

## 0 - INTRODUCTION

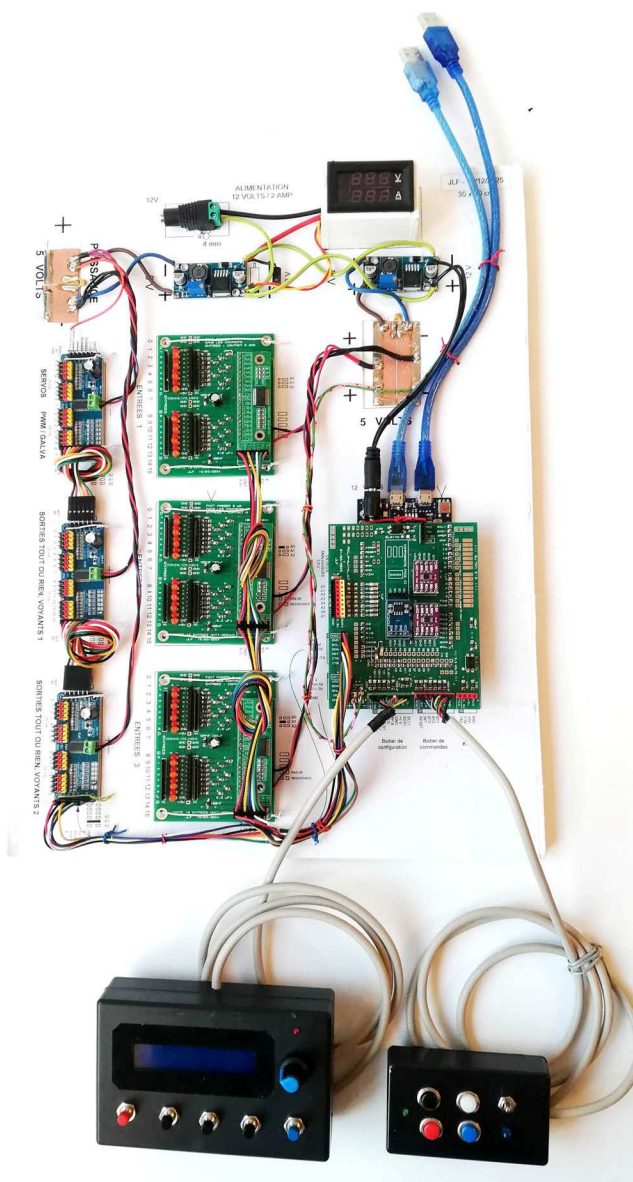
Ce document se trouve sur ce site : [http://www.la-tour.info/uts/uts\\_page15.html#conduite](http://www.la-tour.info/uts/uts_page15.html#conduite)

Les questions ou informations se trouvent sur ce site : <https://www.rmfr-magazine.com/phpBB/viewforum.php?f=21&sid=35cc0b70442dc7d11b0fa34b7e4ef729>

Ce montage sur une plaque unique permet de réaliser l'interface entre un véritable pupitre de locomotive, ou un pupitre de fabrication personnel, et un simulateur de conduite ferroviaire sur ordinateur.

La fabrication est décrite en détail, les fichiers des circuits imprimés sont fournis, ainsi que le code source pour la carte Arduino DUE.

Cette version est aboutie. En faisant appel à une seule carte Arduino DUE, une interface en un seul circuit imprimé et des modules I2C, on facilite le montage.



# I - PREPARATION

Ce document se trouve sur ce site : [http://www.la-tour.info/uts/uts\\_page15.html#conduite](http://www.la-tour.info/uts/uts_page15.html#conduite)

Si vous réaliser des améliorations matériels ou logicielles, merci de les poster sur le forum RMF pour en faire profiter le plus grand nombre.

Si vous publier cette réalisation avec ou sans améliorations, vous devez publier votre code source.

Ce logiciel est un logiciel libre. Exigence du concepteur : Ne pas modifier la ligne d'affichage "JLF xx/xx/xxxx" sur l'écran LCD.

On peut modifier ce programme et le diffuser. Dans ce cas, il faut préciser l'origine et donner accès aux sources modifiées.

Définition : Un logiciel libre est un logiciel distribué avec l'intégralité de ses programmes-sources, afin que l'ensemble des utilisateurs qui l'emploient, puissent l'enrichir et le redistribuer à leur tour.

Note : Un logiciel libre n'est pas nécessairement gratuit et les droits de la chaîne des auteurs sont préservés.

Équivalent étranger : free software, open source software.

(Source : Vocabulaire de l'informatique (liste de termes, expressions et définitions adoptés), NOR: CTNX0710138K, J.O n° 93 du 20 avril 2007 page 7078, texte n° 84)

logiciel libre

Par logiciel libre on entend un logiciel qui offre la liberté aux utilisateurs d'exécuter, de copier, de distribuer, d'étudier, de modifier et d'améliorer le logiciel. Plus précisément, elle fait référence à quatre types de liberté pour les utilisateurs du logiciel :

La liberté d'exécuter le programme, pour tous les usages (liberté 0).

La liberté d'étudier comment le programme fonctionne et de l'adapter à ses besoins (liberté 1). L'accès au code source est une condition requise.

La liberté de redistribuer des copies, (liberté 2).

