

## Recommandations pour réaliser les fiches.

Par Nulentout (Achévé le 24 Juin 2024.)

Mise 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 côté 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.

Réaliser une "paire de fiche" n'est pas spécialement compliqué, toutefois le fait d'avoir à imprimer des deux côtés 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 forcément un 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.

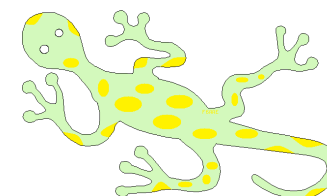
Personnellement, quand j'imprime de telles fiches, je complète leur réalisation en les plastifiant. Outre la belle apparence que cette pratique leur confère, elles sont bien plus rigides et agréables à manipuler. C'est d'autant plus facile qu'actuellement une petite plastifieuse au format A4 est d'un coût peu élevé et devenu vraiment banal dans les boutiques "informatiques".

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

RECTO	VERSO	RECTO	VERSO
2	3	4	5
RECTO	VERSO	RECTO	VERSO
6	7	8	9
RECTO	VERSO	RECTO	VERSO
10	11	12	13
RECTO	VERSO	RECTO	VERSO
14	15	16	17
RECTO	VERSO	RECTO	VERSO
18	19	20	21

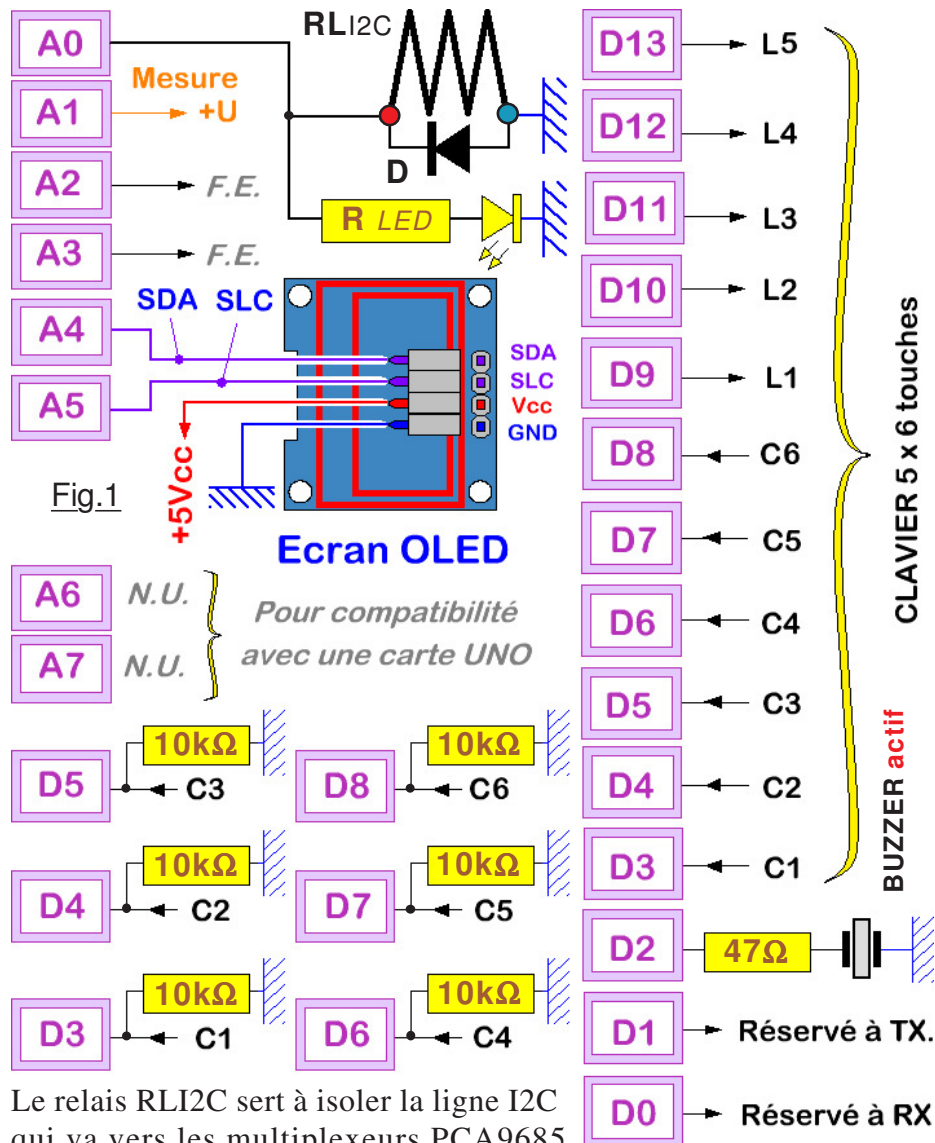
Fig.1

**NOTE :** Les deux dernières Fiches qui à mon sens doivent impérativement être imprimées, plastifiées et rangées avec la machine ne sont pas numérotées. Pour toutes les autres qui sont ordonnées, ce n'est pas indispensable et vous pouvez vous contenter de les consulter sur l'ordinateur ou une tablette.



## Fiche n°1.

### Affectation des Entrées / Sorties.

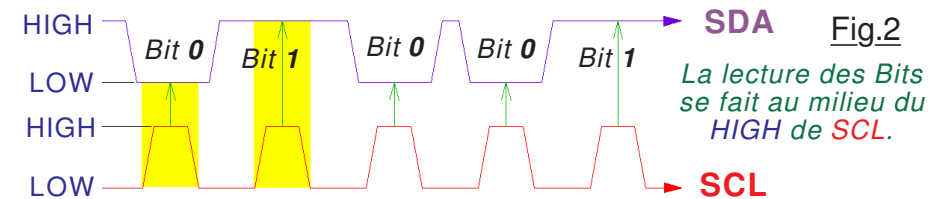


## Fiche n°4.

### 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.

Les concepteurs de modules électroniques dialoguant en I2C fournissent des bibliothèques dédiées à leurs produits. Outre les "library" propres à chaque référence commerciale, **Wire.h** est spécialisée pour gérer sur Arduino les protocoles I2C et devra parfois accompagner celle qui accompagne un module électronique spécialisé. C'est elle qui impose l'usage de **A4** et **A5**.

### Fiche n°3. BUS série au standard I2C. (1/2)

**D**éveloppé initialement en 1982, le bus I2C s'est largement imposé dans le domaine des microprocesseur, et des applications industrielles. 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 **GND** la ligne n'utilise que deux fils "bidirectionnel" :

- **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.)* Dans le cas d'Arduino c'est la carte ATmega328 qui sera toujours le Maître et les modules périphériques les esclaves. 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** contrainte imposée par les bibliothèques qui accompagnent les modules du commerce

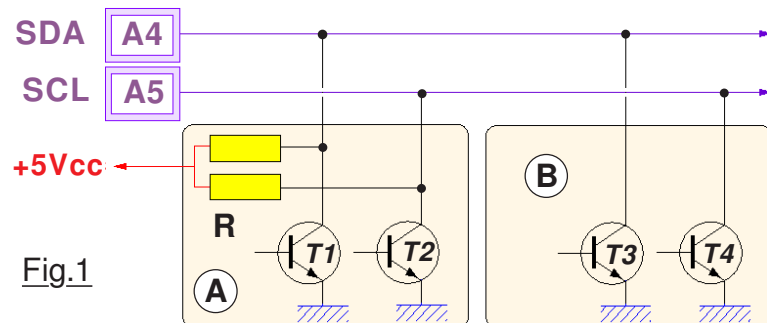


Fig.1

### Fiche n°2.

### Afficheur graphique OLED 1,3 pouce.

**L**es références d'afficheurs graphiques sont kyrielles, avec des dimensions et des caractéristiques très différentes d'un modèle à l'autre. Leurs prix d'achat aussi incitent à éliminer ceux qui sont en couleur, les grandes définitions, les écrans tactiles etc. Le choix pour cette application s'est porté sur l'écran OLED de la Fig.1 monochrome blanc ou bleu pilotable par I2C de définition 128 x 64 Pixels et de diagonale 1,3 Pouces. Seulement deux broches de pilotage sont



Fig.1

nécessaires pour le gérer. Muni d'un contrôleur SSD1306 ils sont lumineux, donc sans rétro-éclairage, de couleur blanche ou bleue selon la référence approvisionnée. *(REMARQUE : Noter avant de passer commande qu'il existe aussi en 0,96 pouces de diagonale. C'est bien celui qui fait 1,3 pouces qui est mis en œuvre dans ce projet.)* **ATTENTION : Certaines références ne sont pas compatibles avec U8glib.h** donc vérifier à la commande.

J'ai approvisionné le mien à l'adresse suivante :

[https://www.amazon.fr/gp/product/B07FYG8MZN/ref=ppx\\_yo\\_dt\\_b\\_asin\\_title\\_o04\\_s00?ie=UTF8&psc=1](https://www.amazon.fr/gp/product/B07FYG8MZN/ref=ppx_yo_dt_b_asin_title_o04_s00?ie=UTF8&psc=1)

**ATTENTION DANGER :** L'afficheur de la Fig.1 est disponible sur plusieurs sources dans le commerce en ligne. Toutes sont équivalentes et compatibles avec U8glib.h. Toutefois il faut bien faire attention car en fonction des adresses d'approvisionnement **les deux broches GND et VCC sont inversées.**

La bibliothèque <U8glib.h> est fournie dans <Documents> et accompagnée du fichier **Bibliothèque U8glib.pdf** formaté pour être imprimé en Recto/Verso et réaliser un petit livret au format A5 qui en résume sur vingt pages les méthodes. Également fourni **Réaliser un petit livret.pdf** pour vous aider à l'assembler.

## Fiche n°5.

### Méthodes de Adafruit\_PWMServoDriver.h.

**S**pécifique au multiplexeur PCA9685 à 16 canaux PWM cette bibliothèque permet facilement de gérer chaque sortie indépendamment l'une de l'autre. L'effet d'une commande reste effectif jusqu'à une nouvelle consigne pour le canal concerné. Toute utilisation de cette bibliothèque doit être précédée de la déclaration :

```
#include <Adafruit_PWMServoDriver.h>
```

### Créer une instance du module PCA9685.

```
pwm.begin();
```

Cette instruction doit être placée en premier dans **void setup()** et permet de générer une instance du module PCA9685.

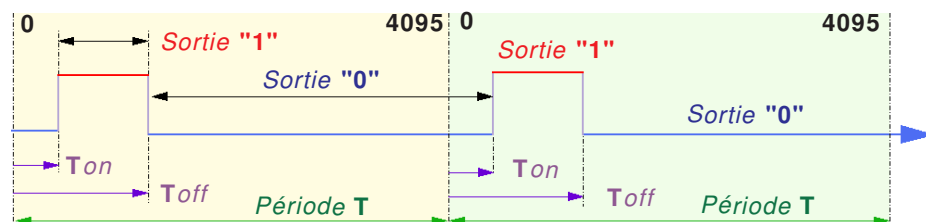
### Fréquence du signal généré sur les 16 canaux.

```
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 40 et 1000, mais en standard adopter 50Hz. (La période **T** fait alors 20mS pour des servomoteurs.)

### Principe de la génération PWM.

La génération PWM est basée sur un compteur à 12 BITS qui fonctionne à une cadence définie par **pwm.setPWMFreq()** et qui en détermine la période **T**. Le signal présentera une période



identique sur toutes les sorties d'un même module. L'instruction qui conditionne une sortie contient deux paramètres **Ton** et **Toff** qui permettent de définir la durée à l'état "1". Ce sont les valeurs du compteur qui déclenchent les changements d'état :

**Sortie "1"** = **Toff** - **Ton**.

**Sortie "0"** = **Période T** - **Sortie "1"**.

Le rapport cyclique et la durée des deux états est donc directement fonction de la fréquence de répétition programmée.

## Fiche n°8.

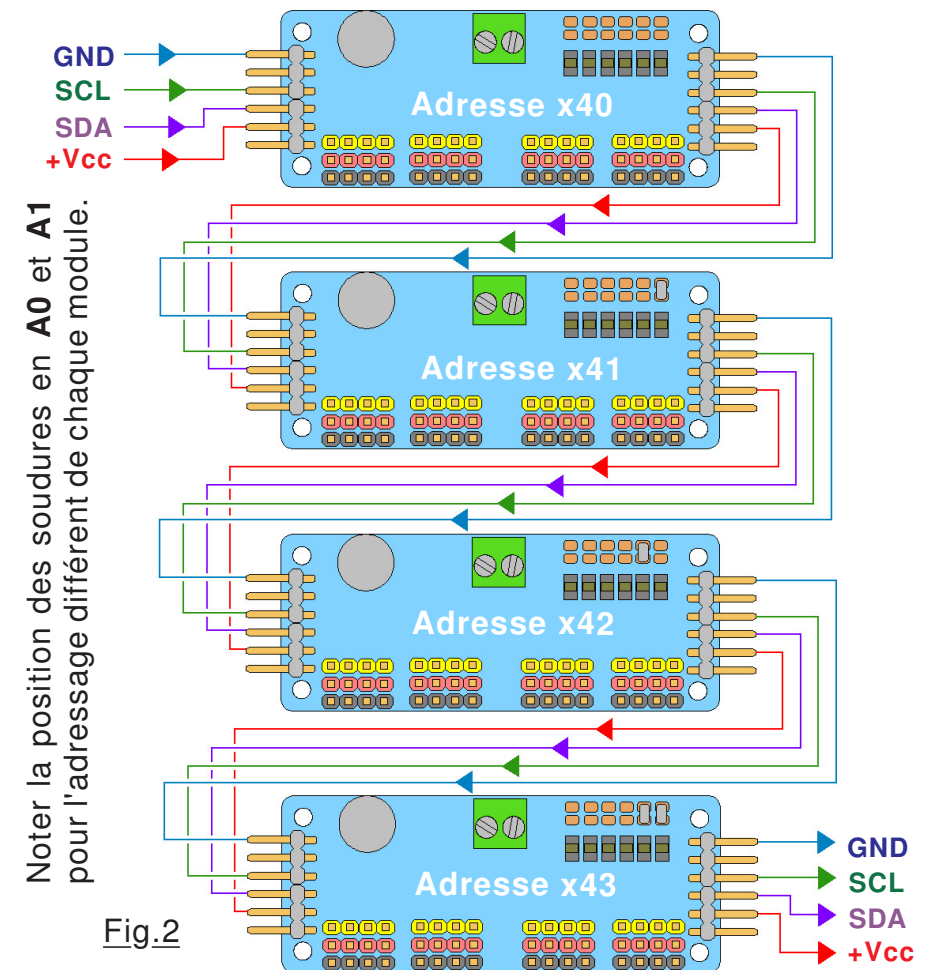
```
pwm4.begin(); pwm2.setPWMFreq(100);
```

```
pwm5.begin(); pwm1.setPWMFreq(1000);
```

On notera au passage que la fréquence de la PWM est propre à chaque module et peut librement être choisie entre 40Hz et 1000Hz.

### ➤ Chaînage électrique de plusieurs modules.

Coupler en "série" un nombre quelconque de modules PCA9685 est enfantin. Il suffit de les réunir "broche à broche" comme montré sur la Fig.2 sur leurs connecteurs situés aux deux extrémités du petit circuit imprimé.



Noter la position des soudures en A0 et A1 pour l'adressage différent de chaque module.

Fig.2

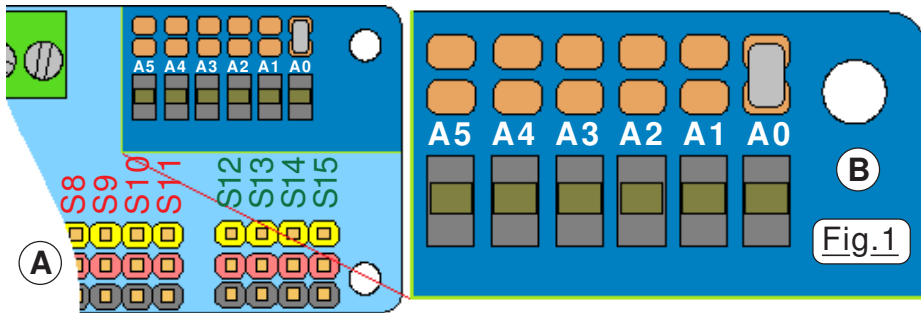


## Fiche n°7.

### Chaînage de deux multiplexeurs PCA9685.

Conçus pour être mis en cascade, on peut facilement en assembler jusqu'à trente trois, tous pilotés par la même ligne I2C à condition d'en répartir les adresses entre x40 et x73.

Considérons la Fig.1 **A** qui situe la zone d'adressage du module. En **B** cette zone est agrandie. Dans cet exemple en **A0** on a réuni



les deux pastilles cuivrées en les soudant. En comparant avec les données du chapitre ci-dessous on en déduit que son adresse est **x41**. S'il n'y a aucune soudure ajoutée, par défaut l'adresse est **x40**.

#### ➤ Principe de l'adressage.

Ce dernier se fait "en binaire" comme précisé ci-dessous :

Module 0: Adr = **x40** Adr binaire 00000 (*Pas de soudure*)

Module 1: Adr = **x41** Adr binaire 00001 (*Soudure sur A0*)

Module 2: Adr = **x42** Adr binaire 00010 (*Soudure sur A1*)

Module 3: Adr = **x43** Adr binaire 00011 (*Soudure A0 & A1*)

Module 4: Adr = **x44** Adr binaire 00100 (*Soudure sur A2*)

#### Déclarer les divers modules PCA9685.

