.
- Vérifier que l'inverseur **Machine** est sur position **Multimètre**.
- Passer l'inverseur Coupé sur la position **LASER**.
- Positionner l'objet cible sur le plateau pour y cadrer l'image prévue.
- Effectuer la focalisation du LASER. À partir d'ici : *Lunettes de protection spéciales impératives pour toutes les personnes présentes.*
- *À ce stade on peut sortir de la procédure avec le B.P. noir*, toute autre B.P. rouge déclenche la pyrogravure. Fin de gravure, repasser **LASER** sur **Coupé**. Durant la gravure la *sortie anticipée* avec  peut être activée, le **B.P.** central *étant maintenu jusqu'à sa prise en compte*.



(1) : La valeur de **Délai** qui vaut **0.01s** sur **RESET** ne peut être modifiée que dans le menu **OPTIONS** ou par programme image avec l'instruction **M6 nnnnn**. (Voir fiche sur les codes complémentaires.)

## Le mode MANUEL.

Lorsqu'il est validé dans le menu de base, le programme effectue une capture des Origines Machine, pour pouvoir ensuite afficher correctement les coordonnées de position du LASER, puis il demande de **Focaliser le LASER** car ce ne sera plus possible après cette phase initiale. La LED verte du CLAVIER clignote pour inciter l'opérateur à cliquer sur une touche. Dès que l'opérateur obtempère, quel que soit le bouton utilisé, l'écran affiche l'**état machine** :

Position X : 0.0	} Coordonnées de l'Origine Machine.
Position Y : 202.0	
Chaque B.P. : 10.0mm	} Valeurs définies dans OPTIONS.
Délai par pas : 0.01s	
LASER NON	



- En standard le bouton central du **C.I.**  ramène au **Menu de base**. Le programme d'exploitation en préalable à la sortie du mode **replace la machine en configuration plateau dégagé**.
- L'économiseur d'écran n'est pas pertinent pour cette option car peu utile. (Le B.P. noir permet alors d'agir sur le LASER.)
- Dans ce mode le LASER ne s'allume que pendant les déplacements.
- Le B.P. noir arme ou coupe le LASER, la LED répéteur s'allume alors entièrement quelle que soit la valeur maximale programmée. En fin de déplacement elle s'éteint sauf s'il y a débordement.




- Par défaut sur RESET le déplacement par clic sur un B.P. rouge sera de 10mm. On peut alterner librement entre les deux valeurs de **0.5 mm** ou **10.0 mm** avec le codeur rotatif  en fonction de son sens de rotation, la valeur étant actualisée sur l'écran OLED si on est en option **LASER NON**.
- Si l'option **LASER OUI** est active, le codeur rotatif  incrémente ou décrémente la valeur de **Délai par pas** de **0.01s** en fonction de son sens de rotation de **0.01s** à chaque "pas".

**NOTE : Il n'y a aucune limite pour les positions.** Les coordonnées peuvent devenir négatives ou supérieures aux courses maximales possibles sur la machine. La mécanique est toutefois protégée et ne va pas en butée, le moteur concerné se fige aux butées logicielles. Les coordonnées affichées sur l'écran OLED sont celles fictives correspondant aux commandes opérateur. Tout dépassement génère un BIP sonore d'avertissement, et ce à chaque activation d'une touche, jusqu'à ce que le débordement virtuel soit résorbé.

## Informations et conseils divers.

**IMPORTANT :** Il est fortement recommandé de **toujours achever une séquence d'utilisation en laissant la machine en configuration dégagée**. Si l'un des détecteurs d'Origine Machine est masqué, la capture peut être faussée, car rien ne garantit que les chariots n'ont pas été décalés manuellement vers la butée physique. De ce fait les coordonnées établies en capture origine seraient fausses. C'est la raison pour laquelle sur RESET le programme teste les capteurs. **Si la LED triple s'illumine en violet, il importe de dégager manuellement la machine en utilisant l'option Dégagement plateau du menu SYSTEME.**

 **IMPORTANT :** En Délai de 0.01s l'affichage de l'**État Machine** ralentit considérablement les déplacements. Le temps de gravure est pratiquement multiplié par trois.   
**CONCLUSION : En gravure "standard" éteindre l'écran.**

 **IMPORTANT :** Si suite à une manipulation FICHIER ou PYROGRAVURE en sortie les 10 fichiers sont affichés   
"---" c'est que l'un de ceux qui ont été traité comporte une ligne probablement de type ";" **supérieure à 25 caractères.** 

**NOTE :** Dans LaserGRBL il faut privilégier les lignes **horizontales** car elles minimisent les déplacements sur Y'Y qui sont plus brutaux puisque les masses déplacées y sont plus importantes.

**NOTE :** Quand le programme est en attente d'une réponse de type OUI / NON, seule la touche **Y+ correspond à OUI**. Toutes les autres touches seront interprétées comme une négation.

### **Signification des couleurs pour la LED triple :**

- **Bleu** : Moteurs en VEILLE.
- **Rouge** : Moteurs activés pour capturer l'Origine Machine.
- **Vert** : Origine machine capturée, coordonnées initialisées.
- **Violet** : Capteurs d'origine machine défaillants ou masqués.
- **Vert cyan** : Capteurs d'Origine Machine effectifs.



## Rangement / Mise en service de la machine.

### ➤ Remiser la pyrograveuse en "configuration propre".

- Si ce n'est pas le cas, utiliser l'item **Dégagement plateau** dans le menu **SYSTEME**, la machine étant encore en énergie de puissance.
- Si la machine est en cours de maintenance, enlever si elle est branchée la ligne USB d'une alimentation auxiliaire ou d'un dialogue **IDE**.
- Inverseur **Machine** placé sur position **Multimètre**.
- Passer l'inverseur **LASER** sur la position Coupé.
- Placer le commutateur de calibre sur X : 500mA. (*LED jaune.*)
- Pour préparer la prochaine remise en service, si nécessaire rétablir l'indication de la tension sur le régulateur **5.0v**.
- Couper l'éclairage du plateau.
- Passer le contacteur de SÉCURITÉ en position verrouillée à droite.
- Laisser une carte SD valide dans le lecteur pour éviter un problème de lecture à la prochaine mise sous tension. S'assurer dans le menu **FICHIERS** que la lecture de la carte SD présente est sans problème.
- Machine en attente dans le menu de base débrancher l'alimentation du bloc secteur. (*Éventuellement enlever le fil du bloc secteur.*)

### ➤ Mise en service de la machine.

**P**rotocole établi en supposant que la machine a été correctement remise. Éventuellement *vérifier la conformité de configuration.*

- Brancher l'alimentation du bloc secteur et promptement vérifier l'établissement du **5.0v** sur le module régulateur. Si la tension affichée n'est pas correcte **débrancher immédiatement le secteur** puis dépanner.
- La LED doit éclairer en bleu et l'écran afficher le **Menu de base**.
- Inverseur Multimètre placé sur position **Machine**. (*Courant nul.*)
- L'indicateur du régulateur sur la mesure en entrée doit indiquer **12.3v**.
- Passer le contacteur de SÉCURITÉ en position centrale pour établir l'énergie 12V sur la motorisation. **La tension sur l'entrée du régulateur ne doit pratiquement pas changer**, le galvanomètre accuse  $\approx 12\text{mA}$ .
- Dans **SYSTEME** activer l'item **Dégagement plateau**. Les moteurs s'activent, l'intensité "titille"  $\approx 200\text{mA}$  puis retombe à  $\approx 12\text{mA}$ .
- Repasser l'inverseur **Machine** sur position **Multimètre**.
- Mesurer à nouveau le **5.0v** sur le régulateur puis éteindre ce dernier.

➤➤➤ **La machine est prête pour pyrograver.**

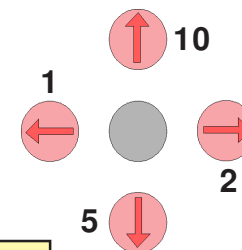
**Épaisseur maximale des objets** à pyrograver : 70mm.

C'est le passage sous le chariot X'X qui limite cette hauteur.

## Option Mode LASER continu.

**D**ébut par une capture de l'**Origine Machine**, puis décale le LASER sur X'X et Y'Y de 60mm. Les moteurs sont ensuite forcés en VEILLE pour minimiser la consommation. Le LASER est à puissance nulle en entrée dans cette fonction. Il est de surcroît possible de décaler différemment manuellement si on le désire.



- La LED rouge du répéteur LASER s'allume au maximum en ouverture de cette option, et s'éteint quand on la quitte.
- Le B.P. central du codeur rotatif termine cette option, force le retour à l'**Origine Machine** suivi de la **Configuration Machine Dégagée**.
- En sortie de cette procédure il y a retour au menu **SYSTEME**. Les moteurs sont replacés en mode VEILLE.
- Durant cette option, la rotation du codeur incrémental augmente ou diminue la puissance du LASER avec un effet de butées logicielles dans la fourchette de valeurs [0% à 100%].
- Pendant cette phase les touches rouges du clavier modifient l'amplitude de chaque **Incrément** ou **décément** avec la valeur indiquées ci-contre.
- En standard la touche noire permet d'éteindre l'écran OLED ou de le rallumer.



En retour à **SYSTEME**, le **Pourcentage** est recopié dans **Puissance\_MAX\_sur\_le\_LASER**.

## Le menu de base.

**D'**un comportement globalement conforme à celui des autres menus, il présente une spécificité dans l'utilisation du clavier dont seulement deux touches ont une action. Les trois boutons rouges **Y+ ↑**, **X- ←** et **Y- ↓** sont sans effet et génèrent un BIP sonore.

- En standard le codeur rotatif  modifie l'option indexée.
- Le bouton noir du CLAVIER allume ou éteint l'écran OLED.
- **Comme la touche X+ → pointe à droite, un peu comme l'index du menu actuellement sélectionné, on a tendance parfois à cliquer dessus pour activer l'item. Aussi, dans le menu racine du logiciel d'exploitation cette touche présente le même effet que celui de .**
- Normalement, en menu de base la LED tricolore doit éclairer en bleu confirmant que les moteurs sont en VEILLE.

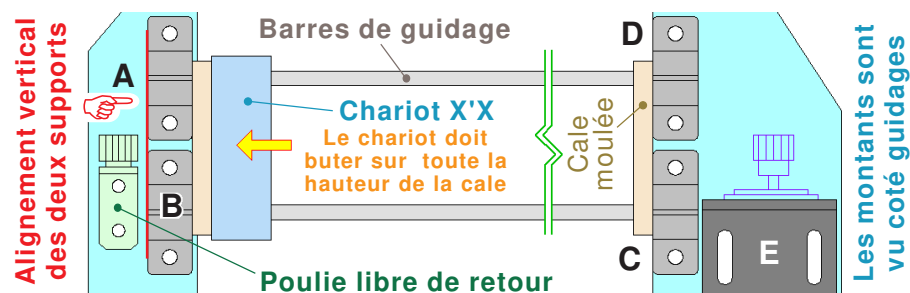
## Protocole d'alignement du guidage X'X.

En préalable, le chariot Y'Y a été assemblé sur le statif de la machine et ses équerres transversales sont assemblées de façon définitive.

**Pour les diverses étapes qui suivent on serre modérément les boulons uniquement pour figer momentanément la géométrie acquise.**

Dans ce qui suit on place en premier les deux montants verticaux.

- 1) Mettre en place sur le montant vertical de gauche l'équerre **E** (Et sa cale d'épaisseur.) qui supporte le moteur pas à pas et la centrer sur les trous de liaison. Assembler les deux montants avec les guidages à billes en serrant à peine les boulons. Les tiges de guidage doivent être positionnées parfaitement à 90° par rapport aux montants en s'aidant d'une équerre de grandes dimensions. Simultanément les montants sont orientés bien verticaux par rapport au statif de la pyrograveuse.



- 2) Immobiliser les tiges de guidage sur les supports **A** et **B** de façon à ce que leurs extrémités affleurent la face extérieure du support.
- 3) Décaler le chariot à gauche pour imposer l'écart correct entre les deux barres de guidage, aligner les supports **A** et **B** **verticalement**.
- 4) Libérer le pincement entre les tiges et les supports **C** et **D** avec un minimum de jeu, mais suffisant pour pouvoir déplacer axialement.
- 5) Décaler contre la butée le chariot à droite pour qu'il impose l'écartement correct entre les deux tiges de guidage et immobiliser en premier le support **C** en ménageant un léger écart avec **E**. Le pincement du support **C** est également immobilisé définitivement.
- 6) Immobiliser le support **D** en le positionnant latéralement pour que le chariot X'X calé à droite il porte sur toute la hauteur de la cale.

Compte tenu de toutes les imprécisions cumulées sur l'ensemble des pièces de la machine, la marge d'ajustement en position est

- 8) Amener le chariot complètement en butée vers l'avant de la machine.
- 9) Le chariot assurant l'écartement transversal idoine entre les deux tiges de guidage, déplacer axialement les supports de telle sorte que le **contact en butée physique soit simultané sur A et B**.