La liberté d'améliorer le programme et de diffuser les améliorations au public pour en faire profiter toute la communauté (liberté 3). L'accès au code source est une condition requise.

Voici la partie informatique et électronique, pour la migration d'un pupitre de locomotive.

Ce document peut aussi servir à la construction d'un pupitre, à partir d'élément du commerce.

Ce code est conçu pour être banalisé le plus possible, et ainsi être installé dans n'importe quel pupitre.

Pour adapter ce montage à n'importe quel pupitre, les programmes d'interface avec l'ordinateur, sont dans des routines indépendantes. Ca facilite leurs modifications.

Ces sous-programmes sont adaptés au logiciel 'Train Simulator Classic de DTG', avec l'interface 'TS Classic Raildriver and Joystick Interface', la locomotive BB67000 de Yohann et la BB7200 de SimExpress.

Le plus gros du travail est déjà réalisé, et l'adaptation de ce montage à d'autres environnements est relativement aisé.

Ce montage peut :

Piloter 8 servomoteurs, pour manomètres...

Alimenter 8 sorties PWM, pour compteur de vitesse, compte-tours, galvanomètres...

Allumer 32 voyants, buzzer ou autres appareils en tout ou rien...

Lire 8 grandeurs analogiques.

Récupérer l'information de 48 entrées, interrupteur, boutons poussoirs...

Ce montage utilise un seul Arduino, pour faire les entrées/sorties. C'est un Arduino DUE, qui fonctionne sous 3,3 Volts. J'ai inséré des convertisseurs 3,3V / 5 V pour y brancher des cartes du commerce.

Pour avoir plus d'entrées/sortie, on peut ajouter des modules I2C supplémentaires.

# LA FABRICATION DES MODULES POUR LA CARTE PRINCIPALE

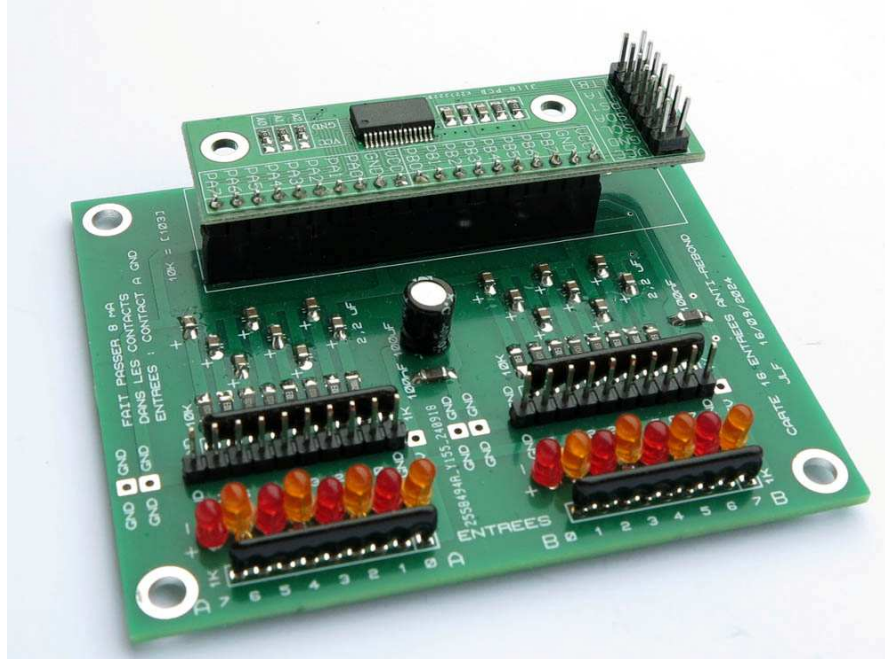
Il faut dans un premier temps, réaliser ces circuits imprimés et les boîtiers qui vont être utilisés sur la carte principale.

## 1 / Les entrées I2C

Pour lire les 48 entrées, j'utilise les modules MCP23017.

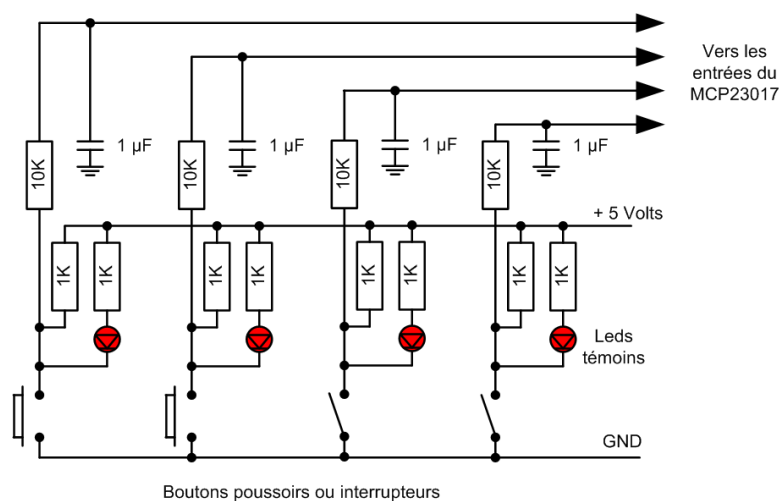
Ils sont insérés sur un plus grand circuit imprimé, pour faciliter leur branchement.