```
Adafruit_PWMServoDriver pwm1 = Adafruit_PWMServoDriver(0x40);
```

```
Adafruit_PWMServoDriver pwm2 = Adafruit_PWMServoDriver(0x41);
```

```
Adafruit_PWMServoDriver pwm3 = Adafruit_PWMServoDriver(0x42);
```

```
Adafruit_PWMServoDriver pwm4 = Adafruit_PWMServoDriver(0x43);
```

```
Adafruit_PWMServoDriver pwm5 = Adafruit_PWMServoDriver(0x44);
```

#### Créer les instances des circuits PCA9685.

```
pwm1.begin(); pwm1.setPWMFreq(50);
```

```
pwm2.begin(); pwm2.setPWMFreq(60);
```

```
pwm3.begin(); pwm1.setPWMFreq(80);
```

... / ...

## Fiche n°6.

### Méthodes de Adafruit\_PWM servoDriver.h.

Fondamentalement le module PCA9685 est créé pour générer de la PWM dans des applications quelconques. C'est la raison pour laquelle on peut librement définir la fréquence de répétition et le rapport cyclique. On peut par exemple piloter en "analogique" des éclairages, des résistances chauffantes etc. *Les exemples et la formule donnée ci-dessous sont donc restrictifs à une période imposée "arbitrairement" à 20mS pour piloter des servomoteurs.*

#### Largeur de l'impulsion "positive" générée.