**NOTE :** Seul le jeu entre les vis  $\phi$  M4 et les trous de passage dans la plaque du statif laissent une possibilité d'alignement. De plus, ce jeu est encore restreint par le passage à travers les trous des cales d'épaisseur dont les positions sont approximatives. Toutes les imprécisions cumulées sur la machine rendent impossible le contact simultané. C'est la raison pour laquelle, en **A** une butée plus longue de 3 mm que celle placée en **D** a été réalisée après mesurage, solution bien plus simple que d'avoir à allonger les trous de traversées. Par ailleurs, il importe de savoir que la géométrie des supports est elle même imparfaite, et en particulier le parallélisme entre l'alésage pour les tiges et la surface portante de la semelle. Pour ne pas forcer sur l'ensemble, que les spécialistes accuseraient d'hyperstaticité, la cale située sous **B** a été moulée en 3D plus épaisse d'un millimètre.

- 10) Serrer les pincements et les boulons de liaison des supports. Vérifier le déplacement sans aucun problème du chariot Y'Y sur toute la course **en exerçant l'effort sur la traverse à l'endroit où sera immobilisée la courroie**. (L'effort appliqué proche des tiges de guidage diminue le risque d'arc-boutement et fausse le test.)
- 11) Si la translation n'est pas parfaitement libre sur toute la course de butée mécanique à butée mécanique, reprendre entièrement la procédure depuis l'item n°3.
- 12) Lorsque le déplacement est totalement libre, sans "points durs", serrer définitivement l'ensemble des vis et des boulons du chariot Y'Y et des deux guidages sur le statif. Les boulons qui assurent la liaison des supports sont retournés pour des raisons esthétiques.

**IMPORTANT :** L'arc-boutement explicité dans le didacticiel peut s'avérer délicat à résoudre. Heureusement que quatre guidages complets ont été approvisionnés multipliant les éléments disponibles. On peut donc optimiser la sélection des éléments. Ce sont les **deux douilles à billes coté gauche** qui augmentent la longueur de guidage et suppriment le problème. Les **appairer avec une tige pour avoir le moins de jeu possible**. Côté droit de la machine, au contraire **choisir une tige et une douille présentant un maximum de "rotulage"**.

## Protocole d'alignement du guidage Y'Y.

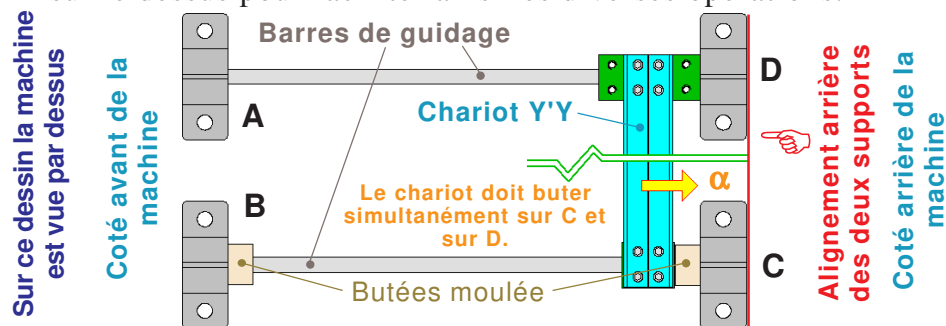
**G**lobalement, pour toutes les étapes qui vont suivre, sauf à la fin, il ne faudra serrer à peine les boulons pour "figer" les ajustements géométriques. Les pincements des supports sur les tiges seront précontraints le juste ce qu'il faut pour annuler le jeu interne.

1) Assembler les tiges sur les supports avec leurs extrémités au raz des faces extérieures. L'écartement interne doit être identique. Les trois douilles à billes sont insérées ainsi que les douilles de butées sur le guidage de droite. (*Vis de pincement coté extérieur.*)

**ATTENTION :** Avant de poursuivre lire l'encadré en bas du verso.

2) Assembler les deux traverses de la structure du chariot. Pour aligner ces dernières, réunir provisoirement les deux montants bien verticaux, les deux renforts ainsi que les deux guidages pour le chariot X'X. Soigner la géométrie. (*Guidages X'X horizontaux ...*)

3) Poser l'ensemble sur le statif sans oublier de placer les cales sous les supports. Introduire les huit boulons d'assemblage des guidages en translation rectiligne en plaçant provisoirement leurs écrous sur le dessus pour faciliter ainsi les diverses opérations.



4) Pousser comme montré par la flèche  $\alpha$  le chariot entièrement vers l'arrière. Les huit vis de liaison des traverses du chariot sont libérées. Placer les supports **C** et **D** bien alignés sur le bord arrière du statif et les immobiliser. (*Toujours le juste nécessaire.*)

5) Placer le chariot bien en contact sur le support simultanément en **D** et sur la butée en **C**. Bloquer les quatre vis de la traverse arrière.

6) Aligner correctement la traverse avant et l'immobiliser à son tour. (*La centrer latéralement sur la machine en contact avec l'autre traverse.*)

7) Serrer normalement le pincement des deux supports **C** et **D** et libérer celui des supports **A** et **B**.

... / ...

insuffisante pour **D**. Comme allonger le trou de passage du boulon impose de tout redémonter, il est plus simple d'intercaler une rondelle d'épaisseur comme montré sur Image1.JPG qui se trouve dans le dossier <Assemblage mécanique> de <Galerie d'images>.

7) Serrer à son tour le pincement sur le support **D**.

8) Quand le chariot X'X se déplace librement sur toute la course de butée en butée, serrer normalement tous les boulons de l'assemblage.

## Protocole de mise en place de la courroie crantée.

**C**ette description est décrite pour le charriot Y'Y, en notant que pour X'X la procédure sera totalement analogue. Noter que par construction, (*Usage des cales d'épaisseur adaptées en dessins d'études mécaniques.*) les deux courroies se trouvent exactement contre les structures des chariots quand elles sont tendues sur les poulies.

1) Chariot placé au centre et la poulie libre poussée tout à l'arrière.

2) Équerre équipée du moteur entièrement décalée vers l'avant.

3) Comme montré sur Image2.JPG immobiliser la courroie mais dans cette phase que d'un seul coté sur la bride crantée.

4) Moteur et poulie libre rapprochés au maximum par les deux premiers items, passer la courroie en boucle et la couper à la bonne longueur, pour être serrée sur un maximum de dents de la bride.

5) Fixer la deuxième extrémité sur la bride avec du ruban adhésif.

6) Placer définitivement la courroie en boucle sur les deux poulies et l'immobiliser sur les traverses de structure du chariot : **Serrer très modérément la bride** ou risque de cisaillement de la courroie. Freiner le boulon avec du vernis à ongles.

7) Avancer au maximum la poulie libre vers l'avant de la machine et bloquer son support bien parallèle au profilé du statif.

8) Déplacer le chariot Y'Y tout à l'avant et ajuster la position axiale de la poulie libre par le biais du système Écrou / Contre écrou. À ce stade la courroie est presque tendue si sa longueur a été correctement établie. (*Presque tendue lors de la coupe.*)

9) Ramener le chariot complètement en butée arrière. Puis, tout en ajustant la position axiale de la poulie motrice pour aligner ses flancs avec la courroie, tirer vers l'arrière pour finir de tendre (*Sans exagérer !*) la courroie et bloquer la position.

**Raire circuler manuellement le chariot en agissant sur la poulie motrice, et vérifier que sur toute la course possible strictement rien ne vienne toucher la structure du chariot ou la courroie.**




## Protocole de validation du matériel. 1/8

**P**réambule de sécurité pour le matériel, le PUPITRE est entièrement débranché, la machine est placée dans une configuration pour laquelle **les deux capteurs d'Origine Machine ne sont pas masqués**. *Mise à part pour vérifier le LASER, l'ensemble des manipulations sera effectué sans la présence du plateau de la machine.*

- 1A) Contacteur de puissance verrouillé sur SÉCURITÉ ÉNERGIE.
- 1B) Mettre en place le "strap" qui remplace le relais du LASER.
- 2) Débrancher temporairement l'alimentation sur la carte Arduino NANO ainsi que sur l'alimentation en 5Vcc du MULTIPLEXEUR.
- 3) Mettre sous tension 220V~ le module secteur 12v 5A :
  - La LED bleue de présence 12V s'allume sur le circuit des shunts,
  - La LED bleue de "sortie avec coupure de sécurité" est éteinte,
  - Activer l'affichage sur le module de régulation 5Vcc et sélectionner la mesure de tension sur **IN**. La valeur affichée doit se situer entre environ **12.1Vcc** et **12.3Vcc**.
- 4) Sur le module de régulation sélectionner l'affichage sur **OUT** et ajuster finement le potentiomètre multi-tours pour avoir **5.0Vcc**.
- 5) Couper l'alimentation secteur 220V~ et brancher le +5Vcc sur la carte NANO. Mettre sous tension. La valeur de **5.0Vcc** ne doit pas changer sur la sortie du régulateur et un BIT d'alerte est généré.

### ➤ Validation du câblage du PUPITRE.


- 6) Brancher le tableau des LEDs sur le MULTIPLEXEUR, puis, fiche **Protocole d'intégration du PUPITRE** en main, brancher intégralement le PUPITRE mais pas de carte **SD** dans le lecteur.
- 7) Mettre sous tension. **Si le 5.0Vcc** sur le module régulateur **ne s'établit pas correctement, couper immédiatement l'énergie** et dépanner. S'il n'y a pas d'erreur grave de câblage :
  - Un BIP d'alerte, OLED indique un problème de lecture **SD**,
  - La LED tricolore bleue s'allume, et la LED verte du clavier clignote.
- 8) Insérer une carte **SD** dans le lecteur et provoquer un RESET.
  - La LED jaune s'allume durant la lecture de la carte **SD**,
  - L'afficheur OLED affiche le **Menu de base**.
- 9) Cliquer sur le bouton noir, l'écran s'éteint, cliquer une deuxième fois pour rétablir la page-écran initiale.
- 10) Tourner le codeur rotatif dans les deux sens, l'index pointant les items doit se déplacer correctement en fonction du sens de .

## Protocole de validation du matériel. 3/8

### ➤ Validation des capteurs d'Origine Machine.

- 24) Contacteur de puissance verrouillé sur SÉCURITÉ ÉNERGIE. Commencer par amener à la main en agissant sur les poulies motrices les deux axes en butée mécanique à l'arrière et à gauche.
- 25) Effectuer un RESET. La LED triple s'allume en violet.
- 26) Dans le **Menu de base** indexer puis activer le menu **SYSTEME**.  
Invoquer la fonction **U sur capteurs ORG** : Les valeurs retournées par les deux CAN doivent être élevées et dépasser notablement 950. L'afficheur indique **Origine X : OUI** et **Origine Y : OUI**.
- 27) Manuellement décaler le chariot pour dégager le capteur X'X : Dès que le "couteau" dégage la cellule photorésistante, la valeur retournée par le CAN est inférieure à 40 et **Origine X : NON**.
- 28) À la main déplacer le chariot pour dégager le capteur Y'Y : Dès que le masque dégage la cellule photorésistante, la valeur retournée par le CAN est inférieure à 40 et OLED affiche **Origine Y : NON**.

**NOTE :** Si les directives proposées sur la fiche **Câblage des capteurs d'Origine Machine** sont scrupuleusement respectées, sur le prototype, plein ensoleillement dans le local on obtient les valeurs suivantes :  
X'X non masqué : CAN = 16 et masqué CAN = 1007/1008.  
La moyenne choisie pour #define **Seuil\_Origine\_X** est **511**.  
Y'Y non masqué : CAN = 13 et masqué CAN = 1013.  
La moyenne choisie pour #define **Seuil\_Origine\_Y** est **513**.

- 29) B.P. central du **C.I.**  pour revenir au menu **SYSTEME** puis activer l'item **Test capteurs ORG** : Il y a retour menu **SYSTEME** et surtout la LED triple s'illumine en **vert cyan**.

### ➤ Validation de la motorisation.

- 30) L'énergie de puissance étant coupée, éloigner les deux chariots des supports de guidage en les plaçant à mi-course.
- 31) Bouton SÉCURITÉ paré pour une coupure URGENTE, valider le **Mode MANUEL** et Établir l'énergie de puissance 12Vcc. Immédiatement les deux moteurs se mettent à tourner pour effectuer la capture de l'Origine Machine. Si l'un des deux **moteur tourne dans le mauvais sens, verrouiller immédiatement la SÉCURITÉ ÉNERGIE**. (Car la course n'est pas encore sécurisée par programme.)

## Protocole de validation du matériel. 4/8

Dans le cas d'un matériel approvisionné identique à celui du prototype et d'un câblage sans erreur, les moteurs doivent tourner dans le sens attendu. Si la rotation n'est pas conforme, il est bien plus simple de changer deux instructions dans P20 au lieu de permuter des fils sur le module de pilotage : Dans la procédure `void Faire_un_pas_sur_X()` ou `Faire_un_pas_sur_Y()` permuter **HIGH** et **LOW** sur les deux instructions `digitalWrite(Sens_sur_Y, État)`.