Il y a besoin de trois plaques de ce type :



Le fichier pour faire fabriquer ce circuit imprimé est fourni. Il suffit d'envoyer le fichier '**MCP23017 - Entrées anti-rebonds-cms - PCB - CAD/CAM.ZIP**', à un site comme 'JLPCB'. <https://jlcpcb.com>

Schéma des entrées :



La led s'allume quand le contact est fermé. C'est pratique pour savoir si l'interrupteur fonctionne bien.

La deuxième résistance de 1K, permet de monter la tension à 5 Volts quand le contact est ouvert.

Quand le contact est fermé, il passe 8 mA dans l'interrupteur, ce qui permet de nettoyer les contacts.

Les contacteurs d'origine sur les pupitres sont fait pour passer des ampères, il faut un minimum de courant pour qu'ils fonctionnent bien.

La résistance de 10 K et le condensateur de 1  $\mu$ F donne une constante de temps de 10 msec.  
C'est suffisant pour éviter des rebonds lors d'ouverture ou fermeture des contacts.

#### Composants de la plaque des entrées :

- 16 x 10 K Ohms cms (format 1206)
- 2 x 100 nF cms (format 1206)
- 16 x 1  $\mu$ F cms (format 1206) 10 Volts minimum
- 4 x Réseaux de 8 résistances de 1K, 1 commun
- 1 x 100  $\mu$ F / 16 Volts minimum
- 16 x leds 3 mm
- Une barrette plots femelles
- Deux barrettes 10 plots mâles bleus

Pour les barrettes, on en trouve de bonne qualité ici :

TME : <https://www.tme.eu/fr/details/zi262-40sg/batis-de-prises-et-prises-de-courant/connfly/ds1023-1-40s21/>

TME : <https://www.tme.eu/fr/details/zi201-40g/batis-de-prises-et-prises-de-courant/connfly/ds1021-1-40sf11-b/>

CHANZON : <https://fr.aliexpress.com/item/1005003610333849.html>

CHANZON : <https://fr.aliexpress.com/item/1005003642939564.html>

#### Fabrication de la plaque du circuit imprimé :

- Souder les résistances cms (format 1206) de 10K.
- Souder les condensateurs cms (format 1206) de 1  $\mu$ F.
- Souder les condensateurs cms (format 1206) de 100 nF.

Souder les diodes leds. Commencer par les paires, puis souder les impaires, car la place est un peu juste.

Donner un coup de lime ou de pince coupante sur les cotés pour les rétrécir.

Souder les réseaux de 8 résistances 1K. Attention, il y a un sens. Le plot commun est marqué d'un rond '●', sur le réseau de résistances et sur le circuit imprimé.

Souder les picots mâles pour les entrées. Attention, j'ai prévu un plot pour le 5 Volts, mais il ne faut pas l'utiliser.

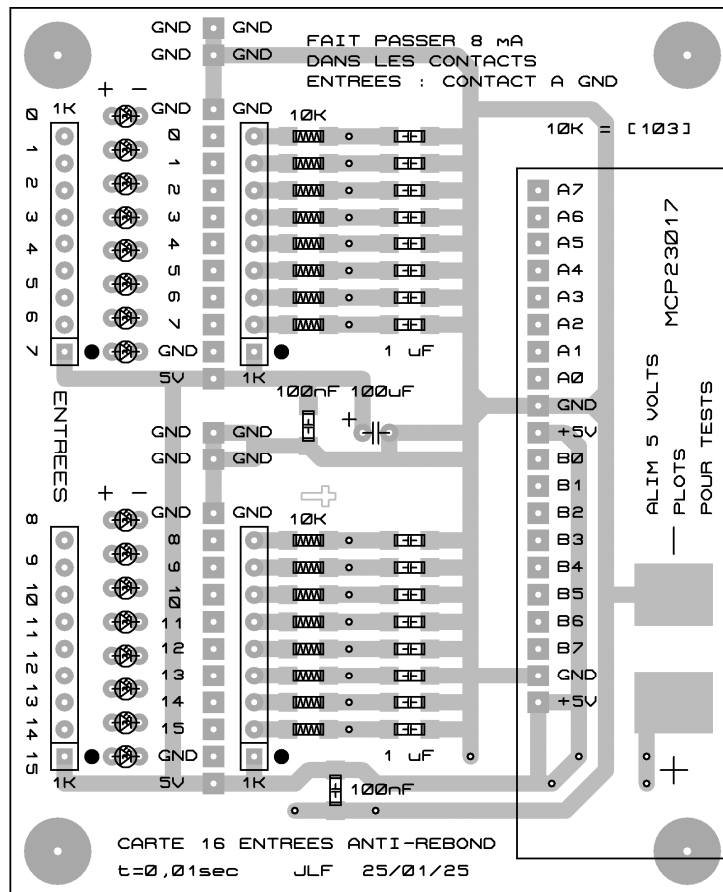
**Les barrettes des entrées, ne doivent pas aller sur les plots 5V. On prendra des barrettes de 10 plots.** Les plots 5V, c'est pour une autre utilisation, comme pour un circuit d'entrée alimenté.

Souder des barrettes de **10** plots mâles.