```
pwm.setPWM(Num_sortie, Ton, Toff)
```

Cette instruction engendre sur la sortie **Num\_moteur** un signal PWM à la **Fréquence** prédéfinie dont la durée de l'état "1" est définie par les valeurs **Toff - Ton** sur une "portée" 4095.

Pour s'affranchir de la différence, **Ton sera généralement égal à 0**.  
Compteur = 4095 pour 20mS soit 20000µS.

On a un comptage pour 20000 / 4095 soit toutes les 4,884µS.

**Toff à indiquer = Sortie "1" / 4,884** (Sortie "1" désirée en µS.)

Exemple : Sortie "1" désirée = 2272µS.

$Toff = 2272 / 4,884 = 465 \Rightarrow \text{pwm.setPWM(Num\_Sortie, 0, 465);}$

#### Pilotage des servomoteurs sans butée.

Chaque moteur peut tourner sans fin. Ils ont des caractéristiques voisines pour la vitesse et le sens de rotation. Toutefois la dispersion de caractéristique fait que d'un moteur à l'autre la **valeur** à adopter **pour le repos stable** sera à déterminer expérimentalement.

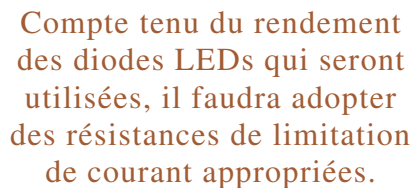
#### Ligne I2C et adressage du module :

Lire le contenu de la **Fiche n°2** et de la **Fiche n°3**.

La bibliothèque **Adafruit\_PWM servoDriver.h** impose pour la ligne **I2C** le signal **SDA** sur **A4** et le signal **SLC** sur **A5**. L'adressage par défaut sur le multiplexeur PCA9685 est en 0x40. On peut à la demande, ce qui sera obligatoire si on chaîne plusieurs modules, modifier l'adresse matériellement par établissement de contacts sur des ponts disponibles sur le circuit imprimé. Lire le contenu de la **Fiche n°7** et de la **Fiche n°8**.

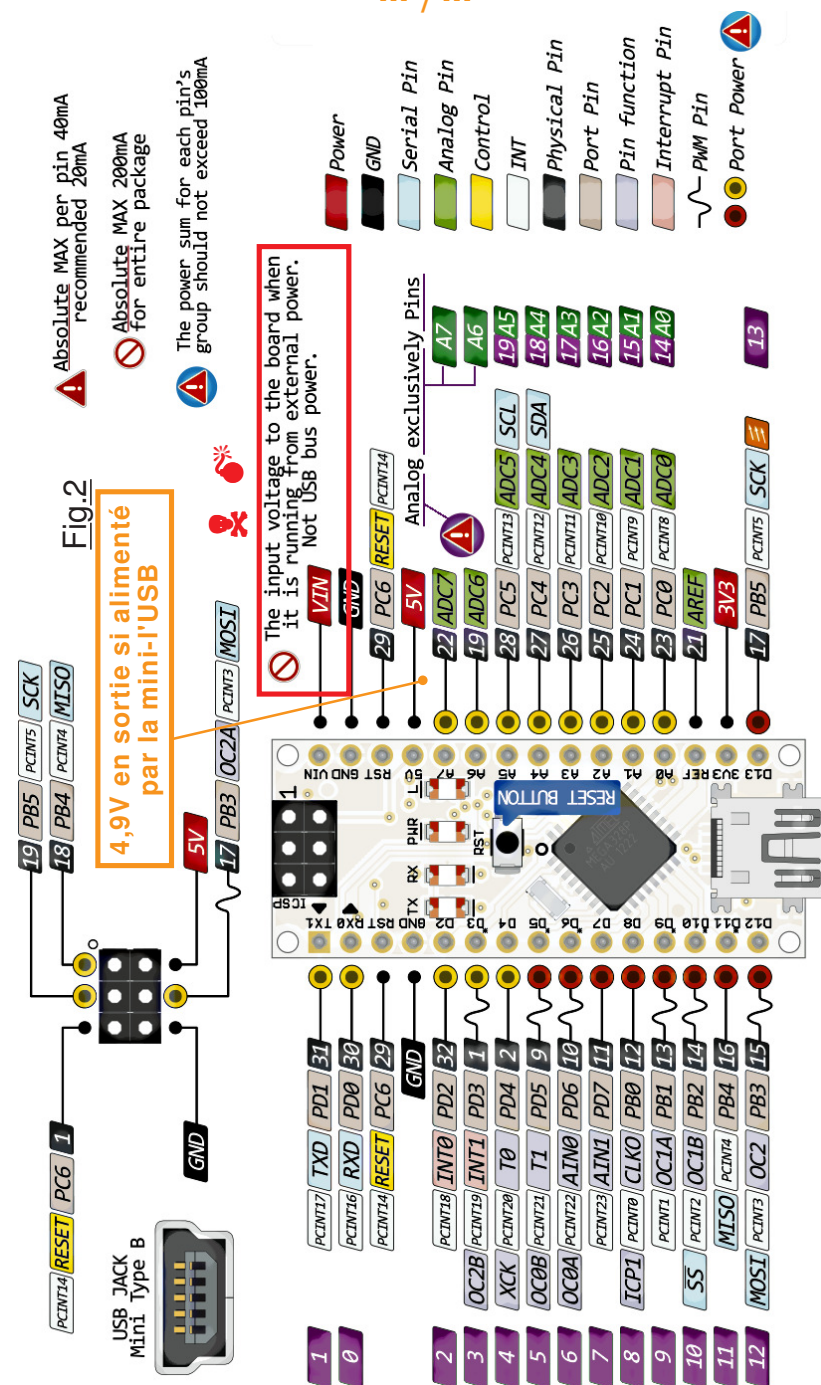
... / ...

### Branchements des deux multiplexeurs.



## Fiche n°12.

Fig. 2



## Verrouillage de la carte Arduino NANO.

## Fiche n°11.

### Présentation de la carte Arduino NANO.

L'arduino NANO se présente sous la forme d'une minuscule carte montrées sur la Fig.1 qui condense l'intégralité des fonctions d'un module 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 : Il ne faut pas de VIN simultanément avec la liaison Mini-USB ou le régulateur 5Vcc local sera détruit.**

Alimentée par le connecteur Mini-USB la carte fournit  $\approx 4,9V$  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**.

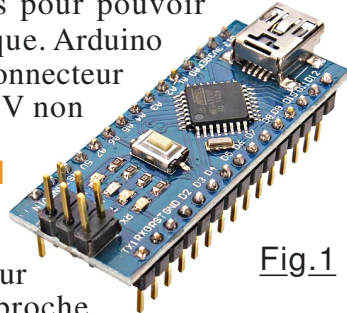


Fig.1

14 broches binaires. (Dont 6 fonctionnant en PWM et deux sont réservées au dialogue avec le **Moniteur de l'IDE**.)

8 broches d'entrées Analogiques dont 6 pouvant fonctionner en E/S.

Courant MAX par broche de sortie : 40 mA. (Total MAX : 100mA)  
L'ATmega328 a 32 Ko pour loger le programme, (Avec 2 KB utilisé pour le bootloader). Il dispose de 2 Ko de SRAM pour les variables 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)`.

- L'impédance d'une entrée binaire est de 5MΩ.
- L'impédance d'une entrée analogique est de 100MΩ.
- Les résistances internes de PULL-UP font entre 20MΩ et 50MΩ.

... / ...

## Fiche n°10.

### Tableau des LEDs simulant les ampoules.

Consultant le câblage des deux multiplexeurs sur la **Fiche n°9** on observe que la répartition des lettres sur les sorties **S0** à **S15** ne sont pas du tout dans l'ordre alphabétique. Cette organisation a pour but de faciliter le câblage des torons vers le tableau des LEDs en fonction de la présentation reproduite en Fig.1 dont le choix des couleurs facilite le repérage des lignes filaires.

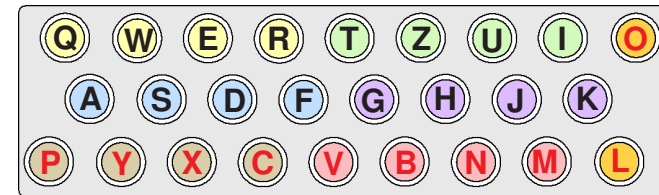


Fig.1

#### ➤ Procédure Allume\_Ampoule().

Compte tenu du fait que les vingt-six sorties sont réparties sur les deux multiplexeurs PCA9685 et que la distribution ne respecte pas un ordre alphabétique, la routine doit effectuer un triage pour adresser correctement le module concerné. Sur la Fig.1 les lettres en rouge sont relatives aux sorties du multiplexeur numéro deux d'adresse x41. La Fig.2

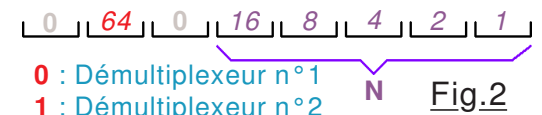


Fig.2

présente le codage sur un OCTET de chaque lettre, la procédure `Allume_ampoule()` traitant la variable `Caractere` qui contient l'ordre alphabétique de la lettre à traiter. Chaque caractère est codé dans `Ampoule[26]` dont les valeurs sont données ci-dessous :

A	B	C	D	E	F	G	H	I	J	K	L	M
8	5	3	10	2	11	12	13	7	14	15	8	7
0	1	1	0	0	0	0	0	0	0	0	1	1
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
6	9	0	0	3	9	4	6	4	1	2	1	5
1	1	1	0	0	0	0	0	1	0	1	1	0

Caractere = H : `Ampoule[7] = 13 + (0 x 64) = 13.`

Caractere = U : `Ampoule[21] = 4 + (1 x 64) = 68.`

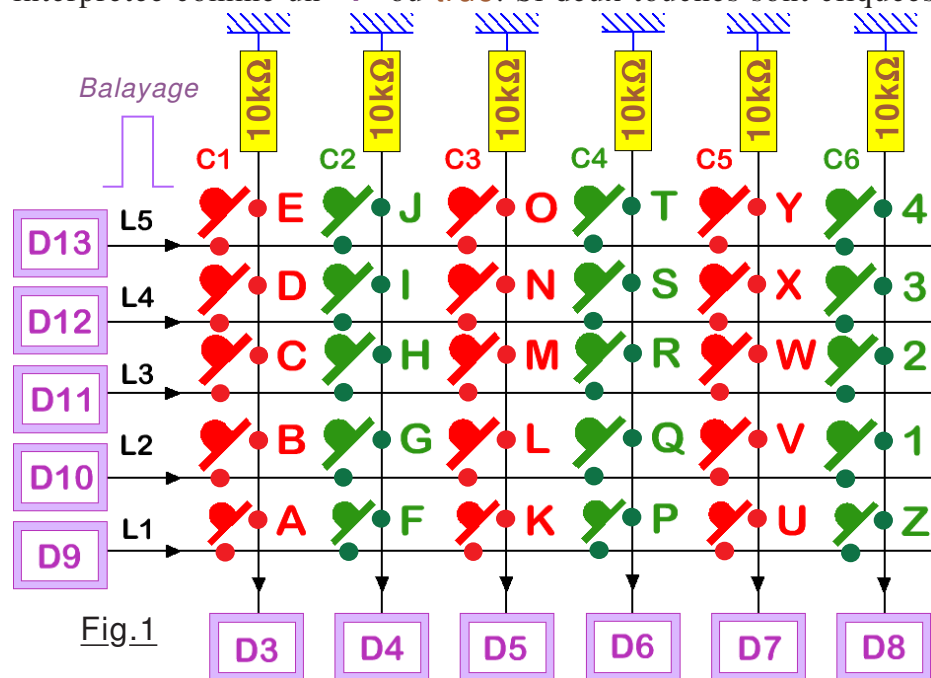
Caractere = P : `Ampoule[15] = 0 + (1 x 64) = 64.`



### Fiche n°13.

#### Architecture du CLAVIER à 30 touches.

Organisé en une matrice de cinq Lignes et six Colonnes la Fig.1 en traduit le branchement choisi pour simplifier au maximum les procédures de traitement. Les Lignes sont des sorties balayées par des états "1", les colonnes étant forcées à GND par des résistances de 10kΩ sont lues par des entrées binaires. La lecture donne un état "0" ou false si aucune touche n'est activée. Si l'un des B.P. est cliqué, la Colonne correspondante sera interprétée comme un "1" ou true. Si deux touches sont cliquées

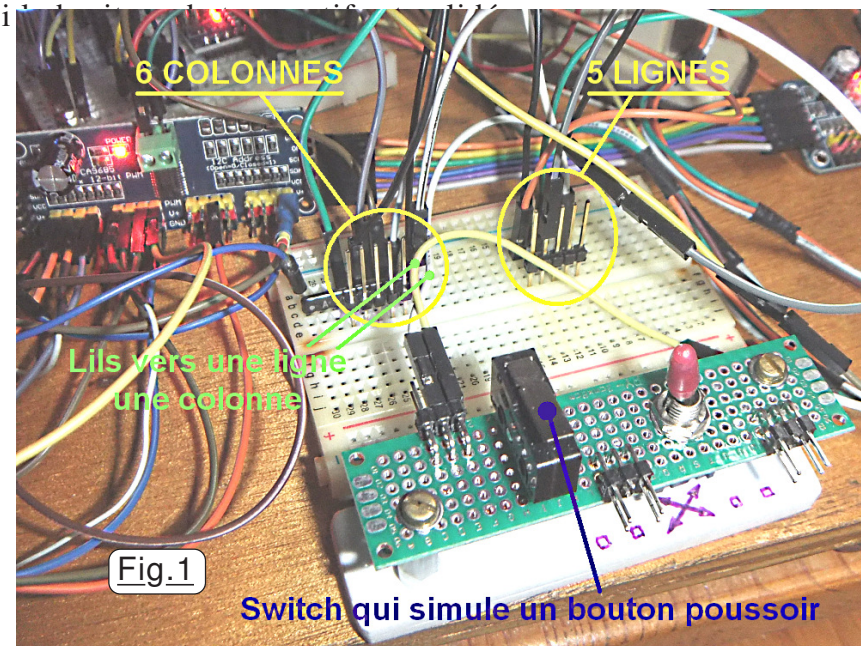


simultanément ce qui en pratique sera rare, seule l'une des deux sera retenue. La première ligne L1 est passée à "1". Puis les six colonnes sont vérifiées pour savoir si une touche a été activée. Sa colonne sera alors dans l'état "1". Si aucune des touches de la ligne n'est cliquée, les six colonnes sont dans l'état "0". Puis c'est la ligne suivante qui passe à l'état "1". Les six colonnes sont ensuite balayées à nouveau. Dès que l'une des colonnes est lue dans un état "1", conformément à la Fig.1 il y a codage du caractère. Par défaut si aucune touche n'est activée, le caractère sera '@'.

### Fiche n°16.

#### Utilisation de P02\_Validation du CLAVIER.ino

Dialoguer avec ce démonstrateur impose d'utiliser la ligne série USB du Moniteur de l'IDE (57600baud.) pour afficher les touches cliquées, et que le clavier soit branché comme indiqué sur la Fiche n°1. Si le circuit imprimé du clavier n'est pas encore réalisé, il sera simulé, comme montré sur la Fig.1 utilisant une plaque à essais. À sa mise en service P02 affiche sa version sur le moniteur et le mode COMMANDE sur OLED. Chaque fois que la touche '1' sera cliquée on inversera le mode. En CRYPTAGE la LED triple clignote en vert. Frapper une quelconque touche provoque son affichage dans la fenêtre du Moniteur. Un Menu de BASE est en place pour le mode COMMANDE. Les seules commandes valides sont actuellement le '1' et la lettre 'B' qui valide ou suspend le Bruiteur. L'état de cette option est alors listé dans la fenêtre du Moniteur. Si c'est l'option OUI qui est active, tout caractère autre que '1' ou 'B' n'étant pas encore affecté engendrera un Bip d'alerte. En mode CRYPTAGE seul le chiffre '1' est actuellement valide. Tout autre touche engendrera un BIP d'erreur si





## Fiche n°15.

### Utilisation de P01\_Programme\_de\_base.ino

Dialoguer avec ce démonstrateur impose d'utiliser la ligne série USB du **Moniteur de l'IDE** pour dialoguer avec la carte Arduino NANO. Les échanges ne seront corrects que si sur le Moniteur la vitesse de transfert est bien sélectionnée à 57600baud. Les LEDS doivent être branchées comme indiqué sur la **Fiche n°1**. À sa mise en service **P01** affiche la date de sa version sur le **Moniteur** et le texte **BONJOUR** en double taille durant une seconde sur OLED. Puis un compteur rapide s'incrémente à la vitesse de la boucle d'attente d'un caractère au clavier dans le champ de saisie du **Moniteur**. *(C'est le rafraichissement de l'écran OLED qui ralentit cette boucle d'attente.)*

Par défaut le menu est en mode **COMMANDE**, cet état étant affiché en permanence sur l'écran graphique. Dans ce cas la LED triple clignote en bleu. Chaque fois que la touche '=' sera cliquée on inversera le mode. En **CRYPTAGE** la LED triple clignote alors en vert. Frapper un quelconque caractère provoque son affichage entre crochets dans la fenêtre du **Moniteur**. Toute lettre sera transformée en équivalent Majuscule. Les seules commandes valides autres que les lettres de l'alphabet sont actuellement le '=', l'**ESPACE**, '\*' et '&'.

- '\*' Force un BIP sonore si l'option est valide.
- '&' Déclenche un cycle tricolore sur la LED triple.
- '=' alterne entre le mode **COMMANDE** et le mode **CRYPTAGE**.
- La LED triple clignote en bleu si commande et vert si cryptage.
- '**ESPACE**' valide ou suspend le **Bruiteur**. L'état de cette option est alors listé dans la fenêtre du **Moniteur**.
- Si c'est l'option OUI qui est active, toute frappe autre qu'une lettre ou les quatre caractères de commande engendrera un Bip d'alerte.
- Dans les deux modes l'effet des consignes est identique.
- Toute lettre majuscule ou minuscule allumera la lampe à incandescence simulée les LEDs oranges durant 0,3 seconde, la répartition sur les sorties des deux multiplexeurs étant précisée dans la **Fiche n°9**. *(Voir aussi la Fiche n°10.)*

## Fiche n°14.

### Codage du CLAVIER à 30 touches.

L'agencement matériel électrique a été conçu pour faciliter au maximum le transcodage en procédant par un simple calcul, au lieu d'une solution plus classique avec utilisation d'un tableau. La technique consiste à organiser la "grille" comme présenté sur la Fig.1 de façon à aboutir à celle de la Fig.2 par une opération élémentaire :