32) Avec la **Fiche Câblage des capteurs d'Origine Machine** réaliser la manipulation ➤ **Protocole de positionnement des capteurs**.

### ➤ Ajustement logiciel des "fins de courses".

Maintenant que le matériel est protégé des collisions avec les supports de guidage vers la gauche et vers l'arrière, il importe de définir les limites de déplacements par des butées virtuelles. Les directives de paramétrage des données stratégiques se trouvent en tête de **P20\_Programme\_exploitation.ino** et sont aisément repérables car terminées par des remarques de type `@@@@@@@@@@@@@`.

33) Imposer au préalable des paramètres "exagérés" pour que le programme permette provisoirement de dépasser les capacités de la machine :

- `#define Course_MAX_sur_X 235.0`
- `#define Course_MAX_sur_Y 215.0`
- Dans `void Capturer_Origine_machine()` modifier :  
`Ancien_Y = 2150`; ainsi que `Position_Y = 215.0`;

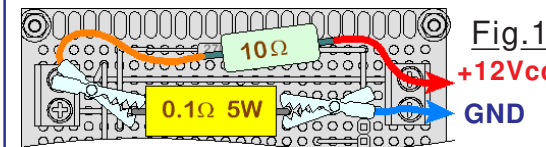
34) L'énergie en puissance est rétablie, et le **bouton d'URGENCE** non verrouillé mais **libre et disponible** en cas d'aléas. Dans le menu **SYSTEME**, activer **Dégagement plateau**. Puis, passer en **Mode MANUEL**, la machine capture les origines et demande de **Focaliser le LASER**. Cliquer sur un B.P. pour continuer.

L'affichage précise **X : 0.0** qui représente l'ordonnée de la position image. Position **Y : 215.0** correspond à la limite logique Y'Y maximale définie actuellement pour le débattement possible. La course sur les deux axes doit être ajusté en fonction de la réalité.

35) L'écran précise que **Chaque B.P.** déplacera le chariot concerné de **10.0mm**. Frapper plusieurs fois sur **X+ ➡** pour approcher la butée physique. Puis à environ 15mm à 20mm tourner le codeur rotatif ⚙ dans le sens antihoraire pour passer en **0.5mm**.

## Protocole de validation du matériel. 2/8

- 11) Par exemple indexer **FICHIER** et cliquer sur **X+ ➡** qui génère un texte fugitif puis liste les fichiers valides de la carte **SD**.
- 12) Cliquer sur **X- ⬅** qui provoque une nouvelle lecture de la carte **SD**.
- 13) Pointer un fichier image valide puis appuyer sur la touche **Y- ⬇** qui doit faire afficher les caractéristiques de l'image indexée.
- 14) Activer deux fois de suite sur **X+ ➡** pour faire apparaître le texte fugitif **Mode Pas à Pas : OUI** puis **NON**.
- 15) Actionner le bouton central du codeur rotatif ⬇ qui doit ramener le logiciel et l'affichage sur le **Menu de base**.
- 16) Tester les quatre positions du sélecteur de calibre et vérifier que les LEDs 3mm du répéteur indiquent correctement le positionnement.
- 17) Passer l'inverseur **Machine** sur la position **Multimètre**. Par des sources diverses de tension externes, vérifier les quatre calibres possibles et la précision des mesures effectuées.
- 18) Inverseur **LASER** sur **Coupé**. Basculer l'inverseur **Multimètre** sur la position **Machine**. Sur le sélecteur choisir la position relative au calibre **U : 20V**. Le galvanomètre reste à zéro.
- 19) Contacteur de puissance **SÉCURITÉ ÉNERGIE** sur "libre".
  - La tension indiquée sur le galvanomètre est d'environ 12,2V.
  - Les deux LED bleues sur le circuit des shunts sont allumées.
- 20) Placer le sélecteur sur la position **LASER : 2A**. Conformément



au schéma de la Fig.1 ponter le point "chaud" du shunt de **0,1Ω** en intercalant une résistance externe de **10Ω**

- au +12Vcc interne, ou en ajoutant une liaison avec **GND** sur une alimentation externe de laboratoire ajustée à 12V. Le galvanomètre de mesure doit indiquer une intensité de 1,2A.
- 21) Effectuer une manipulation strictement analogue sur les deux positions **X : 500mA** et **Y : 500mA** mais remplacer la résistance de **10Ω** par un **composant de 50Ω**. Sur les deux positions la mesure doit indiquer un courant de 250mA.
  - 22) Repasser l'inverseur **Machine** sur **Multimètre**.
  - 23) Basculer l'inverseur sur **Éclairage**, bien que **LASER** soit sur **Coupé** il doit fonctionner normalement.

À ce stade, on a intégralement vérifié le câblage du **PUPITRE**.

## Protocole de validation du matériel. 5/8

- 36) En intercalant une bande de carton un peu épais entre le chariot X'X et la cale moulée, actionner **X+ ➡** jusqu'à détecter le contact matériel. L'écran OLED indique alors le déplacement effectué.
- 37) Par exemple il affiche : **Position X : 230.0** juste au moment du contact physique entre le chariot et la cale moulée. Par mesure de sécurité, nous allons "arbitrairement" limiter le déplacement en laissant une marge de 2mm. La course maximale sur l'axe X'X surveillée par le programme sera donc de 228,0mm.
- 38) Imposer au logiciel la nouvelle limite :
- #define **Course\_MAX\_sur\_X 228.0**
- 39) Rotation de **↻** dans le sens horaire pour passer en **10.0mm**. Frapper plusieurs fois sur **Y- ⬇** pour approcher la butée physique et comme pour X'X repasser en **0.5mm** bien avant le contact avec le témoin en carton. Quand la butée matérielle est effective, l'afficheur indique une valeur du genre **Position Y : 11.0** par exemple.
- 40) Par mesure de sécurité, nous allons "arbitrairement" restreindre le déplacement en ménageant un écart de 2mm. La course maximale sur l'axe Y'Y surveillée par le programme sera donc de :
- $$215.0 - (11.0 + 2) = 202\text{mm.}$$
- Soit les modifications dans P20 :
- #define **Course\_MAX\_sur\_Y 202.0**
  - Dans **void Capturer\_Origine\_machine()** modifier :  
**Ancien\_Y = 2020**; ainsi que **Position\_Y = 202.0**;

À partir d'ici, quand un fichier présentera une **Origine Image** de zéro sur X et de zéro sur Y la capture de l'**Origine Image** amènera le chariot X à 2mm de la cale moulée située à gauche et à 2mm de la cale moulée du support de guidage situé à l'avant.

### ➤ Validation de la précision des mouvements.

- Cette phase de la validation de la motorisation a pour but de s'assurer que les déplacements respecteront les dimensions des images et ce dans les deux directions de mouvements. On va pour cette phase installer une règle graduée pour effectuer les mesures.
- 41) Conformément à la photographies **Image13.JPG** réaliser la règle de mesurage dont le dessin est fourni avec les plans. Disposant de support transparent elle a été imprimée sur film translucide pour faciliter l'opération, mais ce n'est absolument pas impératif.

## Protocole de validation du matériel. 7/8

### ➤ Validation des circuits électroniques du LASER.

- 48) Inverseur **LASER** sur **Coupé**, passer l'inverseur **Multimètre** sur la position **Machine** et le sélecteur pour la mesure **LASER : 2A**.
- 49) Débrancher le connecteur rouge au sommet de l'électronique de pilotage. **Le plateau de la machine est remonté et à sa place.**
- 50) Allumer l'éclairage qui permet de vérifier que les connecteurs HE14 de couleur verte n'ont pas été branchés dans le mauvais sens.
- 51) Vérifier que sur le connecteur femelle rouge on a 0V alimentation.
- 52) La puissance 12V est libérée sur le bouton **SÉCURITÉ**. On n'a toujours pas d'énergie sur le connecteur rouge.
- 53A) Inverseur **Coupé** basculé sur **LASER**. Le 12V est alors présent sur la fiche rouge **et la polarité est correcte.** (Le + sur le fil rouge.)
- 54B) Enlever le "strap" qui remplace le contact du Relais LASER. Le 12V est alors à nouveau coupé, la LED du Relais est éteinte.
- 54) Inverseur **LASER** sur **Coupé**, rebrancher le connecteur rouge sur l'électronique de pilotage et **chausser les lunettes de protection..**
- 55) Couper l'éclairage et valider l'item **Mode LASER continu** du menu **SYSTEME**. La LED rouge du Relais LASER s'illumine. La machine capture les origines puis décale la position de 60mm sur X et sur Y. Affichage **LASER 0 pour cent**.
- 56) Inverseur **Coupé** basculé sur **LASER**. Les deux ventilateurs s'activent et le galvanomètre indique une consommation de 200mA.
- 57) Tourner **↻** pour obtenir **LASER 1 pour cent** avec une tâche visible mais sans plus sur le plateau. Focaliser la lentille pour en minimiser la taille. (Dans la pratique c'est un petit trait lumineux d'environ 3 mm de long légèrement en biais sur le prototype.)
- 58) Consulter **Image10.JPG** à **Image12.JPG** disponibles dans le dossier **<Les circuits électroniques>** en complément d'informations.

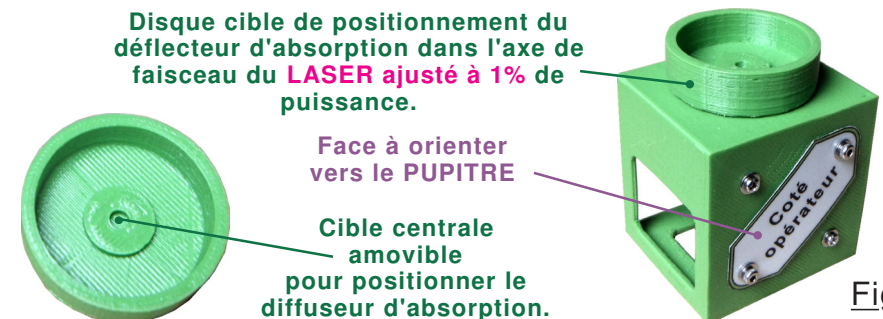

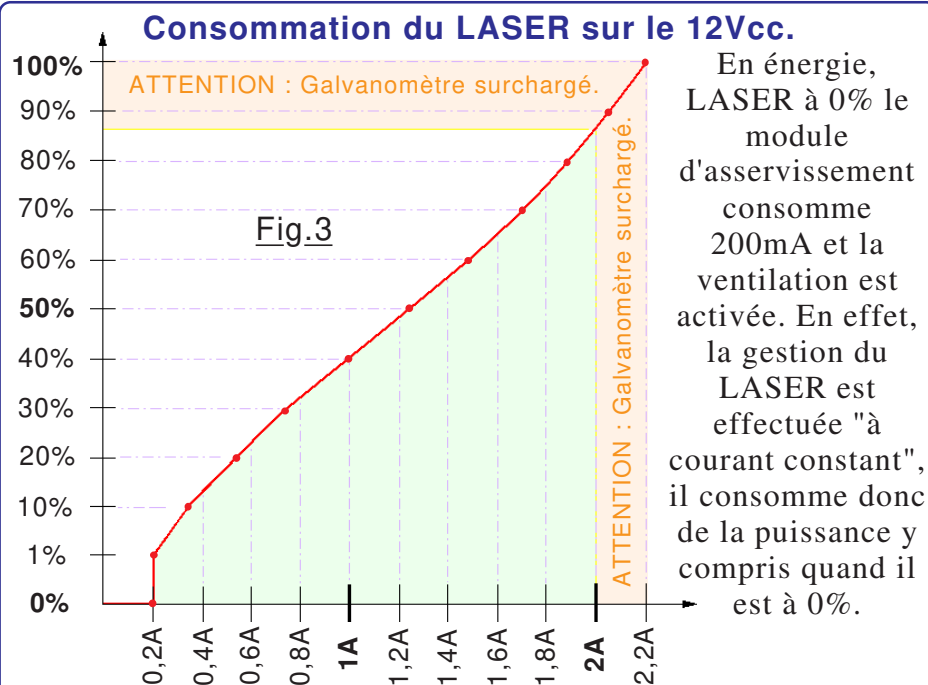


Fig.2



## Protocole de validation du matériel. 8/8

- 59) Mettre en place le **DÉFLECTEUR D'ABSORPTION** et centrer le spot du LASER sur sa **cible** de positionnement. Veiller à ce que la face équipée de l'étiquette soit bien orientée vers le PUPITRE.
- 60) Sans bouger le **DÉFLECTEUR** enlever la **cible** circulaire puis augmenter la puissance à **10 pour cent**. L'ampèremètre indique environ 350mA. Pousser la puissance à **40 %** qui doit exiger 1A de courant accompagné d'une baisse de régime sur le ventilateur. À **86 %** le galvanomètre indique exactement **2A**.
- 61) Repasser le sélecteur pour mesurer **X : 500mA** avant de dépasser les 86%, puis remettre l'inverseur **Machine** sur la position **Multimètre**.
- 62) Faire varier la puissance optique jusqu'à **100 %** et vérifier que l'évolution est conforme à celle de la Fig.3 de l'encadré présenté en bas de cette fiche. Maintenir **100 pour cent** entre 10 minutes et 20 minutes : Le corps du LASER doit rester à peine tiède. Redescendre à **0 %** puis remettre **LASER** sur **Coupé**.
- 63) Les circuits de gestion du LASER sont entièrement validés. Actionner le B.P. central du **C.I.**  pour terminer le protocole. La LED rouge du Relais du LASER doit s'éteindre.