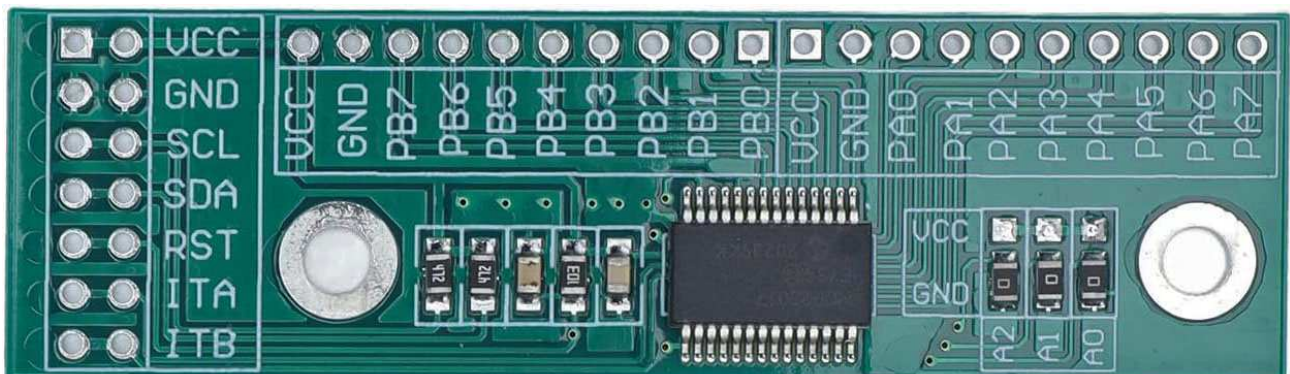
Souder la barrette femelle pour le MCP23017.

#### Remarque :

Pour les interrupteurs uniquement, si l'on constate beaucoup de faux contact, on peut passer la valeur des condensateurs à 10  $\mu$ F. Ne pas le faire pour les boutons poussoirs, sinon on risque de rater une activation en raison du délai de filtrage trop important.



Type de module MCP23017 utilisé :



Quand une entrée change d'état, la sortie INTA passe à 0. Cette sortie INTA est branchée à l'Arduino, qui va interroger le MCP23017, que lorsqu'une entrée a bougé.

Les sorties INTA sont à collecteur ouvert, et elles sont reliées à une seule broche d'entrée de l'Arduino.

#### Composants de la plaque des entrées :

Une barrette plots mâles

#### Fabrication de la plaque du circuit imprimé :

Souder la grande barrette mâle sous le circuit imprimé.

Souder les deux petites barrettes mâles 10 plots sur le circuit imprimé.





## 2 / Les sorties I2C

Pour les sorties, j'utilise des modules PCA9685 tous montés.

Ils sont conçus pour sortir un signal pwm sur 4096 pas.

En réglant la fréquence sur 62,5 Hertz, on peut piloter efficacement des servomoteurs sur 256 pas.

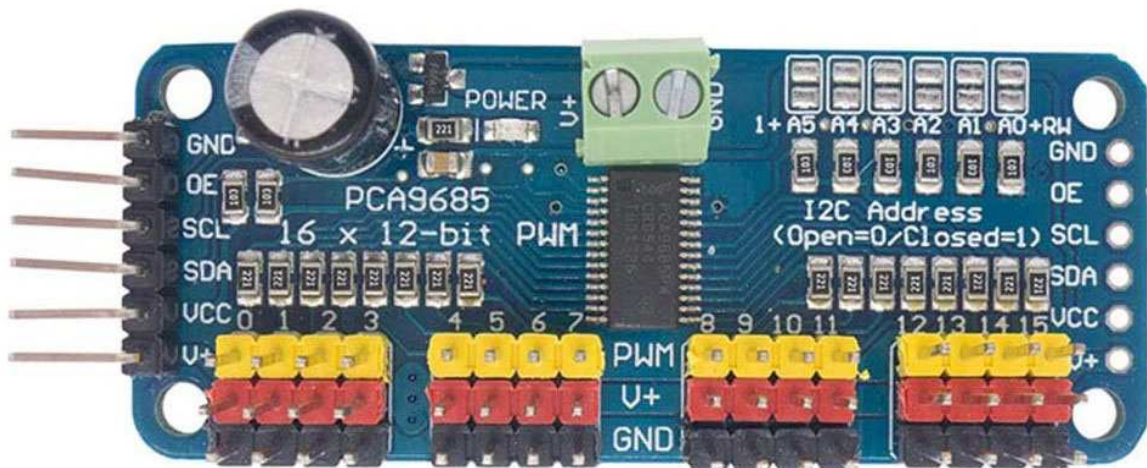
A partir d'un variable du programme Arduino comprise entre 0 et 255, on obtient un créneau standard de 1000  $\mu$ s à 2000  $\mu$ s, avec une résolution de 4  $\mu$ s.

On peut même obtenir un créneau étendu, avec les amplitudes maximums de 700  $\mu$ s à 2300  $\mu$ s. Ca permet d'avoir une amplitude de rotation proche de 180° au lieu des 90° habituels. Il faudra vérifier, si le servomoteur accepte ces créneaux étendus.