$$\text{Caractere} = ((5 \times C_i) + L_i) - 6.$$

Fig.1

		D3	D4	D5	D6	D7	D8	
		C1	C2	C3	C4	C5	C6	L
L5	D13	E	J	O	T	Y	4	5
L4	D12	D	I	N	S	X	3	4
L3	D11	C	H	M	R	W	2	3
L2	D10	B	G	L	Q	V	1	2
L1	D9	A	F	K	P	U	Z	1
	C	1	2	3	4	5	6	

Petit  
CLAVIER

1 : Mode L2  
2 : NON L3  
3 : OUI L4  
4 : ESC L5

Fig.2

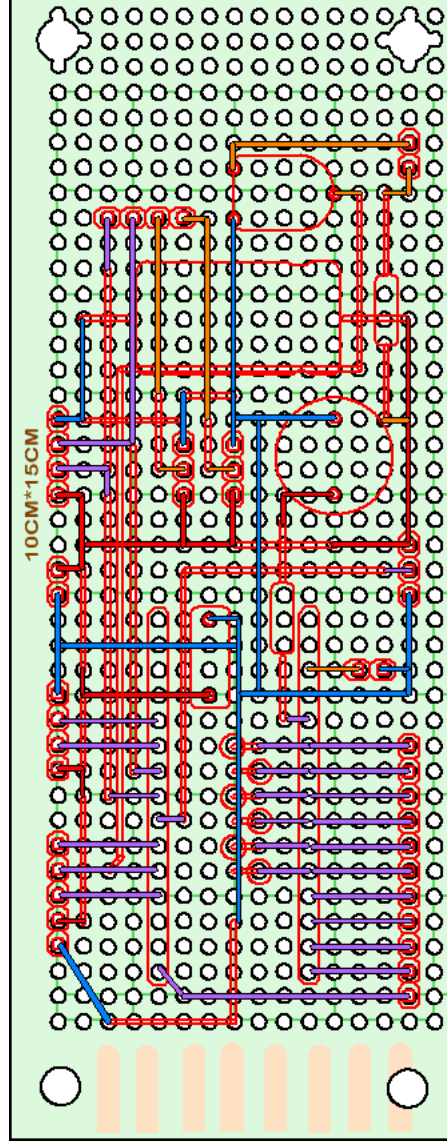
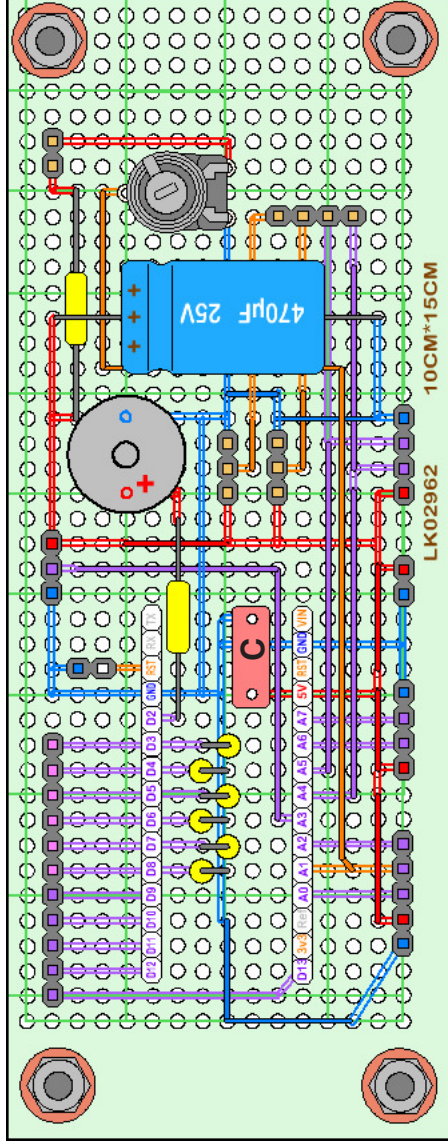
		D3	D4	D5	D6	D7	D8	
		C1	C2	C3	C4	C5	C6	L
L5	D13	4	9	14	19	24	29	5
L4	D12	3	8	13	18	23	28	4
L3	D11	2	7	12	17	22	27	3
L2	D10	1	6	11	16	21	26	2
L1	D9	0	5	10	15	20	25	1
	C	1	2	3	4	5	6	

EXEMPLE pour L3, C4 :  $\text{Caractere} = ((5 \times 4) + 3) - 6 = 17$ . Avec le **traitement particulier pour les quatre chiffres** le codage ne prend que 68 octets de programme et un seul en mémoire dynamique. On observe que sur la Fig.2 chaque calcul donne une valeur ordonnée correspondant à la place de la lettre dans son ordre alphabétique. Transformer **Caractere** en une lettre équivalente est élémentaire avec l'instruction **TOUCHE = char(Caractere + 65)** dont la ligne d'instruction est incluse dans les soixante huit octets d'occupation de la zone programme.

## Fiche n°17.

### Le circuit imprimé du microcontrôleur.

Prévu volontairement "trop long" cette petite carte électronique ménage une libre pour pouvoir si nécessaire accueillir quelques composants supplémentaires pouvant éviter d'avoir à créer un circuit imprimé de plus pour les intégrer au projet.

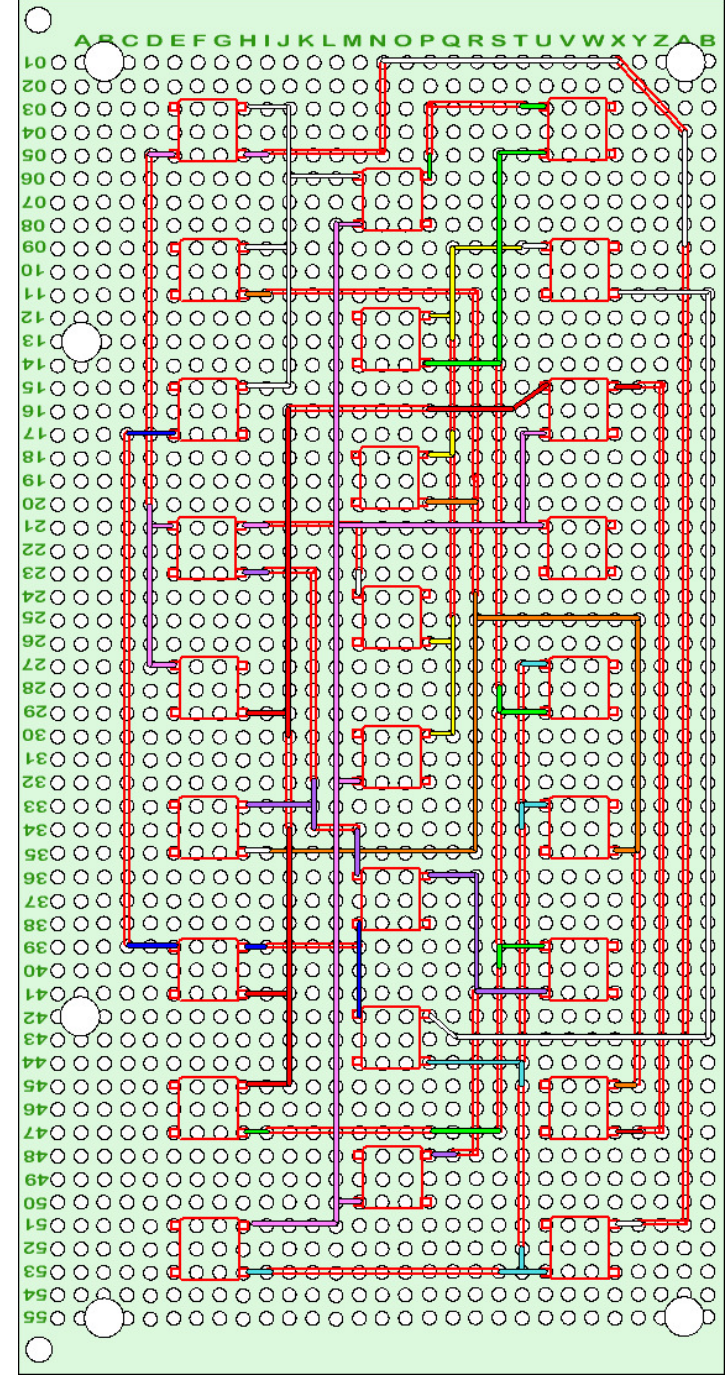


Page 10

## Fiche n°20.

### **Circuit imprimé du CLAVIER vu coté pastilles cuivrées.**

Ce circuit imprimé regroupant les 26 touches du CLAVIER vu coté composants. Les cinq lignes et les six colonnes de la matrice sont reliées directement au petit circuit imprimé de la Fiche n°22.

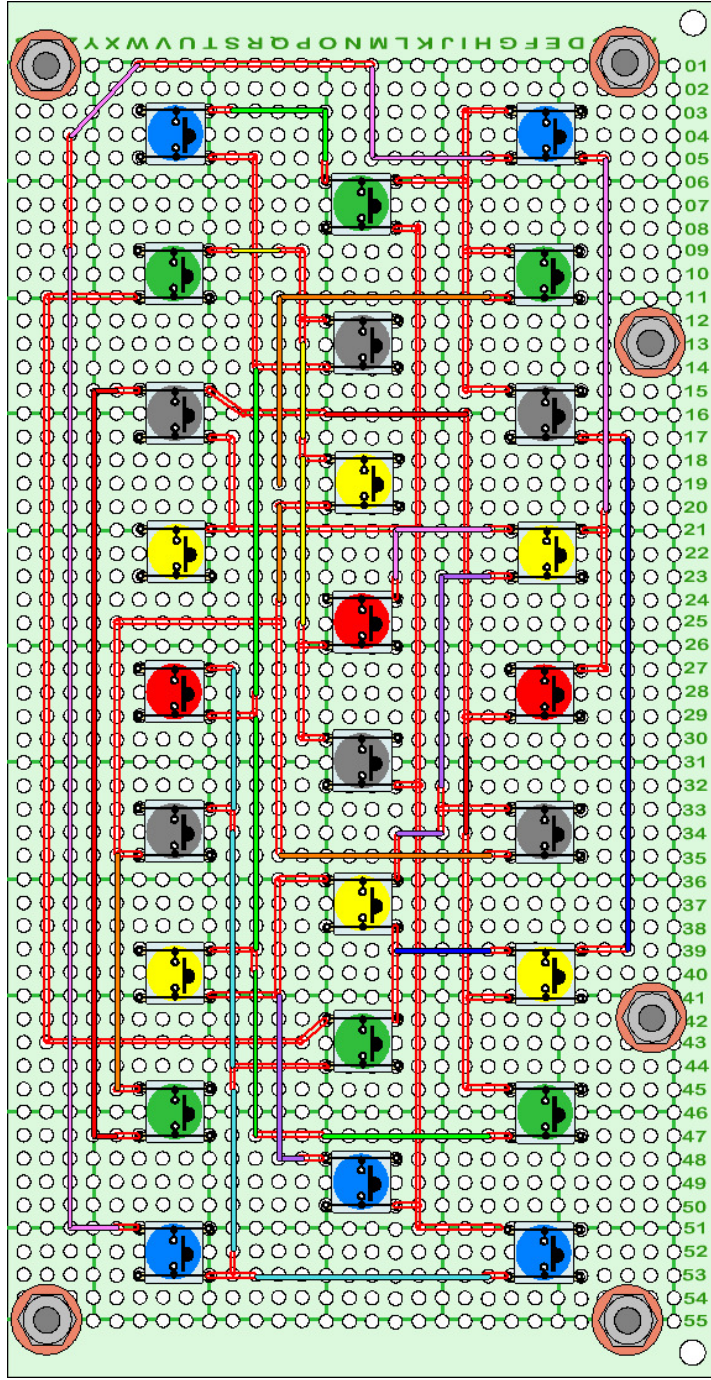




## Fiche n° 19.

### Circuit imprimé du CLAVIER vu coté composants.

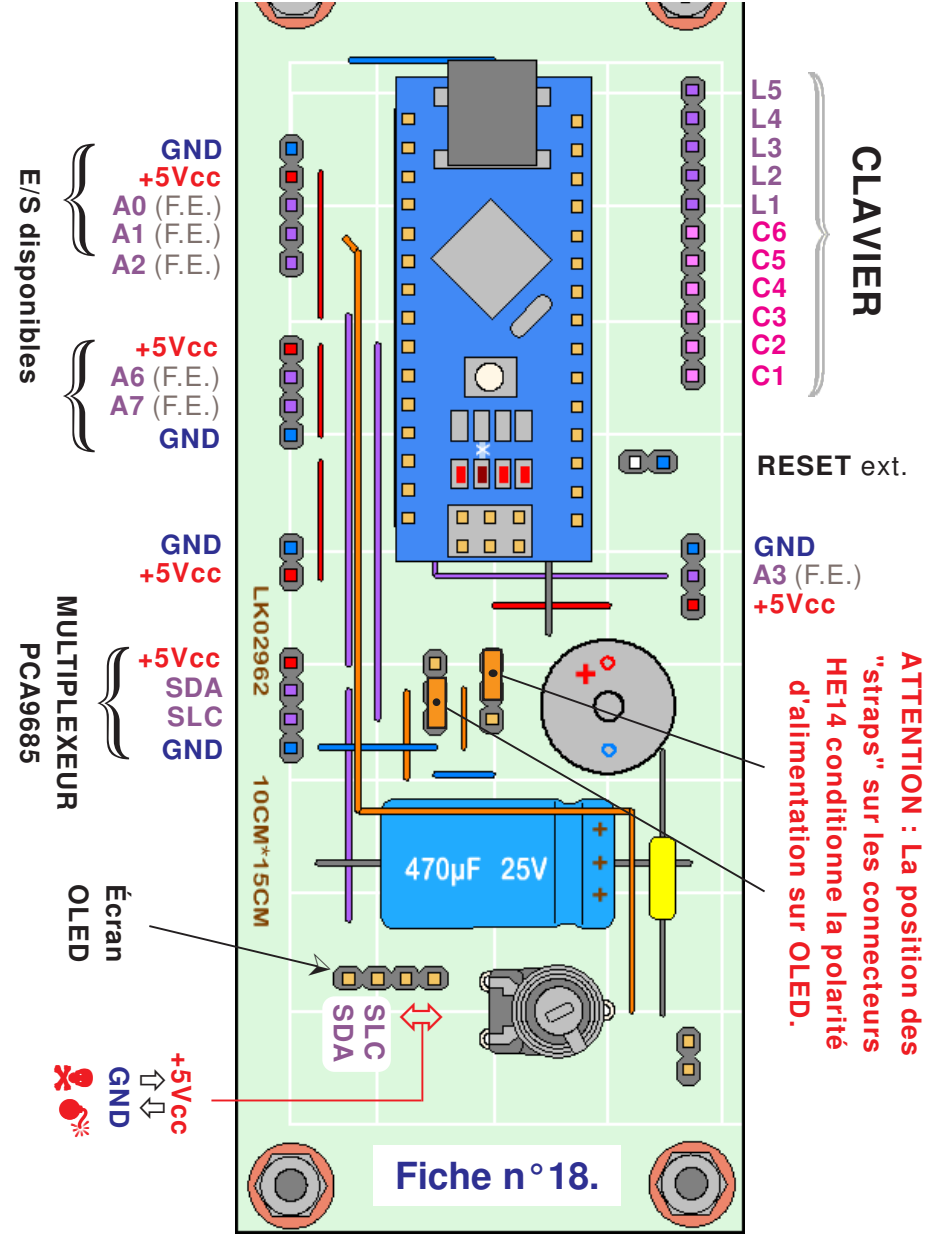
Ce circuit imprimé regroupant les 26 touches du CLAVIER vu coté composants. Les cinq lignes et les six colonnes de la matrice sont reliées directement au petit circuit imprimé de la Fiche n° 22.



Page 11

### Branchements extérieurs du circuit principal.

Positionner les deux "straps" et vérifier la polarité avant de brancher la ligne qui se branche sur l'afficheur OLED.



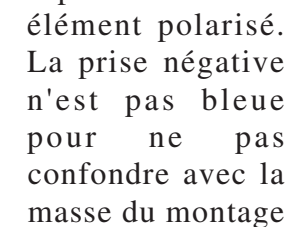


## Répartition des touches et branchement du CLAVIER.



**TESTEUR DE CONTINUITÉ ET DE LED.**

Les prises pour tester la continuité sont colorées pour repérer le sens du courant dans l'éventualité d'une manipulation sur un élément polarisé.



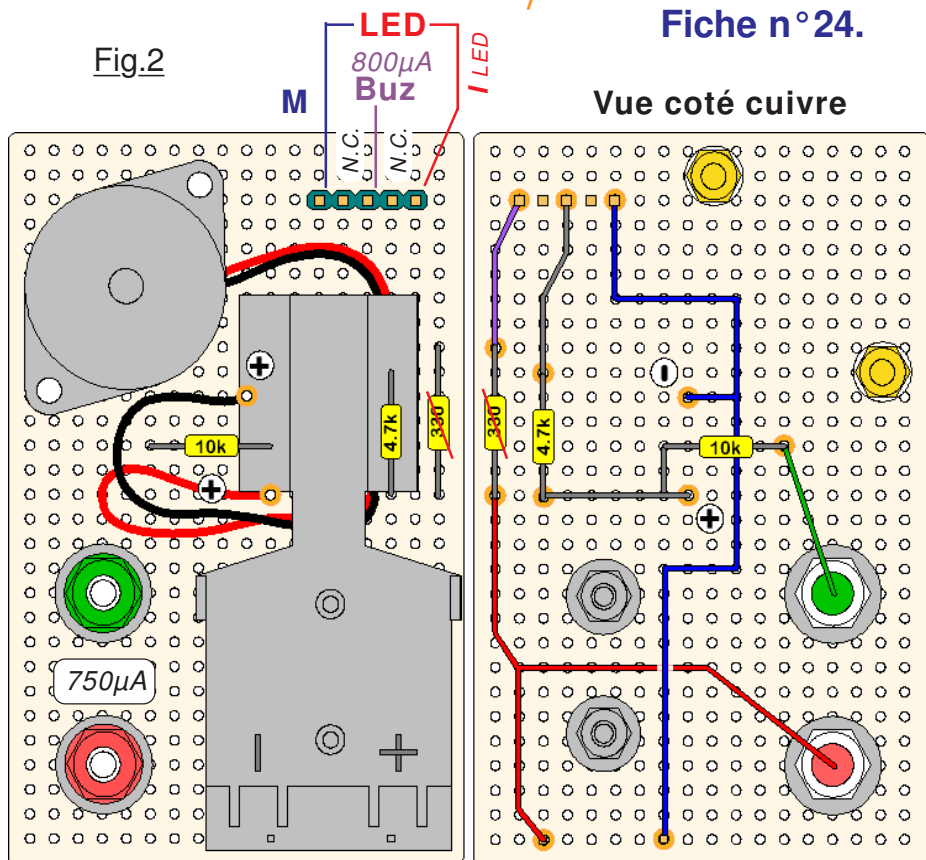
Accumulateur 8,4Vcc  
rechargeable ou pile  
9Vcc de type LR61

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 courant puisque sur un court circuit des deux prises le courant reste inférieur à un milli

Couleur	<i>I LED</i>
Rouge	<i>2 mA</i>
Verte	<i>1.8 mA</i>
Jaune	<i>1.9 mA</i>
Blanche	<i>1.4 mA</i>
Bleue	<i>1.6 mA</i>
Court circuit	<i>0.9 mA</i>

ampère. Si le dispositif doit servir d'élément sonore pour Arduino, l'entrée **Buz** sera reliée directement à l'une des quatorze sorties binaires **D0** à **D13** du microcontrôleur sans oublier de réunir les masses. La résistance de limitation de courant est intégrée à ce

Fig.2



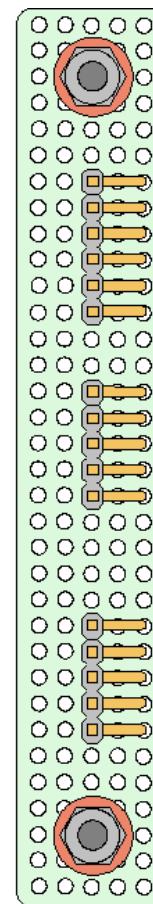
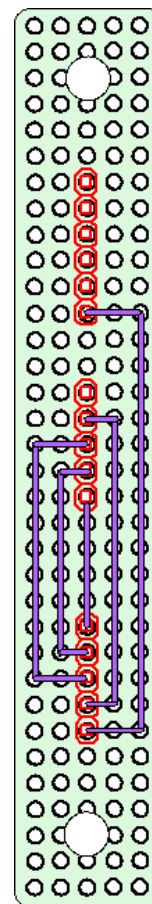
petit montage, et le buzzer ne drainera que  $800\mu\text{A}$  quand la sortie d'Arduino passera à "1".

**A**utonome par son accumulateur de  $9\text{Vcc}$ , ce dispositif permet facilement de tester une LED dans des conditions nominales qui généralement avoisinent les  $2\text{mA}$ . 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<sub>LED</sub>*** 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  $800\mu\text{A}$* ) 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 donne le dessin du petit circuit imprimé qui supporte tous les composants, la pile de  $9\text{Vcc}$  n'étant pas montrée dans son support.

### Circuit imprimé pour les connecteurs de liaison.

Les onze lignes du toron qui vont sur le circuit du CLAVIER sont soudées coté pastilles, y compris les cinq fils rigides pour les pontages.

Fiche n°22.



C1  
C2  
C3  
C4  
C5  
C6

D3  
D4  
D5  
D6  
D7  
D8

L1  
L2  
L3  
L4  
L5

D9  
D10  
D11  
D12  
D13

L5  
L4  
L3  
L2  
C6

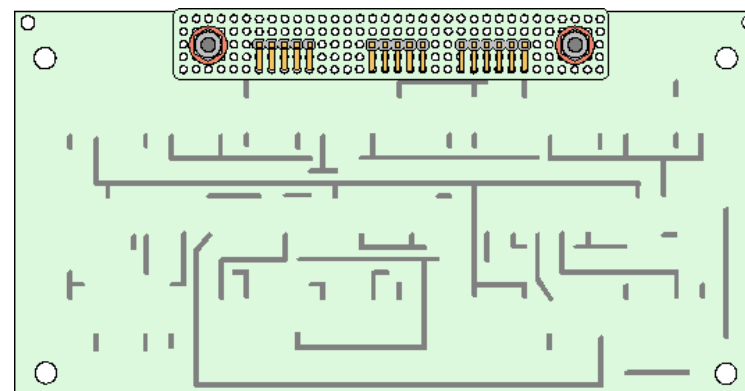
D13  
D12  
D11  
D10  
D8

Vers le C.I. de la carte ARDUINO.

Vers le C.I. du  
clavier secondaire.



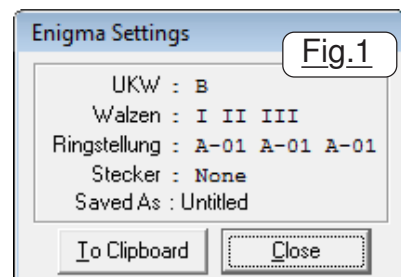
Ce circuit imprimé supporte les connecteurs HE14 qui vont vers le microcontrôleur et vers le clavier secondaire est situé coté cuivre du C.I. du CLAVIER principal les broches dirigées vers ce dernier.



## Fiche n°27. Une Énigma fiable de référence. (1/6)

Pour tester le comportement de notre réplique, il nous faut impérativement disposer d'une vraie machine Énigma. Une telle codeuse n'est pas forcément dans nos moyens, car les prix de ces antiquités sont absolument astronomiques. Aussi nous allons utiliser un simulateur reproduisant avec rigueur le comportement de l'unité réelle. Il en existe de nombreuses versions sur Internet toutes présentant des fonctionnements fiables. Comme il faut bien en choisir une, nous allons utiliser celle de : <https://www.ciphermachinesandcryptology.com/en/enigmasim.htm>

Facile à installer, lorsque l'on active ce simulateur, nous obtenons l'affichage qui ressemble à celui de la Fig.3, avec au départ l'initialisation par défaut de la Fig.1 que nous allons utiliser pour tester la réplique et effectuer les manipulations proposées dans le livret **UTILISER ÉNIGMA.pdf**. Pour quitter cette application il suffit de cliquer en **16** avec le bouton gauche de la souris.



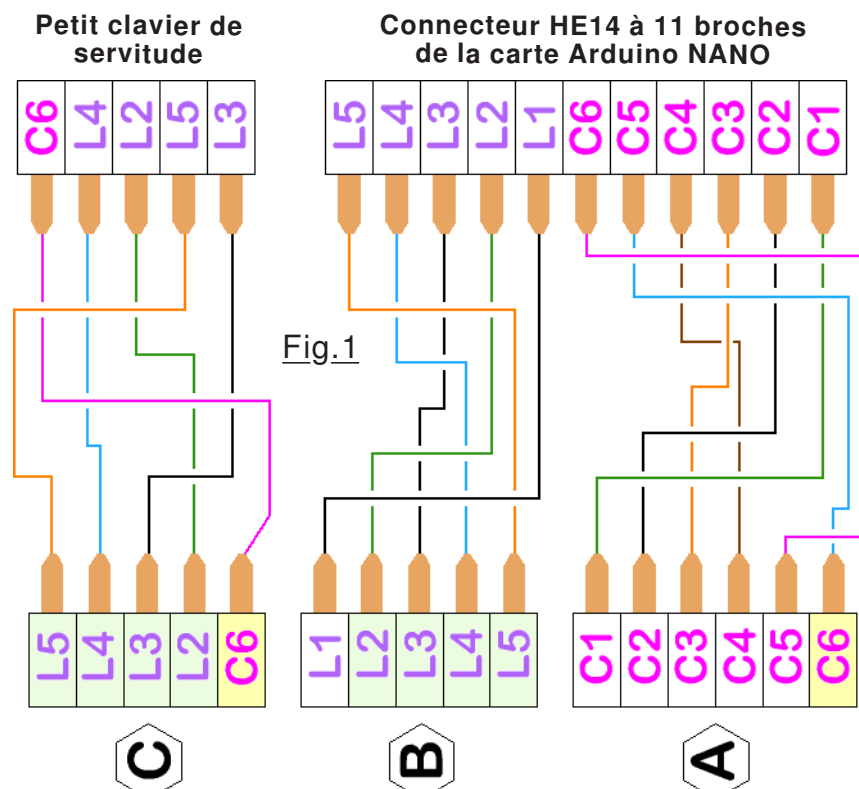
(Le programme nous demande si l'on désire sauvegarder la configuration si cette dernière a été modifiée.) Cliquer en **1** ouvre le menu de la Fig.2 avec l'affichage de la Fig.1 si on valide l'option jaune, avec possibilité de sauvegarder ou de recharger une configuration avec les commandes bleues, voir de sortir par la consigne verte etc. Quand on clique sur une touche du clavier **12** comme en **11** par exemple, elle devient plus petite simulant son enfoncement. Le **Rotor** de droite en **3** tourne en faisant un petit bruit caractéristique, puis l'une des ampoules du tableau **9** en **10** par exemple s'illumine en fonction du cryptage. Lorsque l'on relâche le bouton de la souris, alors elle s'éteint. Quand le **Rotor** de droite a effectué une rotation, il entraîne en **8** son voisin du centre. Puis, ce sera le tour en **7** du **Rotor** le plus à gauche. Machine "fermée" dans l'état de celle de l'exemple on peut

Fig.2

Clipboard
Auto Typing
View Key
Load Key
Save Key
Clear Key
Gallery
About
Help
Exit

## Fiche n°26. Câblage des deux claviers.

Comme le précise le schéma Fig.1 sur la **Fiche n°3** le clavier secondaire dédié au dialogue Homme / Machine utilise les lignes **L2, L3, L4** et **L5** ainsi que **C6**. Pour minimiser le nombre de lignes filaires réunissant les divers modules électronique, les pilotages multiplexés **L2, L3, L4, L5** et **C6** ont été pontés en interne sur le dessous coté pastilles cuivrées du circuit imprimé qui supporte le connecteur du clavier principal pour être distribuées



Connecteurs HE14 du clavier principal

vers le petit clavier sur le circuit imprimé supportant l'afficheur OLED. Sur la Fig.1 les pontages réalisés sur le petit circuit imprimé qui supporte les trois connecteurs **A, B** et **C** sont coloriés en vert pastel et en jaune pour les mettre en évidence.



## Fiche n°25.

### Mesure de la tension +5Vcc sur Arduino.

Alimenté en interne par un accumulateur de type "bloc USB", en mesurer la tension durant le fonctionnement de la machine peut prévenir l'opérateur qu'il est temps de penser à recharger cette source d'énergie. Le modèle utilisé affiche lui même sur quatre LEDs l'état de charge de cette réserve d'énergie, mais ce ne sera pas forcément le cas chez toutes celles et ceux qui vont réaliser ce petit projet ludique.

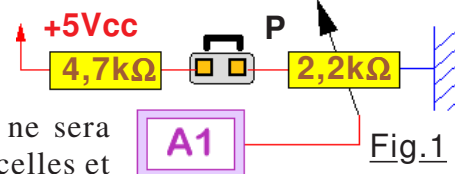


Fig.1

Comme par défaut l'instruction `analogRead()` retourne une valeur comprise entre 0 et 1023 lorsque la tension mesurée sur le **CAN** varie entre 0 et +5Vcc, la valeur affichée serait erronée et une baisse de tension ne serait pas détectée. Il faut impérativement prendre une tension interne stabilisée et précise comme référence pour le **CAN** : `analogReference(INTERNAL)`;

Mais cette tension de référence fait exactement 1,1V. Il faut impérativement ramener le +5Vcc d'Arduino à cet ordre de grandeur ce que fait le schéma de la Fig.1 et *parer les fluctuations* inexorables affectant ce type de mesurage. Avec les valeurs de ce schéma la tension sur le "strap" fait environ +1,6V.

Parer les fluctuations consiste à totaliser 1000 mesures avec le **CAN** et à en faire la moyenne :