## Protocole de validation du matériel. 6/8

- 42) En s'inspirant d'**Image14.JPG** un trait de scie a été pratiqué sur le support de guidage arrière gauche. La règle sera immobilisée sur la douille de guidage à billes et supportée comme montré sur **Image15.JPG** par une longue "poutre" décalée et suffisamment fine pour ne pas interférer avec le support de guidage.
- 43) Les trous oblongs visibles sur **Image13.JPG** permettent de situer le zéro machine exactement au dessus du repère rouge lorsque la machine est en position **Origine Machine**. La règle est conçue pour que les graduations proches de l'opérateur indiquent les **Coordonnées Image**, (*Position du chariot Y'Y*) et sur celle les plus éloignées le déplacement effectué depuis l'**Origine Machine**.
- 44) Activer le **Mode MANUEL**, puis avec **Y+ ↑** déplacer le chariot Y'Y en **Position Y : 0.0** et vérifier que l'origine zéro de la règle est bien au dessus du repère rouge scié sur le support de guidage.
- 45) Activer le **Mode MANUEL**, puis avec **Y- ↓** provoquer de grands déplacements et comparer les coordonnées affichées sur l'écran OLED avec la position réelle observée sur la règle de mesure.

*Sur le prototype, pour un déplacement de 200mm l'écart entre le mouvement réel et celui affiché est inférieur de 0,5 millimètre ce qui prouve la précision étonnante de réalisation des poulies et de la courroie crantée. Cette "imprécision" dérisoire ne sera pas appréciable sur les images pyrogravées.*

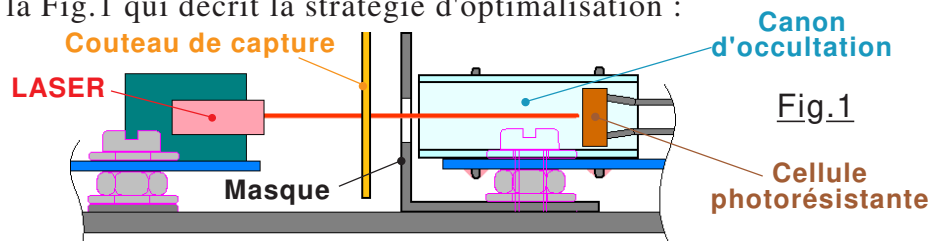
- 46) Pour valider la motorisation sur X'X on se contentera d'une vérification visuelle en effectuant des déplacements et en comparant l'information de position sur l'écran OLED avec la réalité. Un simple réglet d'atelier servira à évaluer la position en tenant compte des 2mm de sécurité entre origine X'X et support.

**NOTE : Étant donné que la motorisation sur X'X est identique à celle pour Y'Y, on retrouve forcément la même précision.**

- 47) Le dernier test de validation consiste à faire fonctionner la machine un peu dans toutes les directions et vérifier qu'il n'y a pas de perte de position par des pas qui ne seraient pas effectués. (*Par exemple utiliser les "images" disponibles dans le dossier <Fichier pour la carte SD\Fichier de test machine>.*) L'expérience montre que les moteurs étant pourtant pilotés avec le courant minimal, pour arriver à les perturber il faut les freiner manuellement de façon notable sur les poulies motrices quand ils sont en fonctionnement.

## Câblage des capteurs d'Origine Machine.

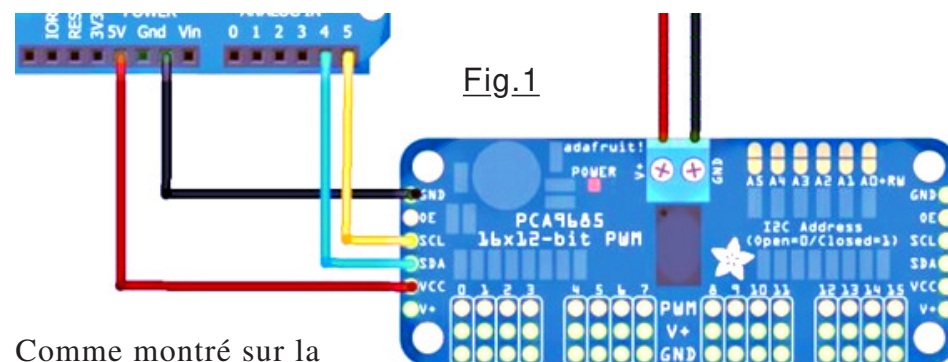
S oigner la réalisation des capteurs d'origine sera un garant de fiabilité. Fonctionnant avec la lumière émise par un LASER, il faut minimiser l'influence parasite générée par la lumière ambiante. Leur orientation sur la machine les place à contre-jour. Considérons la Fig.1 qui décrit la stratégie d'optimisation :



La **Cellule photorésistante** est "enfermée" dans le **Canon d'occultation**, formé d'un tube un peu long par lequel la lumière ne peut pénétrer que dans l'axe. Immobilisé avec le circuit imprimé qui supporte cette cellule, on place un **Masque** qui réduit encore la pénétration possible de lumière ambiante. Le **Couteau de capture** immobilisé sur les chariots passera relativement proche du **Masque** pour limiter les effets de diffraction. La réalisation pratique des capteurs est montrée sur plusieurs photographies dans le dossier **<Les circuits électroniques>** du répertoire **<Galerie d'images>**. Sur **Image1.JPG** on constate que les cellules photorésistantes sont placées préalablement dans un fourreau isolant constitué de deux gaines thermorétractables. Ce dernier centre la cellule dans le **Canon** et obture la lumière "arrière". Sur **Image2.JPG** prise en "macro" on voit dépasser la gaine rouge alors qu'**Image3.JPG** est saisie dans l'axe par l'ouverture où pénétrera le faisceau LASER. Elle montre clairement comment les **Canon**s sont immobilisés sur le petit circuit imprimé. Notez que sur **Image4.JPG** le capteur **X'X est à droite** alors que celui d'**Y'Y est à gauche**. L'**Image5.JPG** a été saisie au tout début du développement logiciel quand la machine n'existait qu'en binaire dans la mémoire de l'ordinateur. Bien que pour le capteur Y'Y situé au fond, le **Masque** ne soit pas du tout aligné sur son LASER, le système fonctionnait correctement ce qui prouve la large tolérance opérationnelle. Sur la machine, le positionnement des quatre petits circuits est précisé sur les plans fournis. Les trous de passage des vis d'immobilisation  $\phi$  M2 présentent un **jeu suffisant pour assurer un ajustement optimal de la position relatives** des divers éléments.

## Branchements de la carte PCA9685.

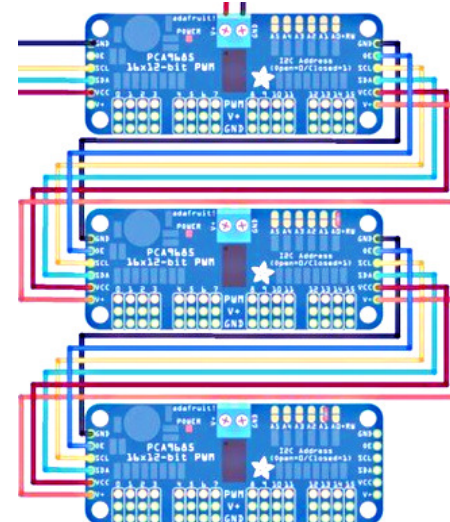
S ituée au verso de cette fiche la Fig.2 présente les liaisons électriques à établir entre la carte Arduino NANO et le module multiplexeur. Comme les servomoteurs provoquent des appels de courants "virulents" sur la ligne de puissance, un condensateur d'environ 100 $\mu$ F par servomoteur piloté doit être placé au plus proche du bornier "de puissance". (470 $\mu$ F pour 5 servomoteurs.)



Comme montré sur la Fig.1, si un seul multiplexeur est utilisé ne rien brancher sur **OE**.

### ➤ Adressage du module.

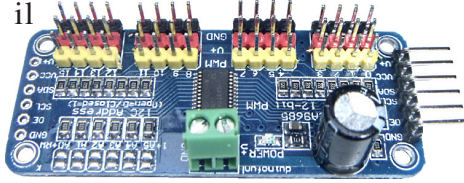
La bibliothèque impose pour la ligne I2C le signal **SDA** sur **A4** et le signal **SLC** sur **A5**. L'adressage par défaut sur le matériel est en 0x40. On peut à la demande, ce qui sera obligatoire si on chaîne plusieurs modules, (On peut en chaîner jusqu'à 64 !) modifier l'adresse matériellement par établissement de contacts sur des ponts disponibles sur le circuit imprimé dans la zone **RW**.



- RW 0** : Address = 0x40 Adrs binaire 000000 (Pas de pont)
- RW 1** : Address = 0x41 Adrs binaire 000001 (Pont sur A0)
- RW 2** : Address = 0x42 Adrs binaire 000010 (Pont sur A1)
- RW 3** : Address = 0x43 Adrs binaire 000011 (Sur A0 et sur A1)
- RW 4** : Address = 0x44 Adrs binaire 000100 etc. (Pont sur A2)

## Utilisation du multiplexeur de servomoteurs.

Fondamentalement ce circuit peut piloter en PWM n'importe quoi avec pour limite que toutes les sorties seront à une fréquence identique. Piloté par la ligne I2C il ne monopolise que les deux broches **A4** et **A5**, et peut les partager avec d'autres modules puisque par "matériel" on peut modifier à convenance son adresse I2C. (Par défaut en 0x40.) Comme le circuit dialogue sur la ligne I2C il faut commencer par `#include <Wire.h>` qui gère la ligne bidirectionnelle. Ensuite on doit déclarer la bibliothèque `<Adafruit_PWMServoDriver.h>` dans l'**IDE** qui gère ce module par des fonctions faciles à utiliser.



### ➤ Programmation basique.

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();
void setup() {
  pwm.begin(); // Initialise le module multiplexeur
  pwm.setPWMFreq(NM); // Fréquence de la PWM.
                      NM compris entre 24 et 1526.
  void loop() {
    pwm.setPWM(Num_sortie, 0, RCY);
```

- **Num\_sortie** sera compris entre [0 et 15].
- **RCY** définit le rapport cyclique compris entre [0 et 4095].

### ➤ Méthodes de Adafruit\_PWMServoDriver.h.

Spécifique au multiplexeur PCA9685 à 16 canaux cette bibliothèque permet facilement de gérer chaque moteur indépendamment l'un de l'autre. L'effet d'une commande reste effectif jusqu'à une nouvelle consigne pour le canal concerné.

#### Fréquence du signal généré sur les 16 canaux.