La fréquence de répétition du signal est de 16 msec au lieu des 20 msec habituel, mais les servomoteurs récents fonctionnent bien à cette fréquence. Ca donne une fréquence de répétition de 62 Hz au lieu des 50 Hz habituelle.

La résolution en sortie est de 1/256. Exemple : Si l'on prend un compteur de vitesse prévu pour aller jusqu'à 255 km/h, ça fera une résolution de 1 km/h.

On prendra ces modules tout montés.



Les sorties de 0 à 7 alimentent 8 servomoteurs.

Les sorties de 8 à 15 sont en mode pwm, pour obtenir un signal haché 62 Hz en sortie, avec une moyenne allant de 0 à 5 volts.

Ces modules peuvent aussi être utilisé en sortie tout ou rien, pour alimenter des voyants, des klaxons ou sonneries.

La charge doit être placée de préférence entre la sortie et le + 5 Volts. La limite du courant est alors de 25 mA par sortie.

Si un accessoire est branché entre une sortie et la masse, la sortie ne peut délivrer que 10 mA maximum.

## La carte d'extension vers les servomoteurs

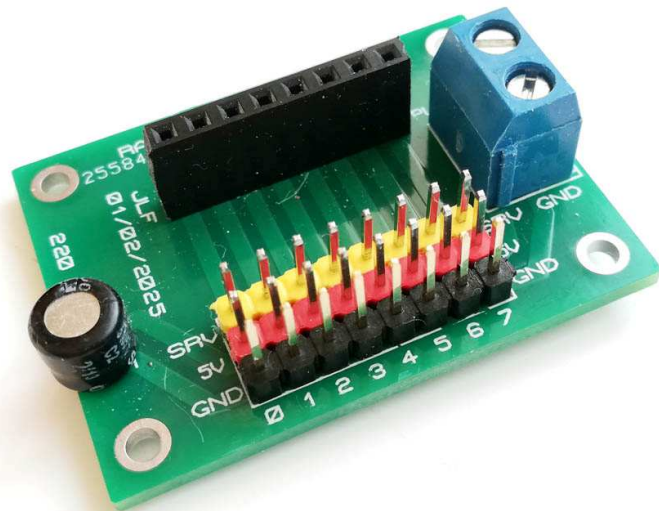
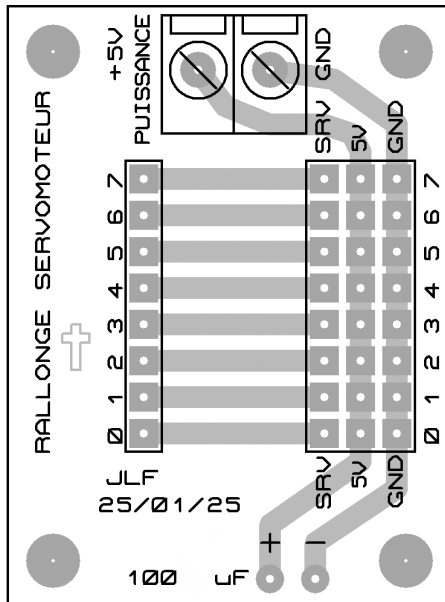
On peut brancher les servomoteurs directement sur la carte PCA9685.

Si les servomoteurs sont éloignés, il est préférable de câbler une plaque intermédiaire.

Les servomoteurs consomment beaucoup, avec des courants de pointe important. Pour éviter les parasites et les chutes de tension, utiliser cette plaque avec son condensateur tampon.

Elle se monte avec des barrettes de plots mâles, de même couleur que la carte PCA9685.

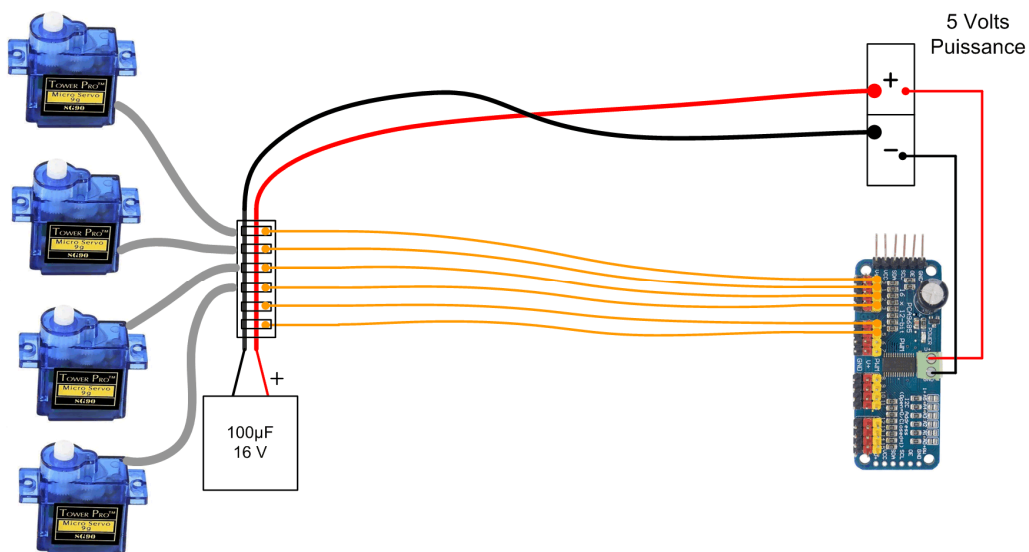
On soudera un condensateur de 100  $\mu$ F / 16 Volts sur le coté.