```

① for (int l=0; l < (1000); l++) {
②     TENSION = TENSION + analogRead(MESURE);}
③ TENSION = TENSION / 1000;
④ TENSION = map(TENSION, 0, 1024, 0, 53);
⑤ U = float(TENSION) / 10;
```

L'instruction ① compte les mesures sachant qu'avant de rentrer dans cette boucle on commence par `TENSION = 0`. En ② on effectue les mesures et on additionne dans le `unsigned long TENSION`. En ③ on calcule la moyenne. En ④ on convertit la plage maximale de [0 à 1023] en [0 à 53] car **1023 est obtenu pour exactement 5,3V**. Enfin le float `U` reçoit la valeur mesurée qui sera affichée avec une décimale. (Pour l'étalonnage voir `Mesurer_U.ino`)

## Fiche n°28. Une Énigma fiable de référence. (2/6)



Fig.3

manuellement positionner à convenance chaque **Rotor** en cliquant soit en **2** sur le haut de la couronne dentée, soit en **5** vers le bas des pétales de la "marguerite". Cliquer sur le verrouillage **6** nous engage dans une promenade technique de découverte des éléments de la machine. C'est en cliquant sur le verrouillage **4** que l'on ouvre le plateau du dessus pour installer et configurer le **Brouilleur**. Commodité apportée par la version virtuelle, le fait de cliquer sur la serrure **13** ouvre la zone **14** dans laquelle vont s'écrire le texte d'origine et celui crypté avec formatage par groupe de cinq lettres. Par exemple en **15** on a sur la ligne du haut la première lettre 'A' frappée au clavier en **11** et en dessous son codage en 'B'. Cet artifice remplace la feuille de papier sur laquelle l'opérateur

## Fiche n°29. Une Énigme fiable de référence. (3/6)

inscrirait son message avant de le transmettre en morse, ou au contraire de le décoder. Il est évident que ce simulateur comme le notre est prévu pour pouvoir changer les **Rotors** ainsi que leur initialisation et sélectionner le **Rélecteur** utilisé. En cliquant dans la zone des prises du tableau des fiches vu ici par le dessus en **17** que l'on peut combiner les dix inversions filaires des **Fiches croisées**.

### ➤ Les informations sur Internet.

**S'**il est un thème qui a fait couler beaucoup d'encre sur son sujet, c'est bien Énigma. Que ce soit dans les archives papier ou en "dématérialisé" sur la toile, les articles décrivant cette machine ainsi que son historique sont légion. Il faut toutefois "pondérer" ces explications et parfois les contextualiser. Par exemple, quand cette codeuse est décrite, on peut lire, *et dans plusieurs sources*, que : **"Comme la roue de droite tourne à chaque frappe sur le clavier, la même lettre sera différente durant 26 saisies."** Cette affirmation péremptoire n'est pas totalement exacte. S'il est vrai qu'au cours de ces rotations la combinatoire de substitution change, il peut arriver que la lettre soit codée de façon identique plusieurs fois durant un cycle. Considérons la Fig.4 issue

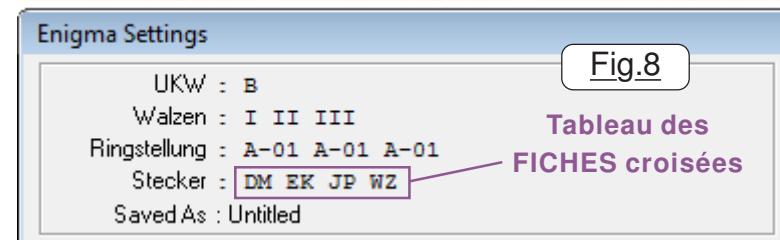
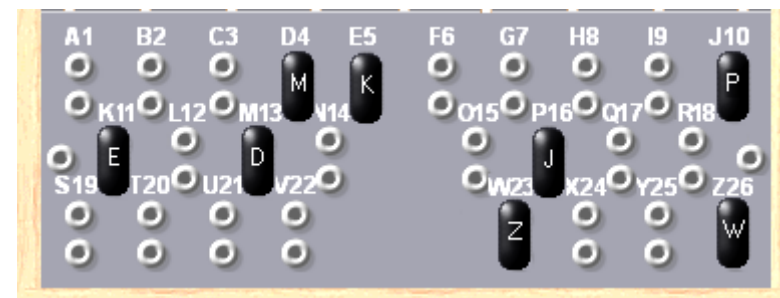


du simulateur que nous avons adopté pour sa fiabilité et sa simplicité de mise en œuvre. (Ce dessin est compacté pour ne montrer que les zones pertinentes.) Initialement les trois **Rotors** sont sur la position **A**. Dans cet exemple on a frappé vingt fois la lettre 'A', c'est à dire que le **Rotor** n'a pas encore fait un tour complet et n'a pas entraîné le **Rotor** central. Pourtant on constate qu'à de nombreuses reprises on retrouve un cryptage déjà effectué.

## Fiche n°32. Une Énigme fiable de référence. (6/6)

### ➤ Initialiser le tableau des Fiches croisées.

**B**ien que sur notre réplique ou sur l'[Énigma] de référence on puisse n'installer que quelques Lignes croisées, voir aucune, historiquement les coffret des machine réelles contenaient dix jonctions souples "filaires doubles" et toutes étaient branchées dans les armées durant le conflit. Considérons le montage graphique

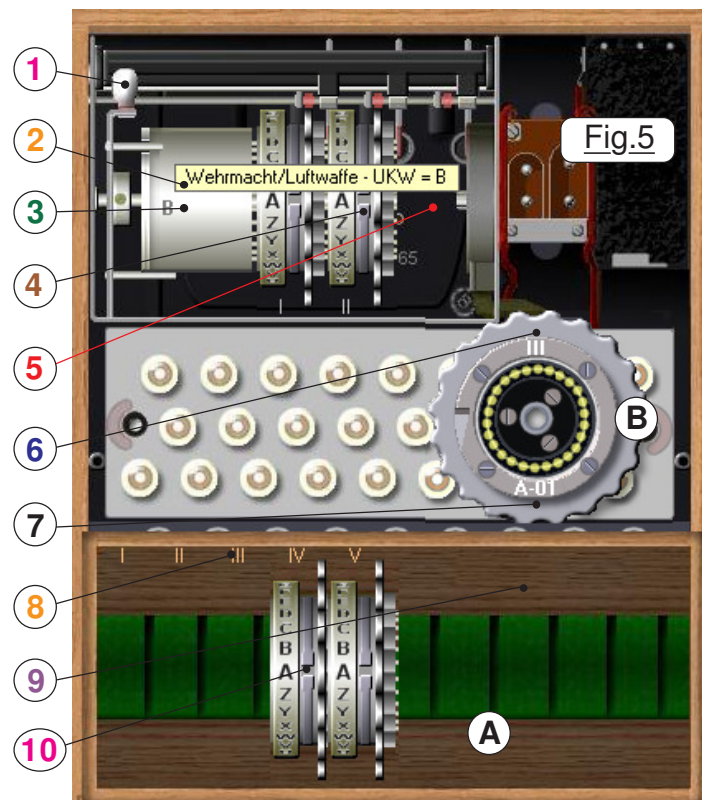


de la Fig.8 avec mise en évidence de la mise en place de quatre **FICHES croisées**. Cliquer en **17** de la Fig.3 pour ouvrir le tableau des **FICHES croisées**. Pour établir une liaison électrique croisée sur cette chiffreuse virtuelle cliquer sur deux prises quelconques. Par exemple sur 'T' puis sur 'G'. Peu importe l'ordre dans lequel on installe la ligne.

*Pour ne pas cacher entièrement le tableau avec les fils de liaison électrique, ces derniers ne sont pas représentés. Seules les deux fiches sont affichées avec la "lettre croisée".*

Elles ont été placées strictement dans l'ordre **[EK]**, **[ZW]**, **[JP]** et **[DM]**. Pourtant, quand on clique sur la commande **View Key** on constate sur la Fig.8 qu'elles ne sont pas du tout listées dans l'ordre de création. Elles sont en fait automatiquement classées par ordre alphabétique trillé sur la première lettre de la paire associée.

## Fiche n°31. Une Enigma fiable de référence. (5/6)



Brouilleur. En cliquant sur l'onglet **Wiew Key** on fait afficher la fenêtre Fig.7 qui résume les paramètres de l'initialisation avec dans la zone jaune le nom du fichier .eni rechargé automatiquement si dans la zone repérée en bleu clair du menu de la Fig.2 on a déjà sauvegardé un fichier de type .eni. en quittant ce simulateur.

**ATTENTION, il faudra à chaque début de cryptage ou de décodage commencer par replacer manuellement les orientations des Rotors en face de la fenêtre** car cette combinaison n'est pas mémorisée.

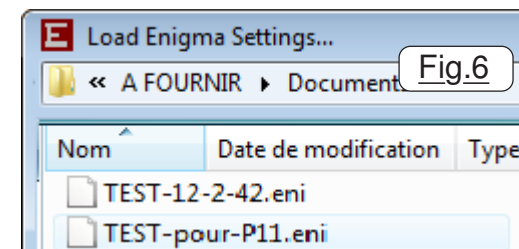
## Fiche n°30. Une Enigma fiable de référence. (4/6)

Les informations proposées dans les exposés ne sont pas franchement inexacts, mais elles sont forcément "simplifiées" pour des raisons pédagogiques. Donc, parfois il ne faudra pas s'étonner d'une divergence entre ces explications et les résultats obtenus sur des simulateurs fiables reproduisant exactement le comportement de la machine d'origine.

### ► Modifier la configuration de l'[Enigma] de référence.

Comparer le comportement des deux entités informatiques suppose d'**avoir une initialisation identique sur les deux codeuses**. Cet impératif nous oblige à voir comment procéder sur [Enigma]. Un moyen radical d'effectuer un test significatif consiste à chambouler intégralement la combinatoire d'une initialisation en adoptant un joyeux brassage des divers paramètres possibles. Commencer par cliquer sur le verrouillage 4 de la Fig.3 pour ouvrir le plateau du dessus et accéder comme sur la Fig.5 aux éléments internes de la chiffreuse. Cliquer n'importe où sur l'un des Rotors en 4 le dépose du Brouilleur et le place en B. C'est dans cette configuration que si l'on clique en 6 ou en 7 on en modifie à convenance l'Indexation interne. En cliquant dans le coffret de rangement en 9 le Rotors déposé en B est

automatiquement remplacé dans son logement repéré en 8. Pour installer un Rotor sur le Brouilleur on procède à l'inverse. On clique sur ce dernier en 10 pour le placer en B. Dans cette position on l'Indexe en interne avec 6 et 7. Puis on clique dans un emplacement libre en 5 pour l'installer sur le Brouilleur. C'est dans la zone restreinte 3 que l'on choisit le type de codeuse et le Réflecteur installé. Bien faire attention à sélectionner **Wehrmacht/luftwaffe - UKW = B** ou **C** indiqué en 2 en fonction de votre choix. On peut alors refermer le plateau en cliquant en 1 et orienter les Rotors à convenance. Noter au passage que l'on ne peut pas refermer le plateau s'il n'y a pas trois Rotors installés sur le





## Fiche n°35. Logement des données en EEPROM.

Les données relatives au codage des éléments du **Brouilleur** sont implantées en haut de l'espace mémoire et commencent à l'adresse décimale **712**. Les données de sauvegarde de l'INITIALISATION de la machine vont de **681** à **711**. À partir de **683** on trouve le codage des **ROTORS** décrits dans l'ordre **Gauche**, **Centre** et **Droite** sur trois octets qui contiennent successivement :

- Le numéro du **ROTOR** installé. (*I à V.*)
- L'**Indexage** interne de la **Bague extérieure**. (*A à Z.*)
- L'orientation initiale du **ROTOR** sous la fenêtre. (*A à Z.*)

ADRS	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0672	...	...	...	...	...	...	...	...	...	10	B	01	A	A	02	A
0688	A	03	A	A	A	B	C	D	E	10 FICHES croisées						K L
0704	M	N	O	P	Q	R	S	T	04	09	10	02	07	01	23	09
0720	13	16	03	08	02	09	10	18	07	03	09	22	06	13	05	20
0736	04	10	20	21	22	03	22	24	25	08	13	16	17	19	16	23
0752	24	04	17	06	00	18	23	13	17	19	16	10	00	08	01	07
0768	14	03	11	13	15	18	01	22	10	06	24	13	00	15	07	20
0784	21	03	09	24	16	05	00	08	13	25	21	17	11	04	23	18
0800	19	25	02	06	10	05	00	15	12	20	13	02	16	11	23	19
0816	01	02	03	04	05	06	22	08	09	10	13	10	13	00	10	15
0832	18	05	14	07	16	17	24	21	18	15	19	25	04	24	11	23
0848	12	22	08	21	10	20	09	00	11	18	08	17	05	16	02	16
0864	21	13	16	13	04	17	12	18	11	20	03	19	16	07	10	23
0880	05	20	09	22	23	14	01	13	16	08	06	15	24	02	07	24
0896	20	18	22	12	13	06	03	23	10	04	11	03	14	15	19	21
0912	09	25	16	08	02	17	10	06	21	24	25	14	02	03	13	17
0928	12	06	08	18	01	20	23	08	10	05	20	16	22	19	09	07
0944	04	11	16	01	22	08	19	17	24	00	25	10	15	03	06	25
0960	07	20	04	12	18	13	14	05	21	18	09	02	25	18	21	08
0976	17	19	12	04	16	24	14	07	15	11	13	09	05	02	06	26
0992	03	23	22	10	01	20	06	22	16	10	09	01	15	25	05	04
1008	18	26	24	23	07	03	20	11	21	17	19	02	14	13	08	12

## Fiche n°34. Implantation des textes en EEPROM.

ADRS	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000	C	O	M	M	A	N	D	E	D	e	u	x		R	O	T
0016	O	R	S	I	N	D	E	X		s	a	t	u	r	é	e
0032	s	!	F	e	n	é	t	r	e		I	I	I			
0048	I	V		V	R	o	t	o	r	:		B	a	g		
0064	u	e	:		D	é	b	u	t	:		[	]			
0080	[	R	é	f	l	e	c	t	e	u	r	:		N		
0096	b		F	i	c	h	e	s	:	d	e		G	a		
0112	u	c	h	e	d	u		C	e	n	t	r	e	d	e	
0128	D	r	o	i	t	e	i	d	e	n	t	i	q	u	e	s
0144	!	F	i	c	h	e		n	o	n		c	r	o	i	
0160	s	é	e	!		L	E	T	T	R	E		D	é	j	
0176	à	u	t	i	l	i	s	é	e	!		E	n	l	e	
0192	v	e	r		l	a	F	I	C	H	E		N	u	m	
0208	I	l		n	'	y	a		p	a	s		d	e	l	
0224	e	s		F	I	C	H	E	S	D	é	b	r	a	n	c
0240	h	e	r		t	o	u	t	e	s	C	o	n	f	i	g
0256	u	r	a	t	i	o	n	d	e		B	A	S	E	R	e
0272	s	t	i	t	u	e	r		l	e	s	t	r	o	i	s
0288	I	N	I	T	I	A	L	I	S	E	R		l	a	V	e
0304	r	s	i	o	n	:		m	a	c	h	i	n	e	O	
0320	r	d	o	n	n	e	r		l	e	s	E	n	l	e	v
0336	e	r		u	n	e	A	j	o	u	t	e	r	d	e	
0352	s	:		?	F	c	h		N	o	m	b	r	e		
0368	:			(	M	a	x	:		0	3	-	C		0	
0384	4	-	D		0	5	-	E		0	6	-	F	0	7	-
0400	G		0	8	-	H		0	9	-	I		1	0	-	J
0416	1	1	-	K		1	2	-	L		1	3	-	M		1
0432	4	-	N	1	5	-	O		1	6	-	P		1	7	-
0448	Q		1	8	-	R	1	9	-	S		2	0	-	T	
0464	2	1	-	U		2	2	-	V	2	3	-	W		2	4
0480	-	X		2	5	-	Y		2	6	-	Z	[	P	I	L
0496	E	-		T	A	S	]		o	c	t	e	t	s	.	
0512	G	é	n	é	r	e	r		u	n	e	c	o	n	f	i
0528	g	u	r	a	t	i	o	n	?	R	O	T	O	R		
0544	>	R	e	s	t	i	t	u	e	r		l	a	C	o	
0560	n	f	i	g	.	E	E	P	R	O	M	S	a	u	v	
0576	e	g	a	r	d	e	r		l	a	N	o	u	v	e	a
0592	u		C	R	Y	P	T	A	G	E	.	P	a	s	s	e
0608	r		e	n		M	o	d	e	T	E	X	T	E	s	
0624	u	r		U	S	B	A	f	f	i	c	h	e	r	l	a
0640	s		a	i	s	i	e		T	e	x	t	e	t	r	
0656	a	n	s	l	a	t	é	>	>	>		R	e	s	p	e
0672	c	t	e	r		l	e	s				..	B	..	A	A..

## Fiche n°33. Implantation des textes en EEPROM.

**Placer des textes en EEPROM devient rentable** en termes de taille occupée par le programme **dès que la chaîne fait deux caractères**. À son démarrage en activant le **Moniteur à 57600 baud P00\_Textes\_en\_EEPROM.ino** inscrit les données dans l'EEPROM puis balaye entièrement cette dernière et en affiche le contenu. En premier il liste les textes et l'on obtient un affichage

qui ressemble à celui de la Fig.1 pour lequel les mots sont listés dans l'ordre où ils sont inscrits. L'affichage du texte se fait dans la police de caractère de la fenêtre du **Moniteur** qui ne connaît pas les caractères accentués. La copie d'écran de la Fig.1 a donc été surchargée par les lettres mises en évidence en vert pastel. Dans l'état actuel du logiciel