```
pwm.setPWMFreq(Fréquence);
```

Cette instruction doit être placée dans `void setup()` et définit la fréquence des signaux PWM générés sur les sorties. La valeur sera comprise entre 24 et 1526, mais en standard on adopte généralement une fréquence de 50Hz si le module pilote des servomoteurs classiques.

## ➤ Protocole de positionnement des capteurs.

1) Placer en position "moyenne" les deux éléments à aligner. La photographie **Image6.JPG** présente l'opération en cours, avec sur l'**Image7.JPG** les informations sur la géométrie à assurer. Il faut simultanément satisfaire :

- La coaxialité entre le LASER, le Masque et le Canon,
  - Outre la coaxialité le faisceau doit être centré sur le trou du Masque,
  - La perpendicularité entre l'axe optique et le déplacement du chariot,
  - Placer axialement le total pour que le Masque soit proche du Couteau.
- 2) Amener le chariot à la main en butée arrière et vérifier que le couteau occulte entièrement l'orifice du Masque à  $\approx 5\text{mm}$  de la butée mécanique. Déplacer à nouveau le chariot vers le centre.
- 3) Cacher le capteur X'X pour faire croire que l'origine est capturée.

**ATTENTION :** Les prochaines manipulations font suite à l'item n°31 de la fiche **Protocole de validation du matériel**. 4/8 et en **constituent l'item n°32**. Les moteurs tournent donc dans le sens correct pour effectuer la capture des **Origines Machine**.

Normalement si la cotation des dessins a été respectée, le moteur s'arrête à environ 7mm du contact physique. (Large marge prévue.)

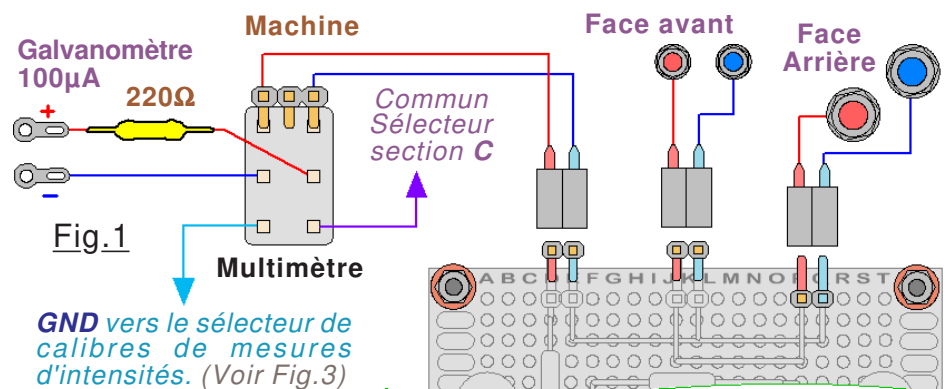
- 4) Sur **Image8.JPG** le couteau est réalisé aux cotes du dessin. Lorsque l'on diminue la largeur du couteau de 5mm, on obtient le profil définitif visible sur **Image9.JPG** qui **en capture d'origine sur le prototype stoppe le moteur à 2mm de la butée matérielle**.
- 5) Reprendre ces manipulations en modifiant la position du couteau sur le chariot pour obtenir un écart avec le **Masque** d'environ 2mm. La face du couteau doit être parallèle à celle du **Masque**.
- 6) Dans le menu **SYSTEME** activer l'item **U sur capteurs ORG**. Déplacer le chariot lentement à la main et vérifier que la capture confirmée par **Origine Y : OUI** bascule à 2mm du contact réel.
- 7) Pour terminer la validation, passer en **Mode MANUEL**, puis par des déplacements **Y- ↓** et surtout avec **Y+ ↑** provoquer volontairement un débordement vers la butée arrière et **vérifier que le moteur se fige sur la butée virtuelle à 2mm du contact physique**.

➡ Pour positionner le capteur d'Origine Machine X'X procéder de façon strictement analogue mis à part le fait qu'il n'est pas nécessaire de masquer le capteur sur Y'Y devenu effectif.

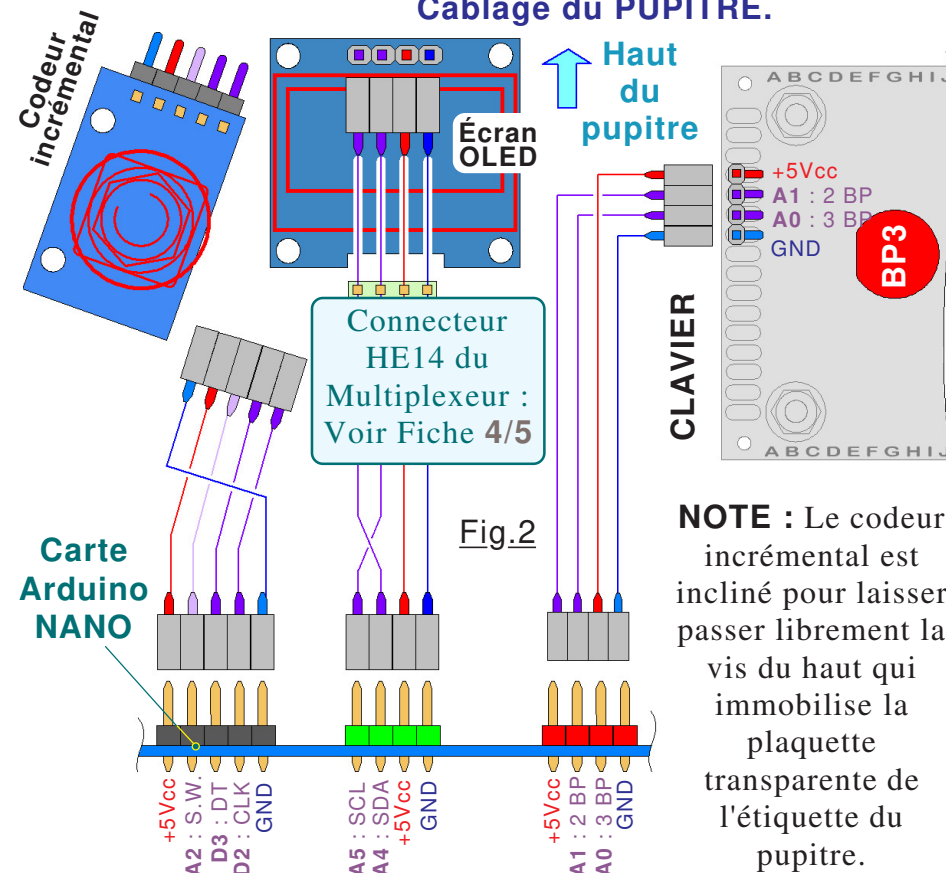


Fiche de câblage des circuits électriques. 1/5

### Circuit imprimé du Multimètre.

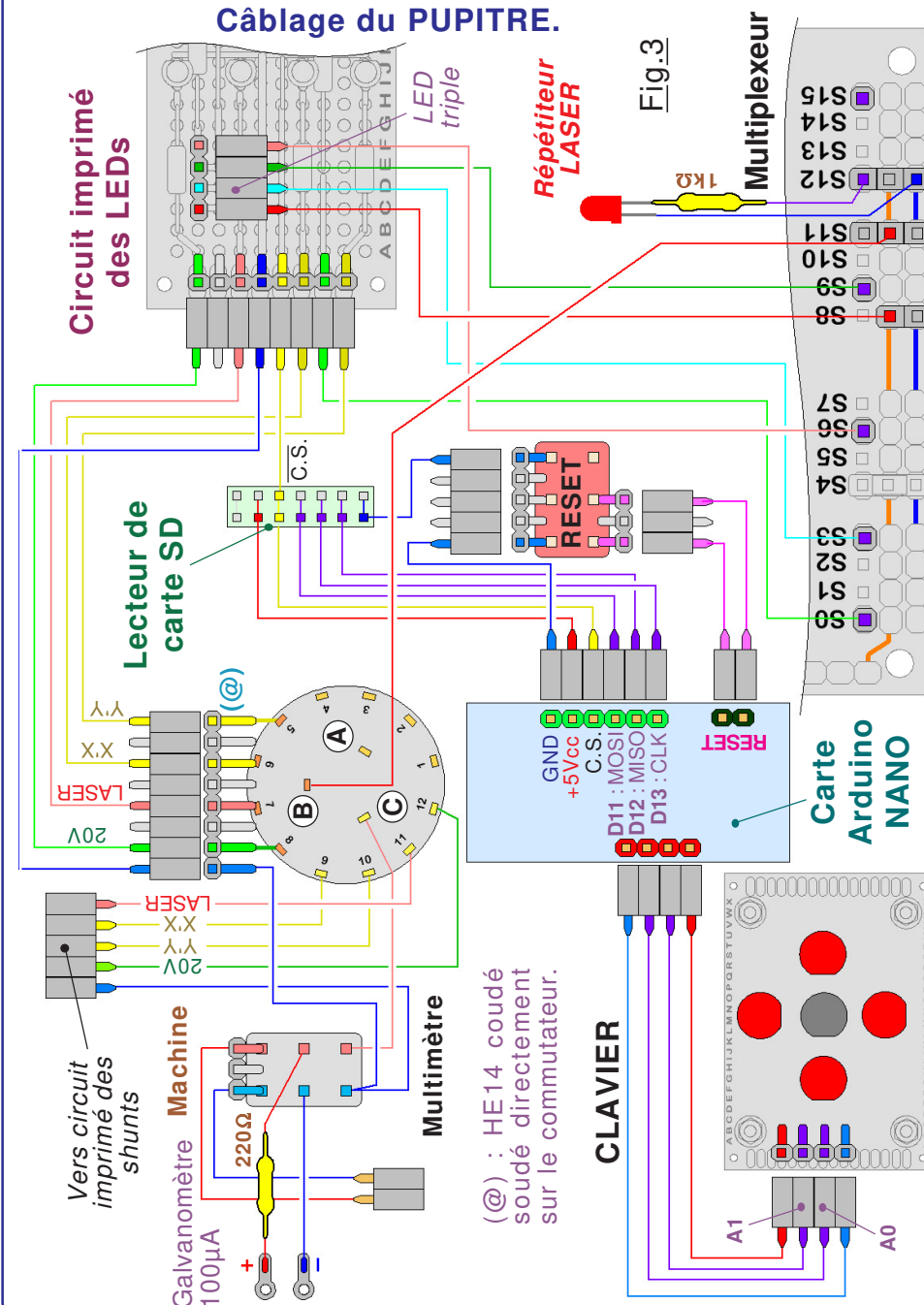


## Câblage du PUPITRE.



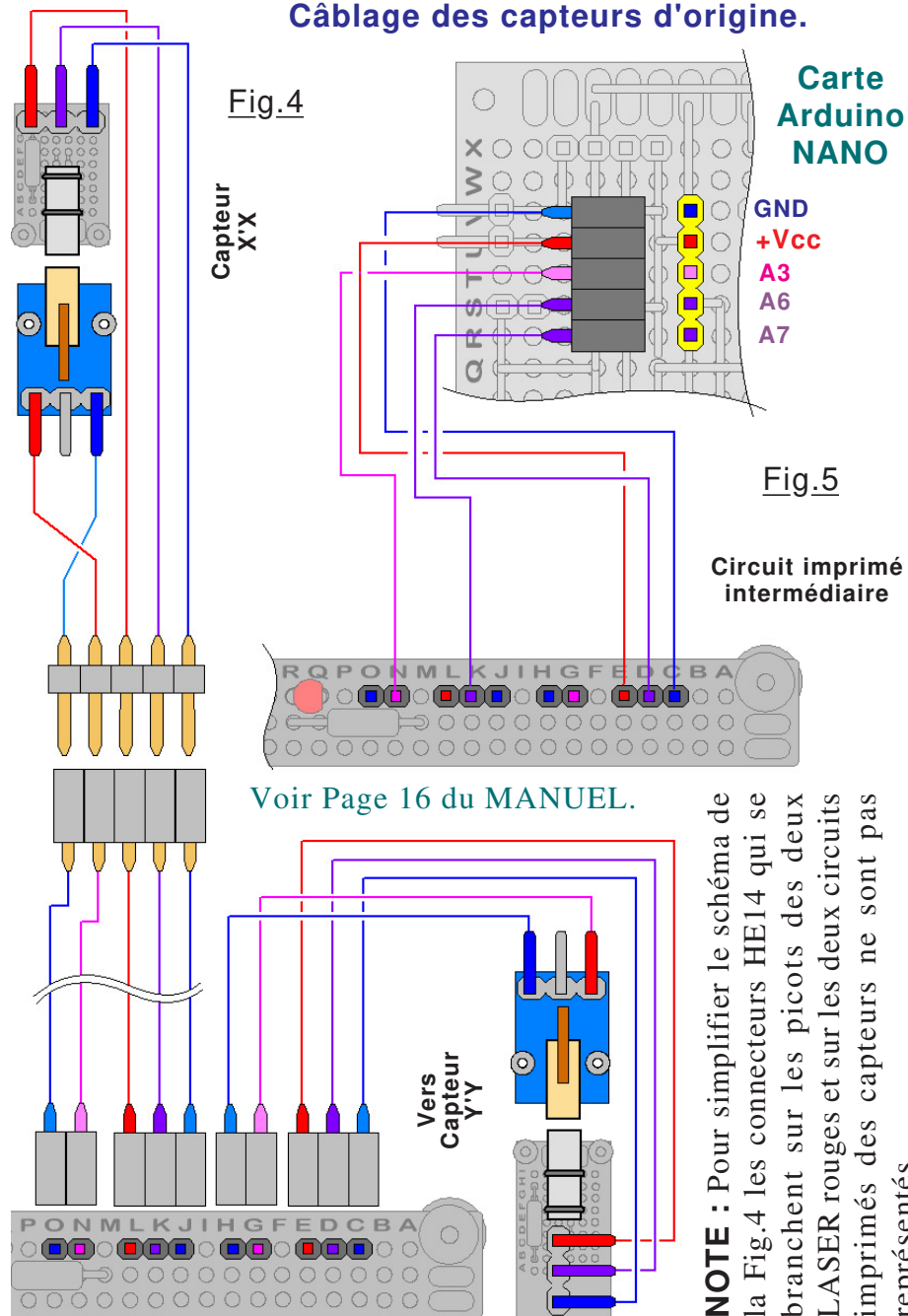
Fiche de câblage des circuits électriques. 2/5

## Câblage du PUPITRE.



## Fiche de câblage des circuits électriques. 3/5

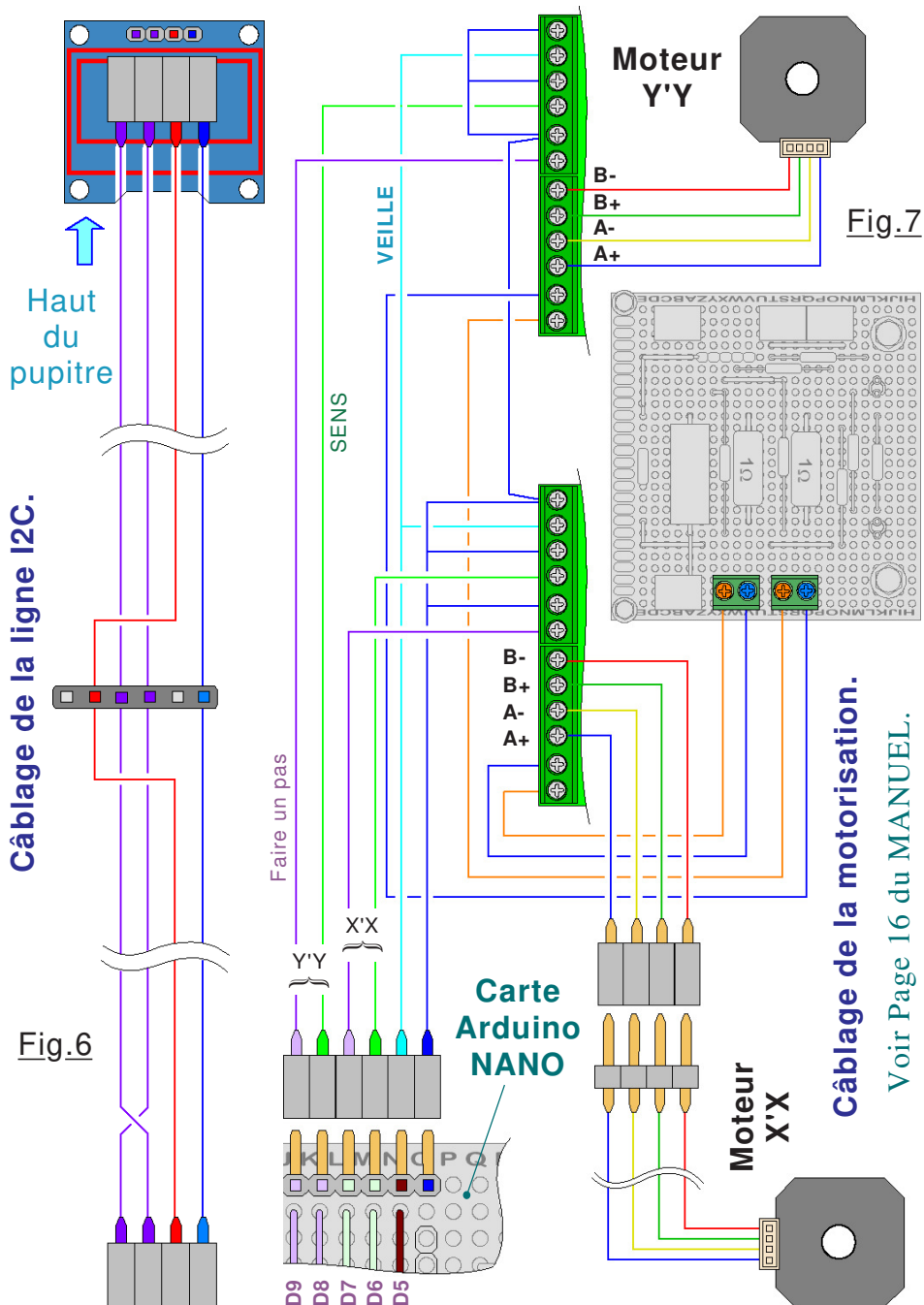
### Câblage des capteurs d'origine.



Page 25

## Fiche de câblage des circuits électriques. 4/5

### Câblage de la ligne I2C.



### Ignorer la saisie des Origines Machine :

N ombreuses sont les circonstances pour lesquelles on désire effectuer des manipulations sans la présence d'énergie de puissance. Si l'intervention consiste à effectuer une gravure virtuelle complète, la procédure commence par effectuer une capture des **Origines Machine**. L'opérateur doit alors masquer en séquence les deux capteurs, X'X en premier et Y'Y ensuite. Pour éviter d'avoir à conduire cette manipulation peu commode, il est prévu d'imposer au logiciel une **consigne pour que la capture d'origine soit ignorée jusqu'au prochain RESET**. Le protocole est le suivant :

- 1) Appuyer sur le bouton de RESET sans le relâcher,
- 2) Appuyer sur la touche **Y+ ↑** et la maintenir active,
- 3) Libérer le B.P. de RESET.
- 4) Attendre que la LED verte du clavier s'illumine.
- 5) Relâcher **Y+ ↑** pour terminer la procédure.

Le programme effectue alors la lecture de la carte **SD** puis, pour avertir l'opérateur que la capture des **Origines Machine** est désormais ignorée, la LED triple s'allume "en blanc". (Les trois couleurs sont activées simultanément.)

### Position LASER sur incident imposant un RESET :

P otentiellement un incident qui stopperait une opération de gravure et figerait la machine n'est jamais à exclure. Que ce soit par microcoupure secteur, parasite intempestif ou erreur dans le programme, il peut arriver que le processus soit suspendu et que la machine reste inerte en position. (Par exemple un problème de lecture de la carte **SD** qui bloque le décodage d'une ligne de **gcode** en boucle infinie.) Pour engager une procédure d'analyse de l'incident, il est intéressant de noter les coordonnées exactes du LASER lorsque le blocage s'est produit. **Le protocole est le suivant :**

- 1) L'incident s'est produit durant l'opération de gravure, donc en principe le sectionneur SÉCURITÉ fournit le **12Vcc** et l'inverseur Coupé est sur la position **LASER**.