Le circuit imprimé est le : PDC\_Arduino\_JLF\_SRV - CADCAM.ZIP

Le fichier pour faire fabriquer ce circuit imprimé est fourni. Il suffit d'envoyer le fichier PDC\_Arduino\_JLF\_SRV - CADCAM.ZIP, à un site comme 'JLPCB'. <https://jlcpcb.com>

Cette plaque sera utilisée de cette façon, avec des gros fils d'alimentation 0 et + 5 Volts :





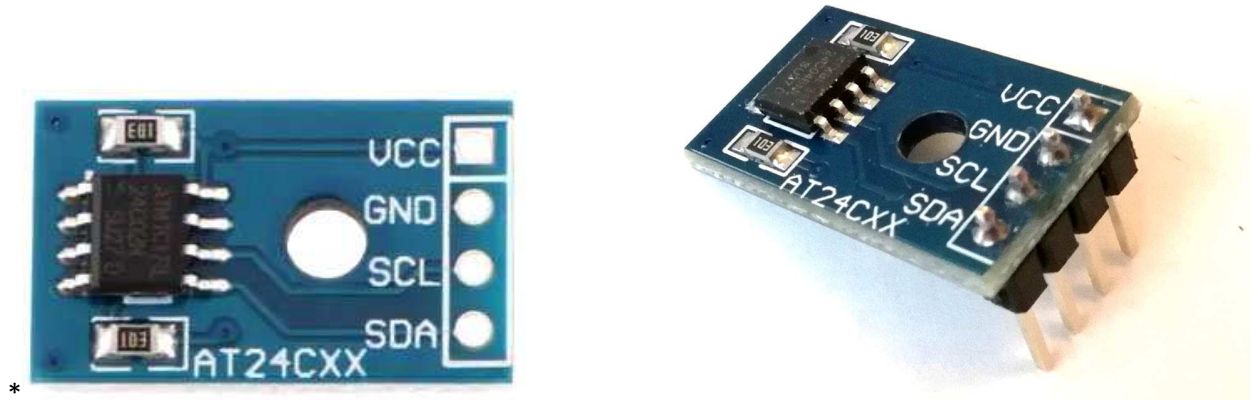
### 3 / L'eeeprom I2C

Pour garder en mémoire les paramètres, il faut les stocker en mémoire.

L'Arduino DUE, n'a pas de zone eeprom en interne.

On va donc utiliser une eeprom externe, accessible par le bus I2C.

J'ai pris ce module :



Mon programme ne fonctionne qu'avec la puce **24C04** de 512 octets.

Si vous prenez une autre puce, il faudra modifier le programme ou changer de librairie.

Composants de la plaque des entrées :

Module **24C04**

Une barrette plots mâles.

Fabrication de la plaque du circuit imprimé :

Souder la barrette mâle sous le circuit imprimé.

Si vous utiliser une 24C08, 24C16,,, 24C 256, mon programme ne fonctionnera pas. Il faudrait changer de librairie.

Il existe aussi un autre module eeprom de dimension plus importante, avec une eeprom au format DIL8 sur un support. Ce module convient aussi.

### 4 / Translateur de tension sur le bus I2C

L'Arduino utilisé est un modèle DUE. Il est très rapide, mais il a l'inconvénient de tourner sous 3,3 Volts.

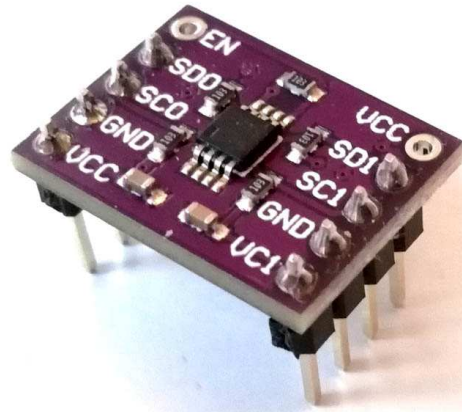
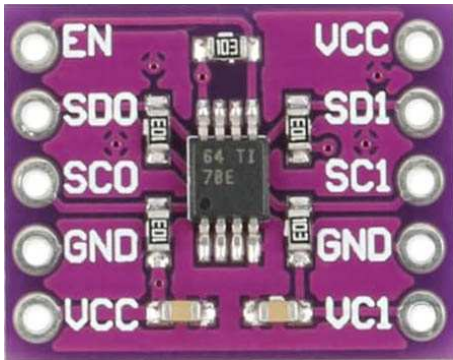
Il ne faut pas mettre plus de 3,3 Volts sur ses broches, au risque de le griller instantanément.

Il faudra donc faire très attention au moment d'assembler toutes les parties.

Pour passer de l'Arduino 3,3 Volts aux circuits I2C, MC23017, PCA9685 et eeprom, il faut des translateurs de niveaux de tension.

J'ai pris des modules avec le circuit : PCA9515A. Ca fonctionne très bien.

J'utilise ce circuit : C'est un module référence : CJMCU-9515 :



#### Composants de la plaque des entrées :

Module CJMCU-9515

Deux barrettes plots mâles.