```

COMMANDEDeux ROTORSINDEX saturées !Fenê
tre III IV VRotor : Bague : Dé
but : [] [Réflecteur : Nb Fiches : de
Gauchedu Centrede Droiteidentiques !
Fiche non croisée ! LETTRE Déjà utilisé
e ! Enlever laFICHE Num Il n'y a pas de
les FICHESDébrancher toutesConfigurationde
de BASERestituer lestroisINITIALISER la
Version : machineOrdonner lesEnlever une
Ajouter des : ?Fch Nombre : (Max :
03-C 04-D 05-E 06-F07-G 08-H 09-I 10-J
11-K 12-L 13-M 14-N15-O 16-P 17-Q 18-R
19-S 20-T 21-U 22-V23-W 24-X 25-Y 26-Z
[PILE - TAS] octets.G
rer uneconfiguration ?ROTOR >
Restituer laConfig. EEPROM
Sauvegarder la
Nouveau CRYPTAGE
Passer en ModeTEXTE sur USB
Afficherla saisie
Texte traduité
>>> Respecter les
    
```

Fig.1

l'EEPROM est entièrement saturée et il ne reste plus aucune cellule de disponible pour des données ou du texte supplémentaire.

**NOTE :** Chaque caractère accentués codés directement en EEPROM "est gratuit" et fait économiser 18 octets par rapport à un texte résidant directement dans le programme. Si l'intégralité des textes ne peut être logée en EEPROM faute de place disponible, **privilégier les mots accentués**.

## Fiche n°36.

### Interaction entre OLED et multiplexeur PCA9685.

Dans la documentation relative à l'écran OLED on apprend que son adressage sur la ligne **I2C** est soit 0x78, soit 0x7A exprimée en hexadécimal. Elle est donc différente de celles des deux multiplexeurs PCA9685. En théorie on ne devrait pas avoir de problème. Mais force est de constater durant le développement des démonstrateurs, qu'un problème épineux se présente régulièrement. On peut constater sur les images de la Fig.1 ci-contre que manifestement il y a une interférence. Les décalages latéraux mis en évidence par les lignes jaunes ne se produisent jamais si les deux multiplexeurs PCA9685 sont débranchés. Il y a donc visiblement une certaine "incompatibilité" entre les trois modules électroniques.

#### ➤ **Solution pour parer ce problème.**

Modeste en encombrement elle consiste à utiliser un petit relais rapide EC2-5NU qui isole la ligne **I2C** des deux multiplexeurs lorsqu'un affichage est effectué sur OLED. Le schéma utilisé est montré en Fig.2 sur lequel l'**INTERFACE** constituée de ce relais 2RT est précisée en Fig.1 de la **Fiche n°37**.

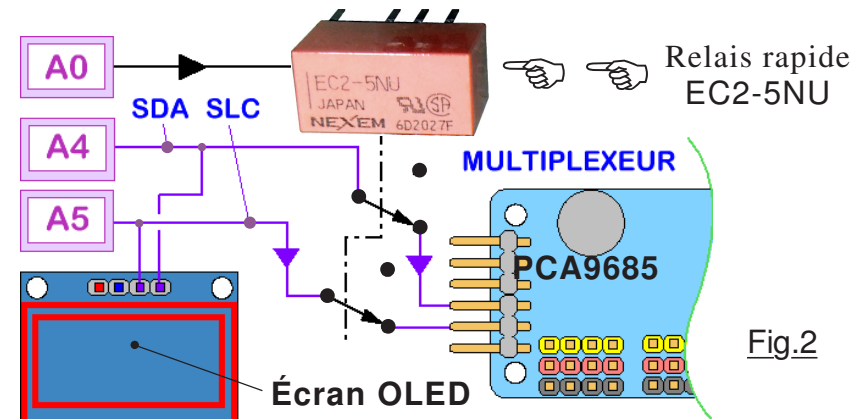
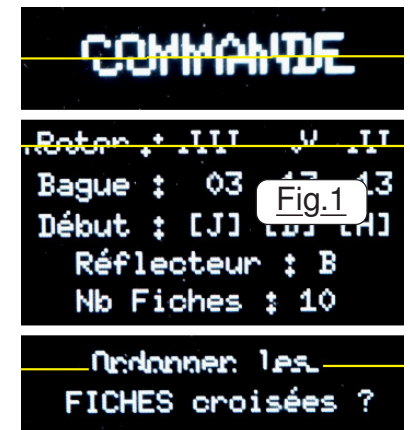
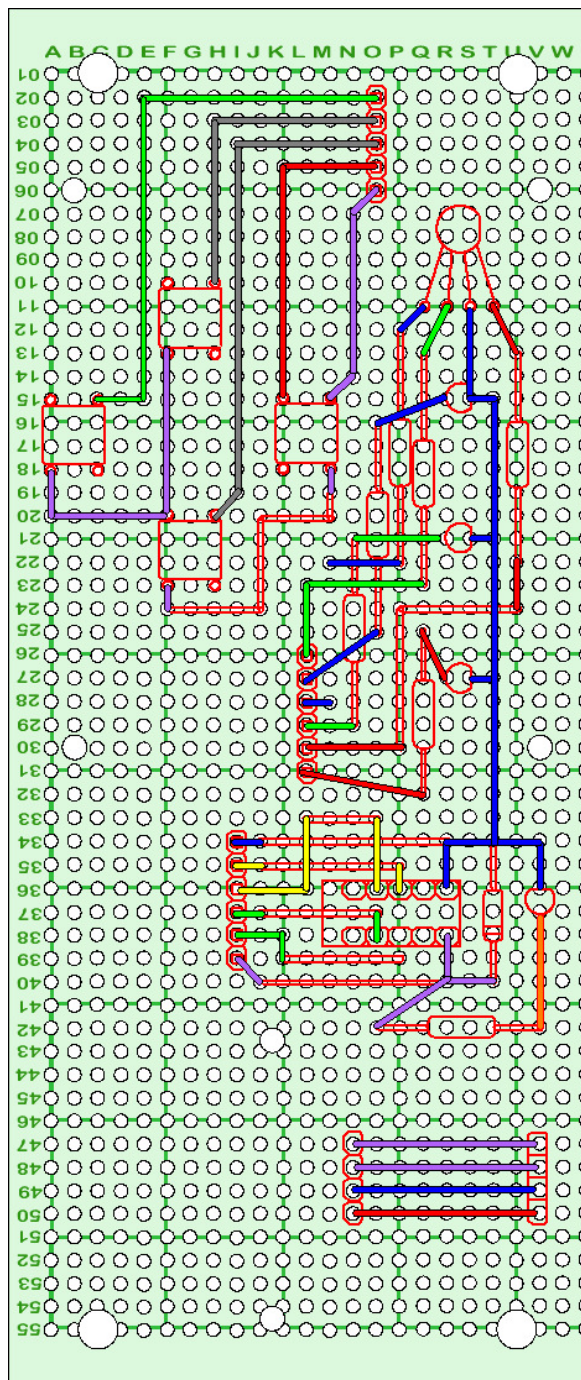


Fig.2



## Fiche n°39. Circuit imprimé du Petit clavier et de l'afficheur OLED.

### Circuit imprimé vu coré pastilles.



Sur ce circuit imprimé, l'intégralité des liaisons filaires qui sont sur le dessous est réalisée en petit fil de cuivre dénudé. Le circuit a été conçu pour éloigner au maximum le clavier de l'afficheur OLED. Tous les connecteurs HE14 sont soudés pour un départ horizontal des lignes et ainsi minimiser l'encombrement en hauteur. Ceux des LEDs et du clavier sont placés pour être décalés au mieux latéralement pour passer à côté des vis  $\phi$  M2 supportant la plaquette du clavier.

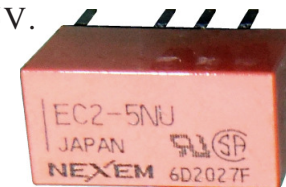
Page 20

## Fiche n°37.

### Le relais d'isolement KEMET : EC2-5NU.

Ces relais miniatures de type 2 RT fonctionnent en 5Vcc et présentent une impédance interne de  $178\Omega$ . Le courant nominal consommé est de l'ordre de 28mA avec une puissance dissipée d'environ 0,14W. Ce courant est compatible avec celui d'une sortie binaire Arduino, et A0 peut le piloter directement.

- Le relais passe au travail à environ 3,75V.
- Il passe au repos pour la tension  $\approx 1,3$ V.
- Durée de commutation : 2mS.
- Durée de passage au repos : 1mS.
- Courant de commutation nominal : 2A.

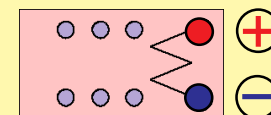


**Sans que ce soit précisé dans la documentation ... ils sont polarisés.**

Ce n'est absolument pas une "diode de roue libre" qui serait intégrée, car quelle que soit la polarité de la tension appliquée, la consommation reste strictement identique.

**On doit absolument respecter la polarité ci-dessus :**

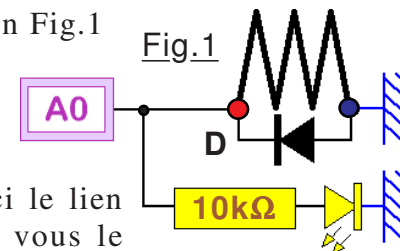
Vu par dessus.



Le schéma d'utilisation est fourni en Fig.1 avec en D la diode "de roue-libre".

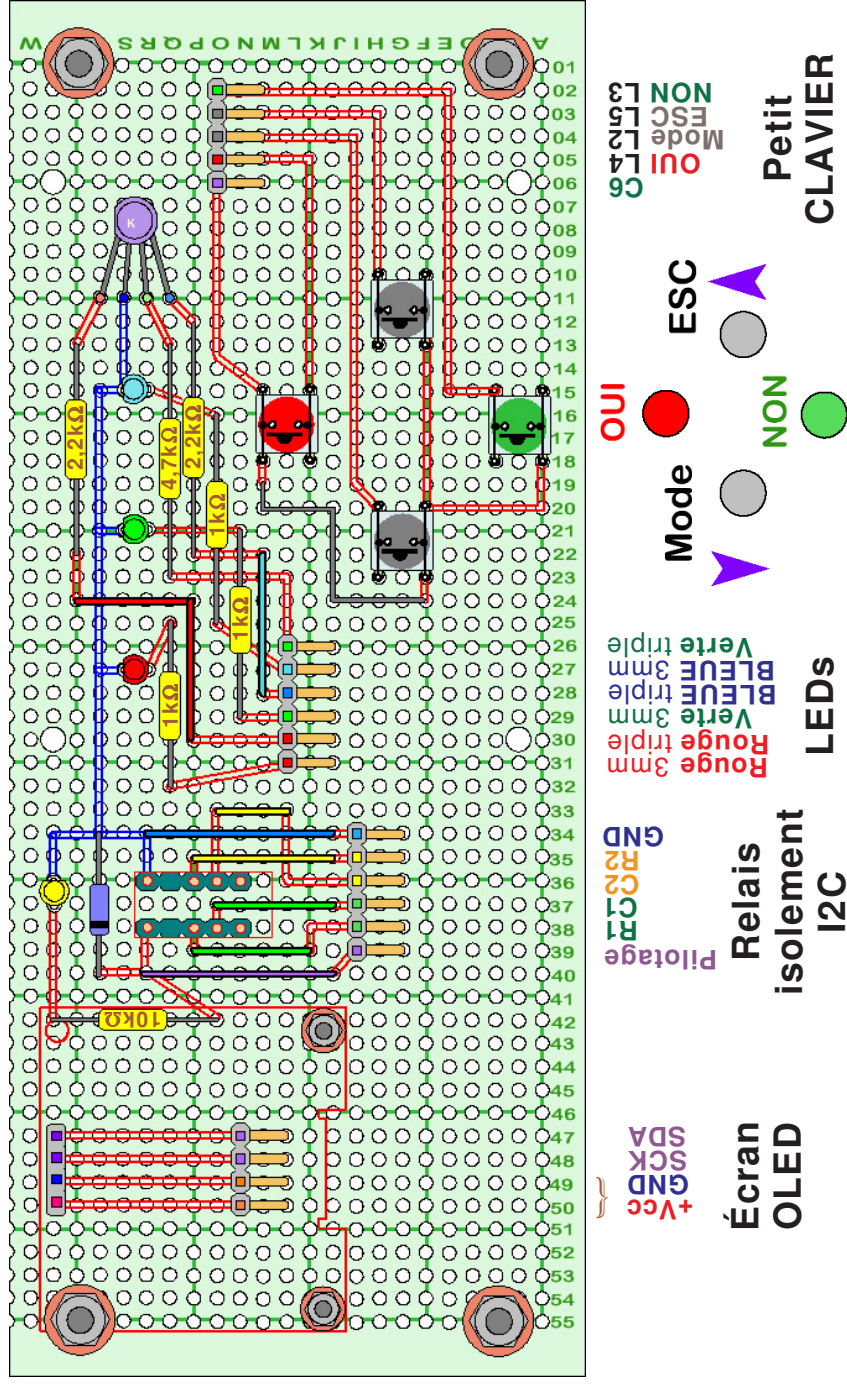
Tout petit relais miniature capable de commuter à cette rapidité conviendra, toutefois, voici le lien pour se procurer ce composant si vous le désirez. Pour pouvoir commander chez ce Farnell il faut au préalable s'inscrire sur le site en tant que client.

[https://fr.farnell.com/kemet/ec2-5nu/relais-de-signal-dpdt-250vac-220vdc/dp/2360823?CMP=KNC-GFR-GEN-KWL-High-Audience-Target-Imp-Share-14-Jul-21&mdv=s\\_dp&pid=50174494077/pd/wod/match&psd/product/gid/106438076612/paid/ud-9-8-7-9-8-0-1-3-9-0-2:d-s-a-915065063416&gclid=CjwKCAjwybyJBhBwEiwAvz4G71ibp3aWFXJ\\_1g-D5hGEJTbU2DHQOqsxTVa8nNxsKquqM9Bj\\_XeyRoCveQQAvD\\_BwE](https://fr.farnell.com/kemet/ec2-5nu/relais-de-signal-dpdt-250vac-220vdc/dp/2360823?CMP=KNC-GFR-GEN-KWL-High-Audience-Target-Imp-Share-14-Jul-21&mdv=s_dp&pid=50174494077/pd/wod/match&psd/product/gid/106438076612/paid/ud-9-8-7-9-8-0-1-3-9-0-2:d-s-a-915065063416&gclid=CjwKCAjwybyJBhBwEiwAvz4G71ibp3aWFXJ_1g-D5hGEJTbU2DHQOqsxTVa8nNxsKquqM9Bj_XeyRoCveQQAvD_BwE)





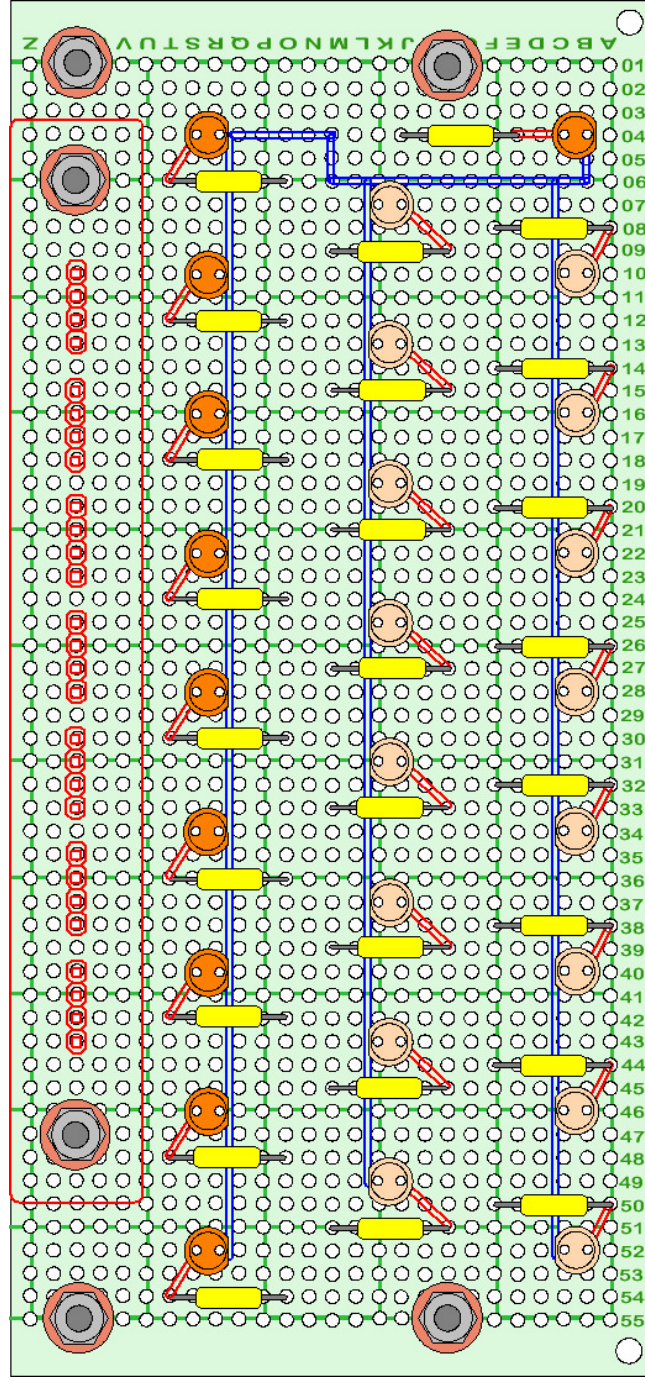
## Fiche n°38. Circuit imprimé du Petit clavier et de l'afficheur OLED.



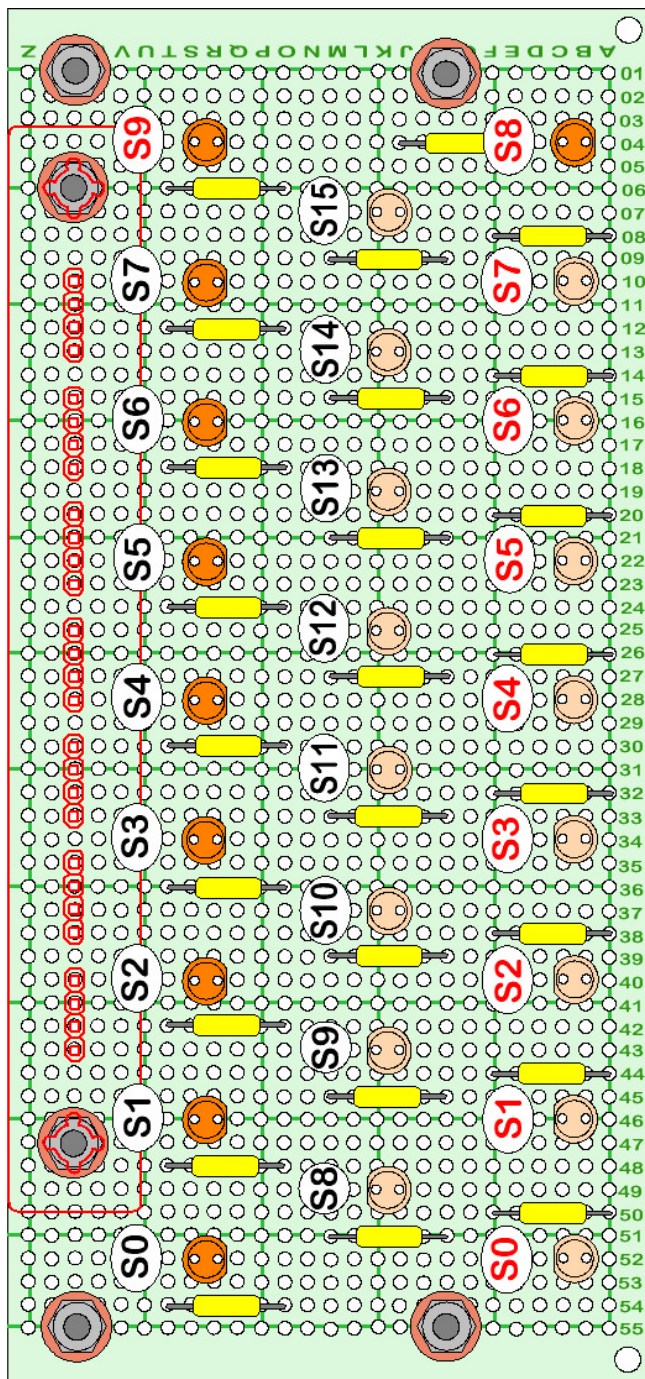
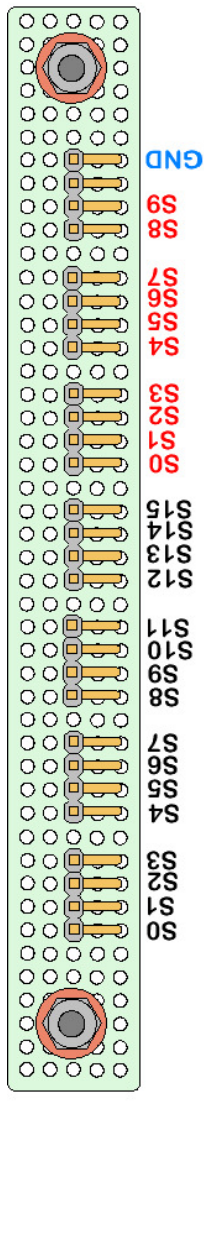
Page 21

## Fiche n°40. Circuit imprimé du Tableau des LEDs oranges.

Toutes les résistances de limitation du courant dans les LEDs sont de  $1k\Omega$ .  
Les extrémités "positives" des résistances sont réunies par des fils souples au circuit imprimé des connecteurs HE14 leur répartition étant précisée sur la Fiche n.41.  
**ATTENTION : Les LEDs ne sont pas toutes orientées de façons identiques.** Celles dont la cathode est dirigée vers le haut sont en orange clair alors que les autres sont en orange foncé.







## Fiche n°43.

pour arriver en **g**. (Noter au passage que la ligne noire représente un autre cheminement non utilisé dans cet exemple.) Dans le **Rélecteur** le circuit se poursuit en **h** pour retraverser le **Brouilleur** et ressortir en **i**. La ligne électrique correspondant au barillet de sortie arrive alors sur la prise double **S**. Hors la **Fiche croisée** pousse avec les broches **b** la plaquette **P** par les isolants **i**. Du coup la tension en **j** transite par la ligne croisée en **k**. Puis de **k** elle arrive alors en **i** sur le commun de l'inverseur **D** dont le contact repos alimente l'ampoule **D** en **m**. Lorsque l'opérateur relâche la touche **A** le mécanisme fait changer le **Rotor** de droite d'une position. **Du coup les croisements internes de la zone colorée en vert pastel changent de configuration**. Si on clique à nouveau sur la lettre **A**, le cheminement ne sera plus identique et la sortie du brouilleur se fera sur une autre ligne mais plus en **i**. Lorsque l'on frappera 26 fois la lettre **A**, chaque fois la transposition dans le **Brouilleur** sera différente. Un caractère **A** de plus et c'est le rotor central qui se décale d'une position. Les 26 lettre **A** qui suivent seront encore différentes. Pour qu'un cycle dans le brouilleur se répète, il faut  $26 \times 26 \times 26 = 17\,576$  caractères frappés sur le clavier. Tant que le message à encrypter ne dépasse pas les 17 576 caractères, il n'y a aucune répétition de l'alphabet de substitution.

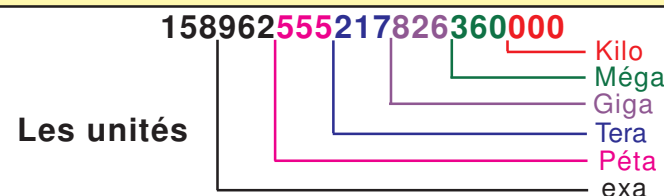
### ➤ La combinatoire explosive d'ENIGMA.

- 17 576 possibilités pour positionner initialement les trois **Rotors**.
- Trois **Rotors** / 5 à placer dans un ordre quelconque : 60 possibilités.
- Dix câbles permutent 20 lettres deux-à-deux. Cette technique engendre 150738274937250 permutations possibles.

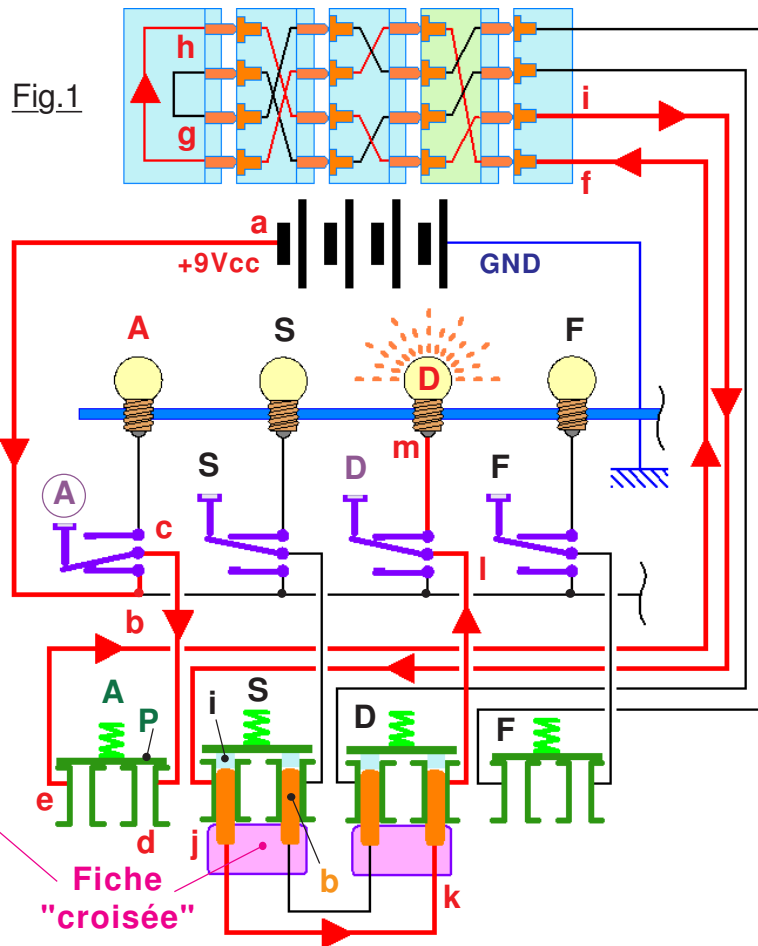
Le nombre total de combinaisons, (Égal au produit des trois combinatoires.) est donc de :

$$17\,576 \times 60 \times 150738274937250 = 158962555217826360000$$