- 2) Effectuer un RESET avec le protocole décrit ci-dessus pour faire **ignorer les Origines Machine**. (LED triple allumée "en blanc".)
- 3) Activer le menu **Mode MANUEL**. Focaliser le LASER. Immobiliser sur la table de la machine un jalon qui précise la position actuelle du LASER. (Morceau de papier avec une croix par exemple etc.)

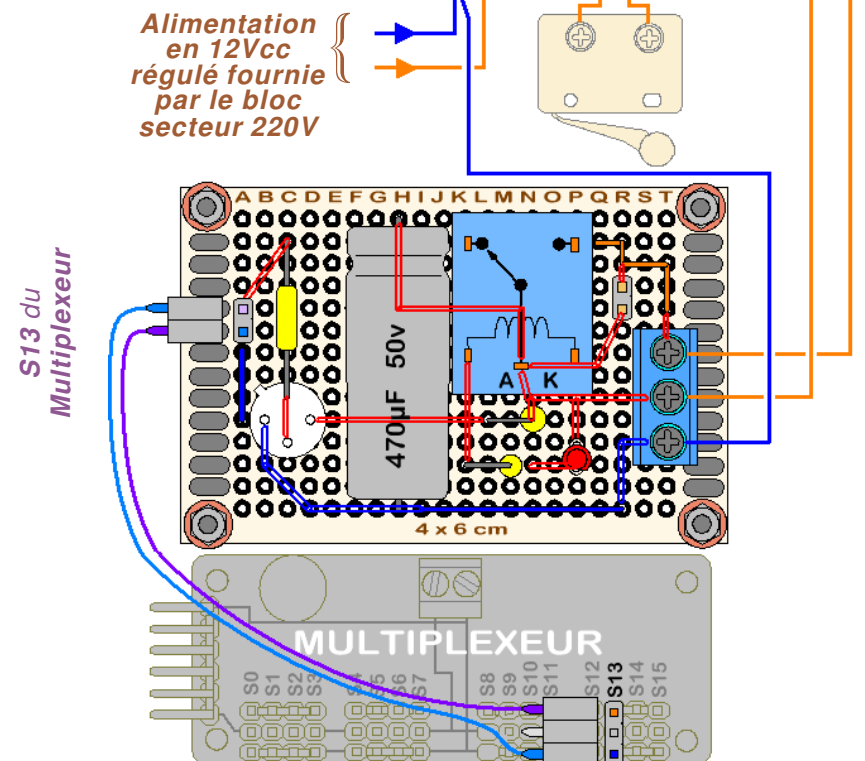
### Fiche de câblage des circuits électriques.

5/5

#### **Câblage du Relai LASER.**

@  
Ligne qui  
transite par la  
potence et qui  
va vers  
l'alimentation  
du LASER via  
l'interrupteur  
LASER/Coupé

Fig.8





## Le Relais d'alimentation du LASER.

Implanté lorsque la machine était achevée et les documents terminés, introduire le schéma et les dessins du circuit imprimé dans le MANUEL n'était plus raisonnable, imposant de reprendre entièrement la pagination, raison pour laquelle une fiche supplémentaire dédiée a été ajoutée en complément.

Lorsqu'une opération de gravure s'achève, la machine est replacée en configuration délogée, puis les moteurs sont mis en veille. La sortie de `Traite_ligne_a_ligne_le_fichier()` se termine par `Couper_le_LASER()` qui force à zéro la puissance lumineuse sur **S15** du multiplexeur. Comme l'électronique de pilotage du LASER fonctionne "à courant constant", bien que le module soit "neutralisé", il continue à consommer 200mA sur le 12V et la ventilation reste activée. (*Puissance consommée 2,4W.*) Pour éviter cette perte d'énergie si l'opérateur n'est pas présent, un relais de puissance dont le schéma Fig.1 précise l'agencement a été ajouté.

C'est la sortie **S13** du Multiplexeur PCA9685 qui pilote automatiquement le transistor de commutation **T**. Disposant de la place suffisante, un condensateur "réservoir" **C** de 470µF a été ajouté entre **GND** et le **+12Vcc**. (*Filtrage partiel de certains transitoires de commutation visibles avec un oscilloscope.*) Pour simplifier le câblage et ne pas ajouter la consommation du relais sur le module de régulation 5Vcc, le circuit puise son alimentation sur le **+12Vcc**. Une LED rouge signale l'état d'activation du relais **RY**.

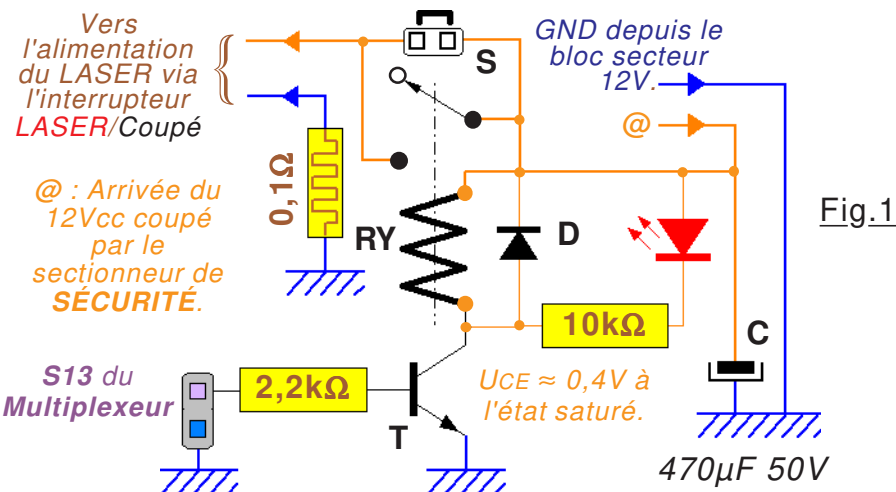


Fig.1

## Dessin du circuit imprimé pour le Relais du LASER.

Alimenté directement en 12V quand le transistor **T** est saturé, le relais **RY** consomme 30mA. Piloté en 5V par la sortie **S13** du Multiplexeur, la résistance de **2,2kΩ** est suffisamment faible pour saturer entièrement **T**. Tout transistor de type NPN conviendra. (*2N1711, 2N2222 etc.*) La diode "de roue libre" **D** n'est absolument pas critique non plus. Par exemple prendre une 1N4002 conviendra parfaitement. Le "strap" **S** sert à remplacer le contact travail du relais lorsque ce dernier n'est pas piloté par **S13** et que l'on procède à de la maintenance ou lors de la validation des divers circuits.

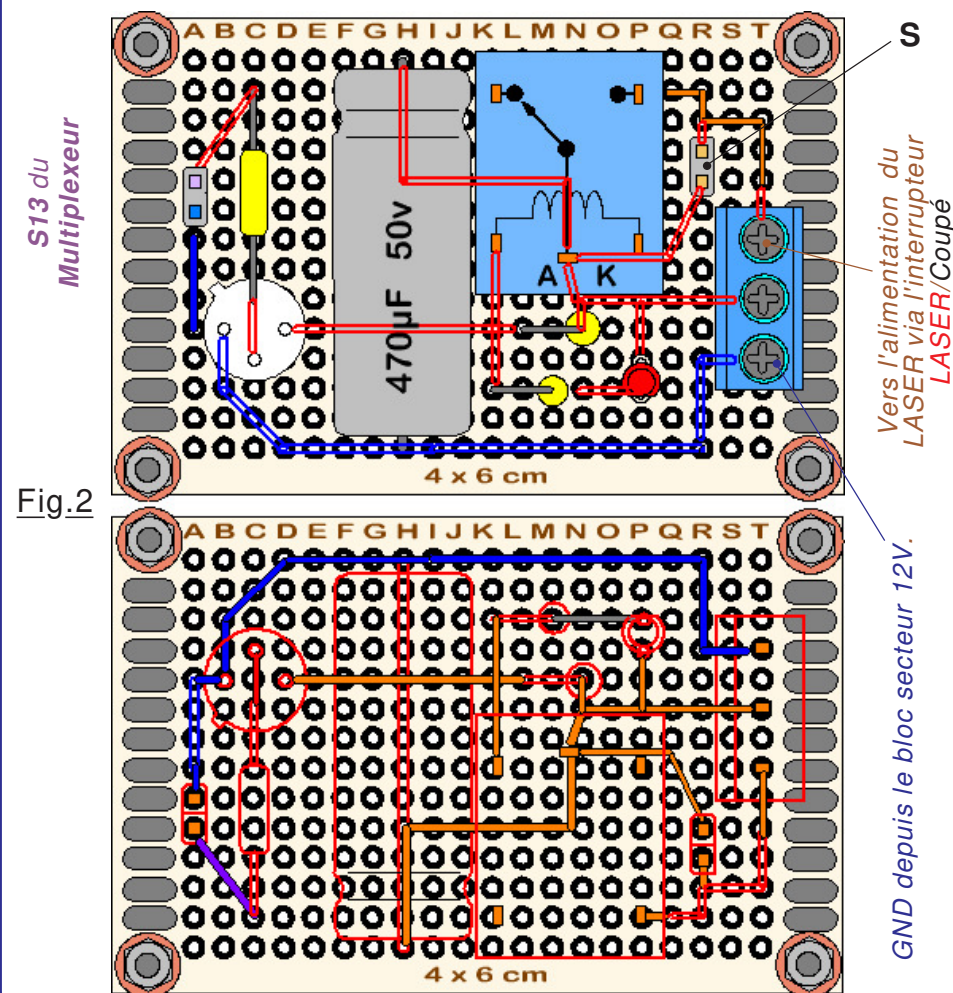


Fig.2

## BUS série au standard I2C. (1/2)

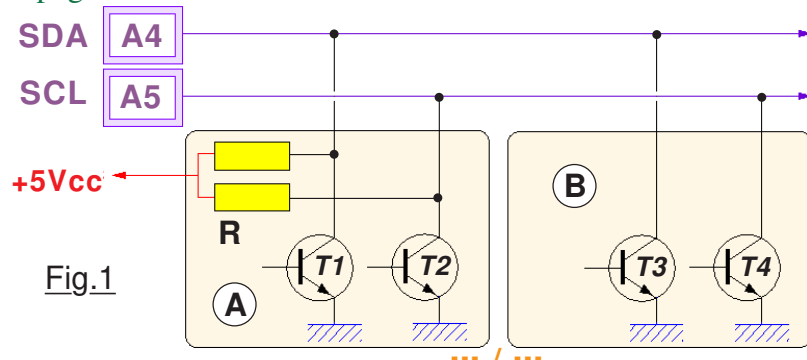
Développé initialement par Philips en 1982, le bus I2C s'est largement imposé dans le domaine des microprocesseurs, microcontrôleurs et applications industrielles diverses. Sa désignation dérive de **Inter-Integrated Circuit**. Il fut à l'origine conçu pour des applications de domotique et d'électronique domestique. La norme I2C est basée sur un **bus série synchrone bidirectionnel** fonctionnant en "half-duplex", où plusieurs périphériques maîtres ou esclaves peuvent communiquer entre eux. Les dialogues ont toujours lieu entre un seul maître et un ou tous les esclaves présents et l'échange de données est toujours déclenché à l'initiative du maître. *(Jamais de maître à maître ou d'esclave à esclave.)*

### ➤ Constitution matérielle du bus I2C.

Outre une masse commune **GND** la ligne n'utilise que deux fils :

- **SDA** (**S**erial **D**ata **L**ine) : Ligne de données bidirectionnelle,
- **SCL** (**S**erial **C**lock **L**ine) : Ligne d'horloge pour la synchronisation également de type bidirectionnel.

Les deux lignes sont maintenues à l'état logique "1" par un niveau de tension +VDD à travers des résistances de forçage. (*Pull-Up.*) Dans le standard I2C le nombre maximal de périphériques est limité à 128 par le nombre d'adresses disponibles, 7 Bits d'adressage et un Bit R/W. (*Lecture ou Écriture.*) Bien qu'une foule de combinaisons complexes soit possible, dans le cas d'Arduino c'est la carte ATmega328 qui sera toujours le Maître et les modules périphériques les esclaves. Traditionnellement si le programme d'application doit intégrer une ligne I2C, comme représenté sur la Fig.1 ci-dessous ce **sont A4 et A5** qui **sont respectivement affectées à SDA et SCL**. Les bibliothèques qui accompagnent les modules du commerce sont basées sur ce schéma.

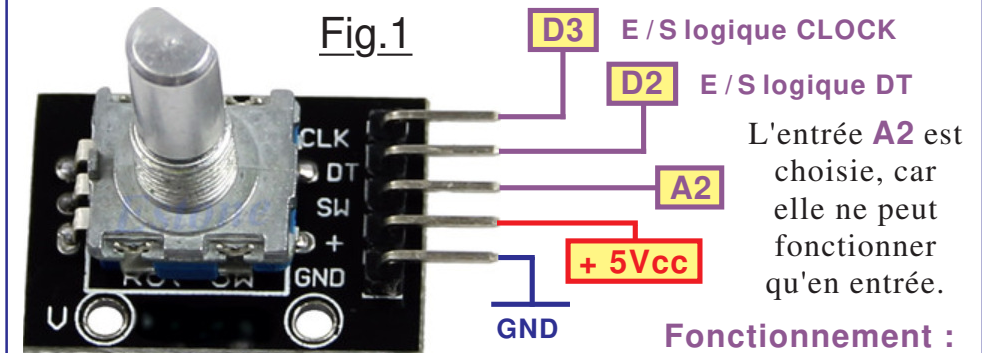


## Encodeur rotatif KY-040.

Le KY-40 est un codeur **sans butée** à 20 points par tour pourvu d'un "bouton poussoir" par appui sur la tige de commande centrale. La sortie se fait par deux lignes pilotées par des capteurs de type codage Gray. La Fig.1 donne le schéma des branchements à réaliser sur ARDUINO pour utiliser le programme "de la Raquette" **P40**.