#### Fabrication de la plaque du circuit imprimé :

Souder les barrettes mâles sous le circuit imprimé. **Attention prendre une barrette de 4 picots.** Il n'y a pas de picot sur les pastilles du haut : EN et VCC.

**Pour éviter de griller l'Arduino si l'on retourne le module en le rebranchant, et d'envoyer le +5 Volts sur les entrées 3,3 Volts de l'Arduino, prendre des barrettes de 4 picots.**

- A gauche, ne pas mettre de picots sur EN.
- A droite, ne pas mettre de picots sur VCC.

## 5 / Le boîtier de commande déporté

Pour faciliter l'exploitation de l'Arduino, on va installer un petit boîtier de commande sur le pupitre.

Il va permettre d'exploiter le pupitre. On configurera les entrées de l'Arduino en mode 'Pull Up'.

C'est un boîtier optionnel, la carte peut fonctionner sans ce boîtier.

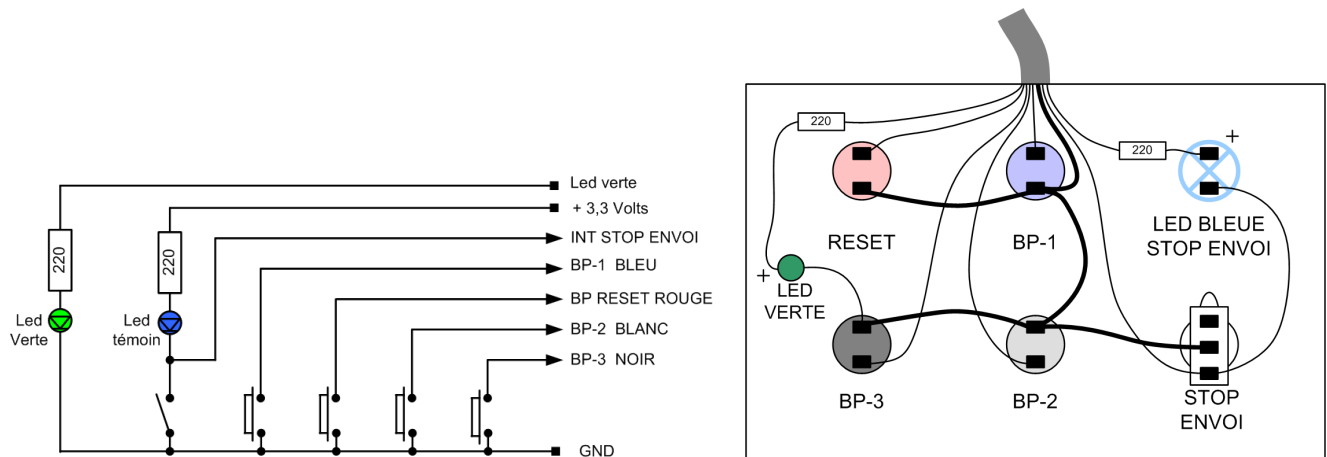
#### Les commandes disponibles :

- L'interrupteur 'STOP ENVOI', bloque toutes les émissions de l'Arduino. C'est très pratique, car l'Arduino peut envoyer des codes de touche du clavier, ce qui est gênant quand le simulateur n'est pas en fonctionnement. Dans ce cas, la led bleue 5 mm est allumée.
- Le bouton rouge 'RESET' est relié au bouton reset de l'Arduino.
- Bouton n° 1 bleu = Sectionneur de batterie (*Exemple*).
- Bouton n° 2 blanc = Commutateur d'éclairage (*Exemple*).
- Bouton n° 3 noir = Frein à main (*Exemple*).
- Une led verte 3 mm.

### Le boîtier de commande :



### Schéma du boîtier de commande et le câblage du boîtier de commande, vue de l'intérieur :



### Composants :

- 1 x Led bleue 5 mm
- 1 x Led verte 3 mm
- 2 x 220 Ohms
- 1 x Interrupteurs
- 4 x Boutons poussoirs
- 1 x Câble 8 conducteurs minimum.

Souder ces composants. Le câble ne doit pas dépasser deux mètres. Si il y a des fils en trop, les utiliser pour le fil 'GND'.

La led verte qui clignote, permet de savoir si le programme fonctionne normalement.

L'extrémité du câble sera reliée à une barrette mâle de 8 plots, suivant le câblage suivant :

- GND - Masse
- 3,3V vers + Led bleue
- LED Verte (+)
- BT1 - Bouton poussoir Bleu
- BT2 - Bouton poussoir Blanc
- BT3 - Bouton poussoir Noir
- INT - Interrupteur
- RESET - Bouton poussoir Rouge 'Reset'.



## 6 / Les échanges avec le KVB ou d'autres équipements distants

Pour le KVB, que ce soit un vrai modèle modifié, ou un modèle créé de toute part, il faut une isolation par optocoupleur.