clefs journalières initiales possibles.



## Fiche n°42. Fonctionnement électrique d'une ENIGMA.



Pour le codage initial de la journée envisagée, on a introduit les deux **Fiches croisées** du schéma de la Fig.2 respectivement sur la lettre S et la lettre D. Ainsi, une lettre **S** sera transcodée en **D** et une lettre **D** sera codée **S**. Supposons par exemple que l'opérateur radio frappe sur la touche **A**. L'électricité partant de **a** et arrivant en **b** transite par le contact travail de l'inverseur en **c**. Du plot en **d** sur la prise double **A** le courant traverse la palette **P** poussée sur les deux plots par le ressort symbolisé en vert clair. Partant du plot en **e** le courant se retrouve en **f**. Il traverse alors le **Brouilleur**

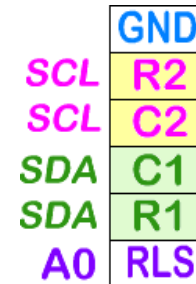
... / ...

Page 23

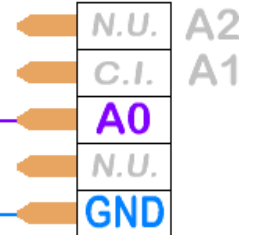
## Fiche n°44.

### Câblage de la "pieuvre" de raccordement.

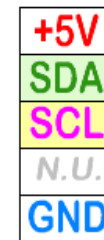
C.I. Relais et  
petit clavier



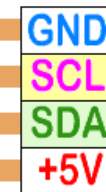
Carte Arduino  
pilotage relais



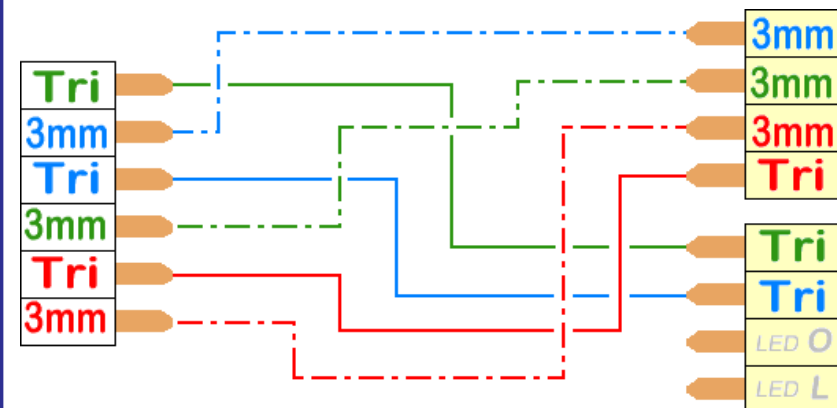
Multiplexeur n°1



Carte Arduino  
commande  
PCA9685 n°1



### Câblage des LEDs du petit clavier.



8

7

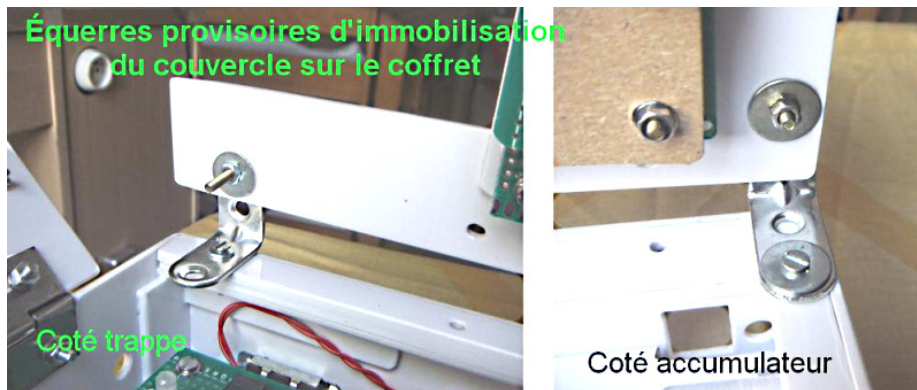


## Fiche n°45.

### Dépose du C.I. supportant l'afficheur OLED.

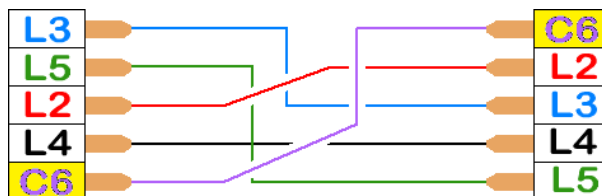
Cette procédure est optimisée pour tenir compte du fait que deux connecteurs sont masqués sous la façade du petit clavier et de ce fait d'un accès très inconfortable voir impossible.

- 01) Libérer les écrous de liaison du C.I. afficheur sur la semelle.
- 02) Dévisser le dessus et l'immobiliser sur deux équerres ou le déposer contre le coté latéral du B.P. de RESET.
- 03) Débrancher la ligne de l'afficheur OLED.
- 04) Libérer la ligne de liaison avec le relais électromagnétique.
- 05) Écarter un peu le C.I. et débrancher les deux lignes [7] et [8] qui le relie au multiplexeur n°2.
- 05) Retourner le C.I. et déposer la façade du petit clavier
- 06) Débrancher le connecteur des LEDs.
- 07) Libérer le connecteur du petit clavier.



Le C.I. de l'afficheur OLED est libre et peut être déposé.

**NOTE :** La ligne qui relie le petit clavier de servitude au connecteur de complément du clavier principal contient des liaisons croisées.



## Fiche n°48. Table des matières.

- Fiche n°1 : *Affectation des Entrées / Sorties.*
- Fiche n°2 : *Afficheur graphique OLED 1,3 pouce.*
- Fiche n°3 & n°4 : *BUS série au standard I2C. (1/2) & (2/2)*
- Fiche n°5 & n°6 : Méthodes de *Adafruit\_PWM servoDriver.h.*
- Fiche n°7 : *Chaînage de deux multiplexeurs PCA9685.*
- Fiche n°8 : *Chaînage électrique de plusieurs modules.*
- Fiche n°9 : *Branchements des deux multiplexeurs.*
- Fiche n°10 : *Tableau des LEDs simulant les ampoules*
- Fiche n°11 : *Présentation de la carte Arduino NANO.*
- Fiche n°12 : Brochage de la carte Arduino NANO.
- Fiche n°13 : *Architecture du CLAVIER à 30 touches.*
- Fiche n°14 : *Codage du CLAVIER à 30 touches.*
- Fiche n°15 : *Utilisation de P01\_Programme\_de\_base.ino*
- Fiche n°16 : *Utilisation de P02\_Validation\_du\_CLAVIER.ino*
- Fiche n°17 : *Le circuit imprimé du microcontrôleur.*
- Fiche n°18 : *Branchements extérieurs du circuit principal.*
- Fiche n°19 & n°20 : *Circuit imprimé du CLAVIER principal.*
- Fiche n°21 : *Répartition des B.P. et branchement du CLAVIER.*
- Fiche n°22 : *Circuit imprimé pour les connecteurs de liaison.*
- Fiche n°23 & n°24 : *TESTEUR DE CONTINUITÉ ET DE LED.*
- Fiche n°25 : *Mesure de la tension +5Vcc sur Arduino.*
- Fiche n°26 : *Câblage des deux claviers.*
- Fiche n°27 à n°32 : *Une Énigme fiable de référence..*
- Fiche n°33 & n°34 : *Implantation des textes en EEPROM.*
- Fiche n°35 : *Logement des données en EEPROM.*
- Fiche n°36 : *Interaction entre OLED et multiplexeur PCA9685.*
- Fiche n°37 : *Le relais d'isolement KEMET : EC2-5NU.*
- Fiche n°38 & n°39 : *C.I. Petit clavier et OLED.*
- Fiche n°40 & n°41 : *C.I. tableau des LEDs oranges.*
- Fiche n°42 & n°43 : *Fonctionnement électrique d'une ENIGMA.*
- Fiche n°44 : Câblage de la "pieuvre" et du petit clavier.
- Fiche n°45 : *Dépose du C.I. supportant l'afficheur OLED.*
- Fiche n°46 & n°47 : *Câblage : Ordre des opérations.*

## Fiche n°47.

### Câblage : Ordre des opérations. 2/2

Test de l'étape 10 : Comme le tableau des LEDs oranges n'est pas câblé on ne peut pas utiliser la commande [Z]. Pour tester le clavier principal il suffit de passer en Mode COMMANDE et de tester les 26 touches en comparant leur effet avec les comportements attendus.

11) Déposer le C.I. de l'afficheur après avoir libéré tous les HE14.

12) Réaliser toutes les lignes qui vont des deux multiplexeurs PCA9685 et les connecteurs du tableau des LEDs oranges.

**NOTE :** Les lignes qui partent des multiplexeurs vers les LEDs n'ont de la gaine thermorétractable que coté tableau des ampoules :

- Souplesse pour fléchir les lignes sous le C.I. de l'afficheur.
- Fiabilité mécanique coté LEDs car on débranche à chaque dépose du couvercle lors d'une intervention de maintenance importante.

13) C.I. de l'afficheur OLED non immobilisé en rétablir toutes ses connections sauf [7] et [8] qui sont libérées du multiplexeur. Le décaler vers l'accumulateur, le retourner et immobiliser la façade du petit clavier.

14) Rétablir les liaisons [7] et [8].

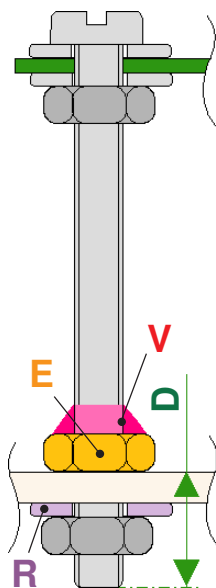
15) Immobiliser le C.I. de l'afficheur sur la semelle du coffret.

**NOTE :** Pour que les écrous E soient parfaitement positionnés sur les vis, ils sont placés à la distance D de 8mm par rapport à l'extrémité des vis  $\phi$  M3. Les écrous E sont alors collés en position sur les vis avec du vernis à ongles V par exemple. Pour faciliter l'immobilisation il n'y a des rondelles R que sur le dessous de la semelle du coffret.

16) COMMANDE [V] et affiner éventuellement la résistance ajustable sur l'U.C. pour indiquer exactement la tension de l'accumulateur.

17) Vérifier le tableau des ampoules oranges :

- Passer en Mode COMMANDE.
- Consigne [Z] pour ne plus chiffrer.
- Imposer le Mode CRYPTAGE.
- Tester les 26 lettre "en direct".



## Fiche n°46.

### Câblage : Ordre des opérations. 1/2

01) Souder la ligne de l'accumulateur à l'inverseur (Placé sur arrêt.) et celle de l'inverseur vers la nourrice d'alimentation.

02) Établir la liaison entre le C.I. alimentations et les borniers des multiplexeurs. (Dessin du C.I. de la nourrice en Fig.1)

03) Liaison entre le B.P. et le HE14 du RESET C.I. Arduino.

04) Ligne entre les deux multiplexeurs PCA9685.

05) Liaison provisoire directe entre C.I. Arduino et multiplexeur n°1 comme montré sur l'Image 84.JPG.

**NOTE :** Pour chaque vérification on met l'ensemble sous tension, on teste, puis on replace l'inverseur hors tension.

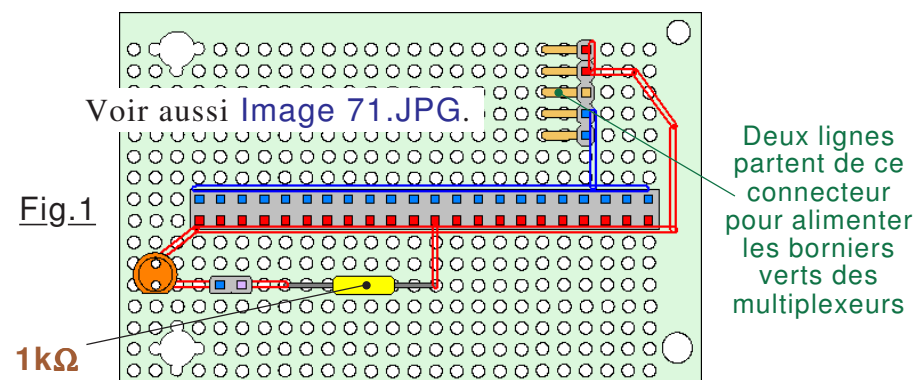
06) Installer provisoirement le C.I. de l'afficheur sans la façade du petit clavier. Compléter par la ligne U.C. vers afficheur. Mettre provisoirement sous tension : OLED doit afficher "le compteur".

07) Réaliser la pieuvre d'Image 85.JPG. Sur RESET OLED doit afficher "le compteur" et le petit relais doit cliquer.

08) Relier les deux lignes [7] et [8] du multiplexeur vers le HE14 des LEDs de servitude. (Voir Image 86.JPG) La petite LED de BIP doit s'illuminer et la LED triple verte clignoter.

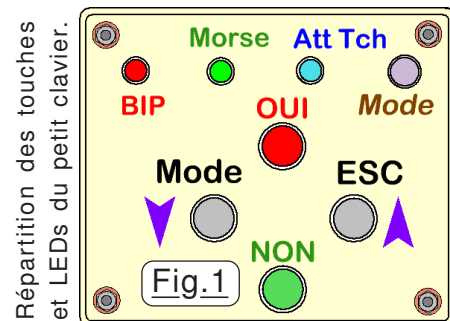
09) Réaliser la ligne de [C] vers les 5 broches du HE14 petit clavier. Le toron [C] doit être décalé sur le coté (Voir l'Image 90.JPG) pour dégager le trou de passage de la vis supportant la petite façade.

10) Établir la ligne de l'U.C. (U.C : C.I. de la carte Arduino) vers [A] et [B] du C.I. du clavier principal.



## Liste des lettres de COMMANDE valides. (1/3)

Cette liste est relative aux LETTRES du clavier principal qui sont affectées pour activer une fonction ou une option lorsque