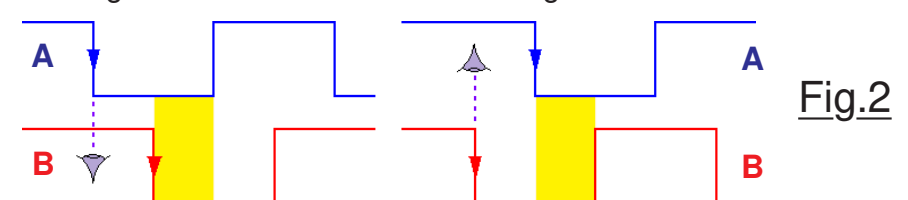
### Caractéristiques techniques du module KY-040 :

- Consommation maximale : 10 mA sous 5 Vcc.
- Température de fonctionnement : - 30 à + 70 °C.
- Température de stockage : - 40 à + 85 °C.
- Durée de vie du capteur de rotation : Minimum 30 000 cycles.
- Durée de vie du contact central : Minimum 20 000 cycles.
- Résistance de passage du contact de RAZ : 3 Ω maximum.



Concrètement les trois sorties sont alimentées au + 5 Vcc par des résistances de 10 kΩ. La sortie **SW** est celle de l'inverseur piloté par appui sur la tige du rotor. Les deux sorties **CLK** et **DT** sont en réalité deux sorties classiques avec déphasage de 90° souvent nommées **A** et **B** pour ce type de capteur. Le déphasage de 90° électriques des signaux **CLK** et **DT** permet de déterminer le sens de rotation. (*Voir Fig.2*) **Le capteur est traité par interruptions.**

A change d'état avant B      B change d'état avant A



## Petit module LASER.

### Caractéristiques techniques du petit module LASER :

- C'est une simple diode LASER associée à une résistance de  $103 \Omega$  montée en série pour limiter le courant.
- Longueur d'onde 650 nm. (*Couleur rubis*)
- Puissance lumineuse 2 à 5 mW.
- Le **+5Vcc** peut être appliqué directement sur la broche d'alimentation, mais pour augmenter la durée de vie de la diode LASER il est possible d'insérer une résistance. Le tableau proposé ci-dessous précise le courant d'alimentation en fonction de **R**, la tension d'alimentation étant de +5Vcc.
- Le courant maximal consommé est compatible avec la sortance d'une broche binaire de l'ATmega328. Il est parfaitement possible de moduler la luminosité par PWM, la fréquence de 490Hz de découpage étant largement assez élevé pour donner l'impression d'une clarté constante.

R	0	22 $\Omega$	47 $\Omega$	82 $\Omega$	100 $\Omega$
I	22 mA	19 mA	17 mA	14 mA	14 mA

Fig.1

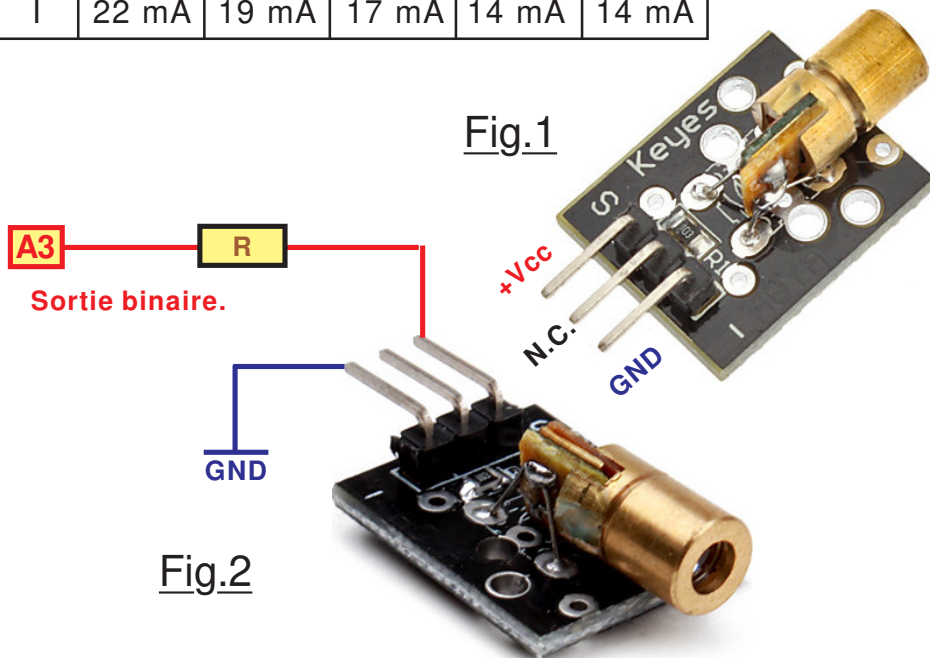
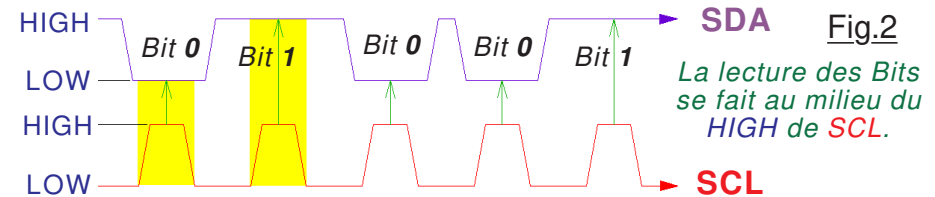


Fig.2

## BUS série au standard I2C. (2/2)

### ➤ Signaux échangés sur un bus I2C.

Le niveau **HIGH** ou **LOW** de la ligne **SDA** doit être maintenu stable pendant le niveau **HIGH** sur la ligne **SCL** servant à déclencher la lecture successive des bits du protocole de dialogue. (*Voir la Fig.2*)



Comme montré sur la Fig.1 les équipements sont connectés au bus par des électroniques de type drain ouvert. (*Ou collecteur ouvert.*) Fonctionnant en ET câblés deux périphériques tels que **A** et **B** peuvent "parler" simultanément. Dans ce cas un état logique "0" "écrase" un état logique "1". Pour caractériser ce genre d'incident potentiel sur un bus I2C on utilise le vocable :

- L'état logique "0" **LOW** est un état dominant,
- L'état logique "1" **HIGH** est un état récessif.

Lorsque le bus n'est pas utilisé, il est forcé niveau haut par les résistances telles que **R** de l'un des modules connectés. Il suffit d'un seul rappel à **+5Vcc** pour que la ligne fonctionne. Si plusieurs modules sont pourvus de résistances de forçage, le courant qui devra être drainé par l'ATmega328 et les électroniques branchées sera plus important mais reste généralement faible car le nombre de périphériques est classiquement faible pour des applications ordinaires. Pour les modules dédiés à Arduino les vitesses de transmission sont généralement comprises entre 100kb/s et 400kb/s.

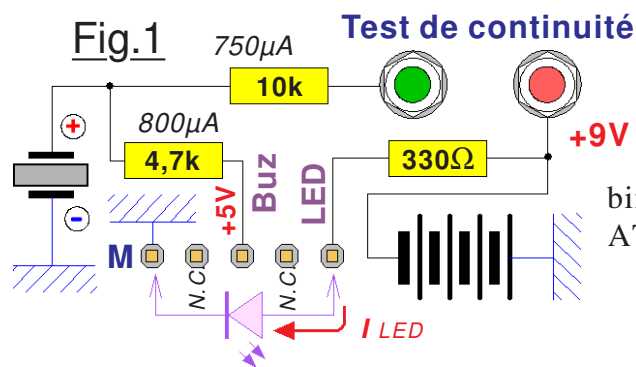
### ➤ Les bibliothèques de programme.

Pour simplifier le travail des programmeurs et se positionner sur le marché, les concepteurs de modules électroniques dialoguant en I2C fournissent des bibliothèques dédiées à leurs produits. Outre les "library" propres à chaque référence commerciale, la bibliothèque **Wire.h** est spécialisée pour gérer sur carte Arduino les protocoles I2C/TWI et devra parfois accompagner celle qui accompagne un module électronique spécialisé. C'est elle qui impose l'usage de **A4** et **A5**.



## TESTEUR DE CONTINUITÉ ET DE LED.

Conceptuellement ce petit dispositif n'est pas spécialement dédié à l'utilisation d'Arduino puisqu'il sert fondamentalement à tester la continuité des lignes électriques au moyen d'un buzzer "actif". C'est avant tout un outil facilitant le câblage de faisceaux électriques. Mais son buzzer actif, consommant moins d'un milli ampère, est totalement compatible avec les sorties binaires d'Arduino et peut en constituer un périphérique. *(La résistance de  $330\Omega$  était prévue pour des LEDs de mauvais rendement. Avec les composants actuel personnellement je privilégierais une  $4,7k\Omega$ .)* La Fig.1 en donne le schéma électronique. En tant que testeur de continuité il est totalement autonome avec son accumulateur rechargeable de



9Vcc, mais il peut aussi fonctionner comme un simple dispositif relié à l'une des sorties binaire du circuit intégré ATmega328.

Accumulateur 8,4Vcc rechargeable ou pile 9Vcc de type LR61

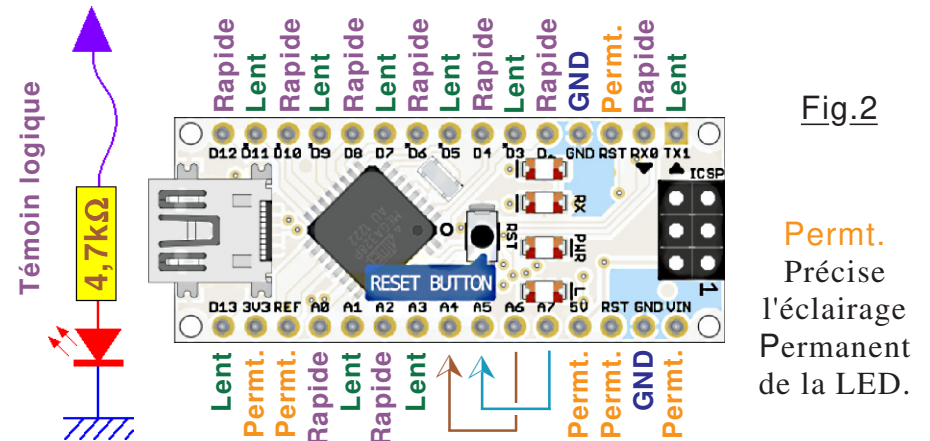