la LED tricolore clignote en bleu, la machine étant alors en Mode **COMMANDE**. C'est la touche **Mode** du clavier secondaire représenté sur la Fig.1 qui permet d'alternier avec le Mode **CRYPTAGE**. En **CRYPTAGE** la LED tricolore *clignote en vert*. Chaque fois

que le programme est en attente d'une touche quelconque la petite LED bleue nommée **Att Tch** s'allume en continu. Dans presque toutes les fonctions et options il y a demande préalable de confirmation. Dans ce cas la petite LED bleue est complétée par la **LED tricolore qui s'allume en rouge également en continu**.

Seule la touche **OUI** du petit clavier sera considérée comme une approbation. **NON**, **ESC**, **Mode** ou n'importe quelle touche du clavier principal font revenir au mode **COMMANDE** sans effet, la commande étant alors ignorée.

[A] : **AJOUTER** une **Fiche croisée**. (*Même action si touche [X].*)

Le nombre maximal de fiches installées sera égal à 10 comme c'était le cas sur la machine d'origine.

• *Durant la saisie, à tout moment les touches **NON**, **ESC** ou **Mode** font sortir de ce mode.* Si à l'appel il y a déjà 10 fiches un message d'erreur est généré assorti d'un BIP.

[B] : **BRUTEUR** actif ou silencieux. S'il est actif la petite LED rouge nommée **BIP** du clavier secondaire sera allumée.

[C] : Afficher à l'écran la **Correspondance Lettre / CHIFFRES**. Exemples [C ou 3], [D ou 4], [E ou 5], [F ou 6], etc.

[D] : Imposer une configuration par **DÉFAUT** sur la machine.

[E] : **EFFACER** toutes les **Fiches croisée** si confirmé.

[F] : Restituer les trois orientations initiales des lettres des **Rotors** situées sous la **FENÊTRE**. (*À faire avant tout cryptage.*)

## Liste des lettres de COMMANDE valides. (3/3)

[T] : Mode **TEXTE** affiché en formaté sur le **Moniteur** de l'**IDE**.

Dans ce mode l'affichage est rapide, par défaut le **Morse** s'il est actif est suspendu. La saisie se fait par phrases complètes dans la fenêtre du **Moniteur** à **57600** baud.

Mode **TEXTE** : À l'ouverture les petites LEDs sont éteintes et la LED triple s'allume en blanc. La sortie de ce mode se fait avec '&'. On peut saisir un texte d'au maximum 60 caractères dans la fenêtre contextuelle du **Moniteur**, les suivants seront ignorés.

*Durant la saisie on peut à tout moment utiliser six options :*

- '\*' : Active la génération du Morse en vitesse lente.
- '/' : Suspend la génération des caractères en code Morse.
- '+' : Passe la génération du Morse en vitesse rapide.
- '-' : Impose à nouveau la génération du Morse en vitesse lente.
- '9' : Ne fait plus tourner les **Rotors** pour obtenir un cryptage ultra rapide *à condition toutefois que le Morse soit suspendu*.
- '0' : Rétablit la rotation des **Rotors**.

**ATTENTION** : Un caractère d'option prendra effet pour l'intégralité de la ligne saisie quel que soit son emplacement dans le texte. Exemples :

\* [AAA\*AA+A] : Les six lettres sont codées en **Morse Rapide**.

\* [AAAAA&AA] : Pour le '&' seules les cinq premières sont cryptées.

**NOTE** : L'option '9' reste mémorisée et se retrouve lorsque l'on recommence une saisie avec la commande [T].

[U] : **UTILISER** les protocoles de l'époque. (*Voir le verso.*)

[V] : Mesurer de la tension d'alimentation **+Vcc**.

[W] : (*Word*) Affiche les textes proposés dans la fenêtre du **Moniteur** de l'**IDE** lorsque l'on est en Mode **TEXTE**.

[X] : **AJOUTER** des **Fiches croisées**. (*Même action que [A].*)

[Y] : Test du tableau des ampoules. Bascule de type OUI/NON qui allume toutes les LEDs du tableau ou les éteint.

[Z] : Lettres en Morse générées *sans les crypter* pour apprentissage de ce code ancien. (*Bascule de type OUI/NON*) Si validé la petite LED verte clignote si l'option Morse est active.



## Protocoles de cryptage en 1939/1945.

Ces usages imposés par les états majors permettent de coder des textes avec ponctuations et lettres particulières sur une machine dont le clavier ne contient que les 26 lettres de l'alphabet de base. Les signes de ponctuation sont codés à l'époque par des lettres peu utilisées en allemand :

- **X** correspondant à un **point**.
- **Y** correspondant à une **virgule**.
- Les chiffres et les nombres sont écrits en toutes lettres.
- Les caractères accentués fréquents en allemand sont remplacés par : **ä** devient **AE**, **ö** devient **OE** et **ü** devient **UE**.
- **CH** est remplacé par **Q**.

Les abréviations sont possibles, pourvu que le texte reste intelligible. (*Zur Zeit deviendra par exemple z.Zt*)

- Le codage des noms propres consistait à les doubler.

### **Mise en œuvre d'ÉNIGMA.**

Tous les jours, à partir de 00H00 heure de Berlin la configuration d'initialisation était modifiée. Ces clefs de chiffage étaient distribuées à l'avance aux unités de manière secrète, et les officiers devaient tout faire pour qu'elles ne tombent pas dans des mains ennemies. L'opérateur préparait son message conformément aux conventions précisées ci-avant. Puis, texte constitué, le militaire organisait le **préambule** (*Groupe de 12 lettres*) **dont les trois premières lettres constituaient l'orientation initiale des trois rotors**. Ce préambule non crypté constituait le début du message. Enfin, **pour envoyer chaque nouveau message, toute la journée on devait replacer en orientation initiale les trois Rotors**. Pendant que l'assistant crypteur frappait les lettres sur le clavier, l'opérateur radio notait sur papier les lettres cryptées par la machine pour les transmettre en morse.

**NOTE :** La clef du jour changeait le plus souvent toutes les vingt-quatre heures, mais la durée durant laquelle elle était valide n'était pas la même d'armée en armée et a changé au cours de la guerre. Certaines étaient gardées deux jours, d'autres huit heures seulement. De même que les protocoles de chiffage étaient fonction des services et de l'évolution du conflit ...

## Liste des lettres de COMMANDE valides. (2/3)

- [G] : **Générer** une configuration aléatoire si confirmé.
- [H] : **HELP** qui affiche les commandes sur l'écran OLED. Cette fonction comporte plusieurs pages-écran et l'on "navigue" verticalement avec ▼ et ▲. Sortie du mode avec **NON**.
- [I] : **INITIALISER** le **Brouilleur** de la machine. (*Choix des trois Rotors virtuels insérés dans la machine, de leur Indexation interne, de leur Orientation et du Réflecteur.*)
- *Durant l'initialisation, à chaque élément la touche **ESC** fait sortir du mode et ignore la saisie des éléments qui suivent.*
  - *Le démonstrateur filtre les entrées :*
    - \* *Toute touche invalide génère un BIP d'alerte et est ignorée.*
    - \* *Si l'on utilise deux fois un même ROTOR il y a message d'erreur durant 2S et les trois ROTORS sont redemandés.*
    - \* *Pour le RÉFLECTEUR seuls [B] et [C] sont acceptés.*
- [J] : Jalonne le cheminement du cryptage dans le **Brouilleur**. Si le cheminement est validé quand on passe en mode **CRYPTAGE** la LED tricolore **clignote en orange**.
- [K] : Effacer une **Fiche croisée** dont on précise le numéro. (**KILL.**) En entrée de fonction les deux touches ▼ et ▲ permettent d'incrémenter ou de décrémenter le numéro de la fiche qui sera effacée si on confirme avec **OUI**. (*BIP si zéro fiche.*)
- [L] : **LISTER** la configuration machine actuellement installée.
- [M] : **MORSE** activé ou suspendu. Si validé la petite LED verte nommée **Morse** du clavier secondaire sera allumée.
- [N] : Vitesse de transmission du Morse rapide ou Normale. (*Basculer de type OUI/NON*) Si rapide la petite LED bleue clignote rapidement durant une seconde. Si lent clignote lentement.
- [O] : **ORDONNER** les **Fiches croisée** actuellement installées. (*Commande acceptée même s'il n'y a pas de Fiche branchée.*)
- [P] : Afficher l'espace actuellement disponible entre la **PILE** et le TAS et la version. (*N'est utile que pour les programmeurs.*)
- [Q] : Éteindre les quatre LEDs du petit clavier.
- [R] : **RESTITUER** la configuration initiale de la machine.
- [S] : **SAUVEGARDER** la configuration initiale de la machine.