Les prises pour tester la continuité sont colorées pour repérer le sens du courant dans l'éventualité d'une manipulation sur un élément polarisé. La prise négative n'est pas bleue pour ne pas confondre avec la masse du montage qui n'est disponible que sur le picot du petit connecteur. La couleur verte précise bien la polarité sans pour autant induire l'idée d'une masse. Le test de continuité est sobre en

Couleur	$I_{LED}$
Rouge	20 mA
Verte	18 mA
Jaune	19 mA
Blanche	16 mA
Bleue	16 mA
Court circuit	24 mA

courant puisque sur un court circuit des deux prises le courant reste inférieur à un milli ampère. Si le dispositif doit servir d'élément sonore pour Arduino, l'entrée **Buz** sera reliée directement à l'une des sorties "digitales" **D0** à **D13** du microcontrôleur sans oublier de réunir les masses. La résistance de

## Vérification du circuit imprimé principal. 2/2

Résumé du comportement attendu par le programme :



- 6) Les 14 sorties binaires clignotent à une cadence d'environ 0,8S. Pour pouvoir s'assurer que deux broches voisines ne sont pas en contact intempestif, les sorties paires génèrent des impulsions lumineuses très courtes, alors que les sorties impaires s'illuminent à peine durant 0,21 seconde. Les broches d'entrées analogiques programmées en sorties **A0**, **A1**, **A2** et **A3** fonctionnent de la même façon. Pour vérifier les deux entrées Analogiques **A6** et **A7** qui ne peuvent pas fonctionner en sorties, elles "recopient" leur état respectivement sur **A4** pour l'entrée **A6** et sur **A5** pour l'entrée **A7**. Pour tester ces deux entrées, il suffit de les relier soit à **GND**, soit au **+5Vcc** alternativement. Le témoin logique branché sur la sortie associée sera éteint ou allumé. (À la temporisation près.)

### Valider une carte NANO.

Avec **P00\_Tester\_la\_carte\_NANO.ino** n'impose en réalité strictement aucun circuit imprimé spécifique, car on peut brancher les fils d'expérimentation directement sur les broches de liaison du petit circuit imprimé. Par ailleurs, pour téléverser le croquis, on branche la ligne Mini-USB. Le circuit électronique est alors alimenté en 4,4Vcc. Déjà sans rien connecter deux LEDs du module clignotent alternativement.

## Vérification du circuit imprimé principal. 1/2

**P**remière vérification impérative, se munir d'une loupe à fort grossissement, nommée "compte fils" et vérifier visuellement chaque soudure et s'assurer qu'elle ne déborde pas accidentellement sur un élément voisin. (*Pastille, fil de liaison, broche d'un HE14 ...*) On peut examiner les nombreuses liaisons et leurs isollements avec un multimètre, mais franchement un **TESTEUR DE CONTINUITÉ** sera bien plus convivial car on n'a pas à regarder un cadran. **La validation commence toujours sans la présence de la carte Arduino NANO.**

- 1) Vérifier l'extrémité de chaque ligne conductrice en pointant la broche du connecteur concerné et la lyre associée sur le support de la carte Arduino NANO. Par exemple le +5Vcc sur le connecteur fournissant l'alimentation régulée, et la broche notée **5V**. On peut aussi tester entre **GND** sur le connecteur fournissant l'alimentation régulée, et la broche **GND** d'Arduino. ATTENTION, il y a deux broches **GND**, une seule va au connecteur d'alimentation. L'autre sera pontée par le circuit imprimé. (*Voir la Fig.1*)

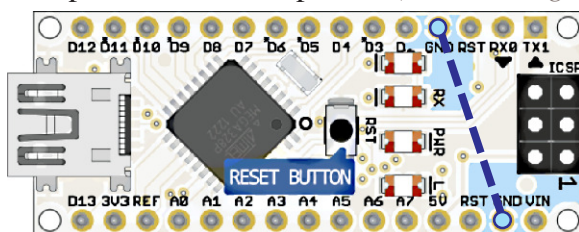
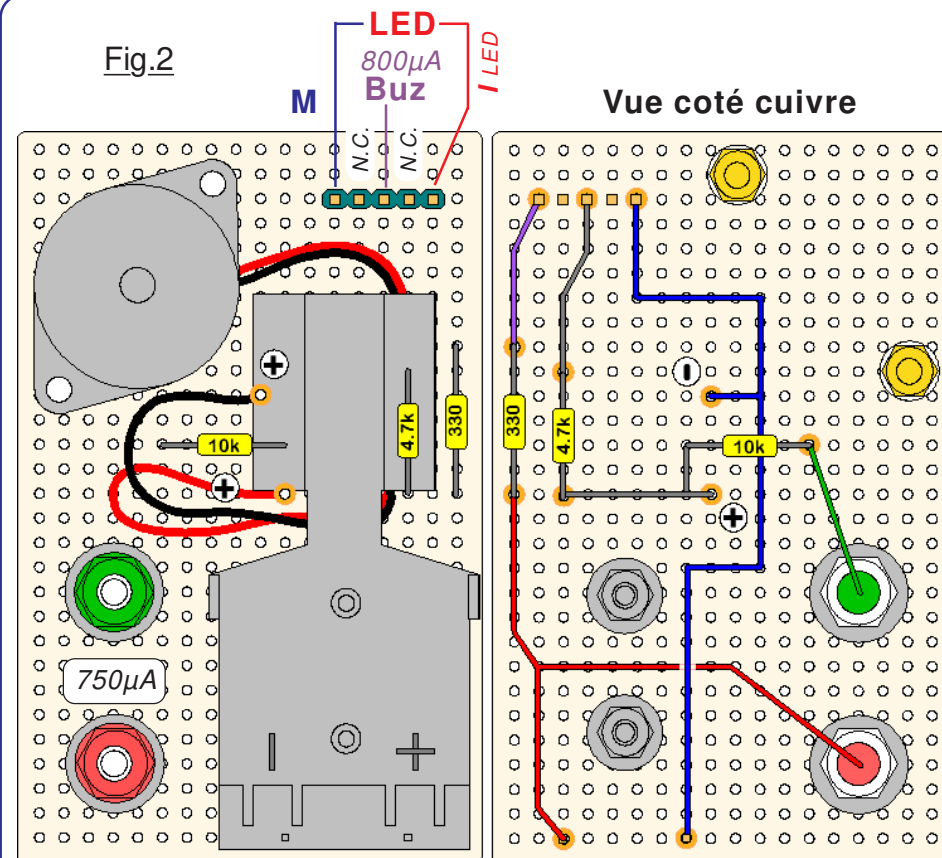


Fig.1

- 2) Quand l'intégralité des liaisons a été vérifiée ainsi que tous les isollements de voisinage, brancher le 5V sur le HE14 à quatre broches dédié. **La consommation doit être strictement nulle.**
- 3) On doit retrouver l'énergie sur le petit connecteur à deux broches situé à coté de celui de RESET. En pontant les deux broches du témoin logique à ce dernier, la LED doit s'allumer.
- 4) Pontre le **+5Vcc** sur la lyre du microcontrôleur qui correspond à la sortie binaire **D4**. Le Buzzer doit s'activer.
- 5) Couper l'alimentation. Mettre en place la carte Arduino NANO. Relier la prise mini USB à un P.C. et télécharger le programme de servitude **P00\_Tester\_la\_carte\_NANO.ino** qui va servir à vérifier l'intégralité des broches d'Entrée/Sorties. Immédiatement, comme précisé dans l'encadré situé en bas de cette fiche, du côté verso, deux LEDs se mettent à clignoter.

... / ...

Fig.2



limitation de courant est intégrée à ce petit montage, et le buzzer ne drainera que 800µA quand la sortie d'Arduino passera à "1".

**P**uisque autonome par son petit accumulateur de 9Vcc, ce dispositif permet facilement de tester une LED dans des conditions nominales qui généralement avoisinent les 20mA. Le courant traversant la diode électroluminescente sera fonction de sa couleur. Le petit tableau donné en recto indique l'ordre de grandeur des courants **I LED** mesurés sur quelques échantillons testés. Ce courant varie peu en fonction des composants mesurés. Si on désire vérifier un dispositif sous faible courant, (*Inférieur à 800µA*) rien ne s'oppose à le brancher sur les deux prises pour fiches bananes, en ayant soin de respecter une éventuelle polarité. La Fig.2 donnée ci-dessus montre le dessin du petit circuit imprimé qui supporte tous les composants, la pile de 9Vcc n'étant pas montrée dans son support.