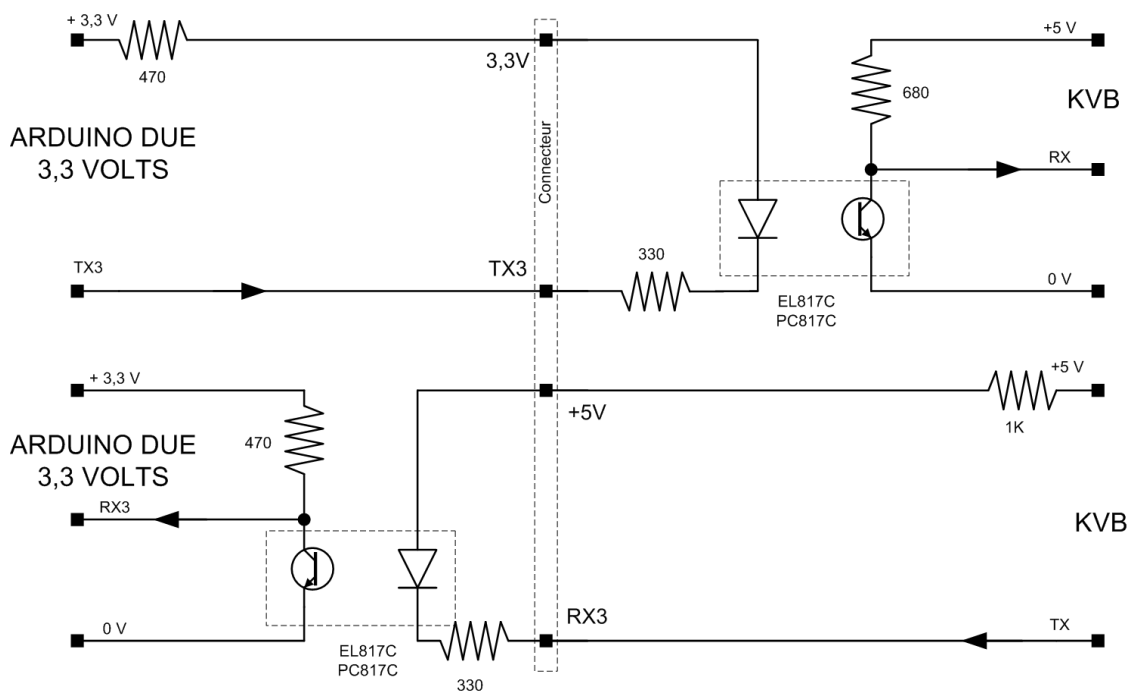
Le KVB modifié : [http://www.la-tour.info/uts/uts\\_page15.html#kvb](http://www.la-tour.info/uts/uts_page15.html#kvb)

Si l'on a un KVB, ou un autre équipement externe branché sur la liaison série, on insère un montage à base d'optocoupleur.

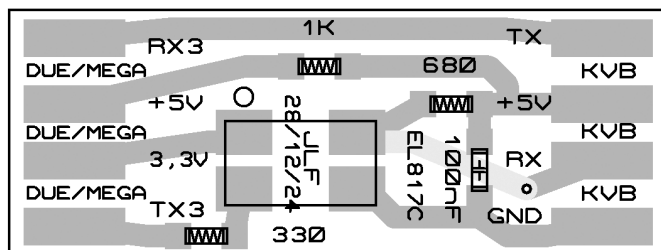
Le montage suivant fonctionne à 9600 bps maximum, à cause de la lenteur des optocoupleurs.

Pour d'autres explications, voir le document KVB : [http://www.la-tour.info/uts/uts\\_page15.html#kvb](http://www.la-tour.info/uts/uts_page15.html#kvb)

Schéma :



Coté équipement KVB, on utilisera cette plaquette en sortie d'équipement :



Le fichier pour faire fabriquer ce circuit imprimé est fourni. Il suffit d'envoyer le fichier 'PDC\_Arduino\_JLF\_OPTO - CADCAM.ZIP', à un site comme 'JLPCB'. <https://jlpcb.com>

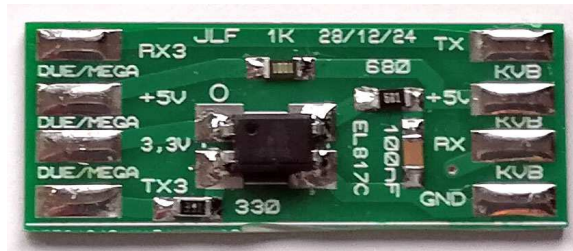
Composants :

- 1 x Optocoupleur EL817c
- 1 x 330 Ohms cms (format 1206)
- 1 x 680 Ohms cms (format 1206)
- 1 x 1 KOhms cms (format 1206)
- 1 x 100 nF cms (format 1206)



#### Fabrication du circuit d'isolation :

Souder tous ces composants. Repérer le rond '●' sur l'optocoupleur, pour l'aligner avec celui sur le circuit imprimé.



Il faudra installer cette plaque pour dialoguer avec le KVB, ou un autre appareil utilisant une liaison série. Elle se situe près du boîtier KVB.

TEST : Brancher ce montage derrière l'équipement. Mettre sous tension.

⇒ Mettre une tension de 3, 3 à 5 Volts entre TX3 et 3,3V. La tension en sortie entre GND et RX doit descendre à presque 0 Volt : oui / non ?

## **7 / Le boîtier de configuration général**

L'intérêt de ce montage et du logiciel l'accompagnant, c'est l'utilisation d'un écran pour configurer le pupitre.

Pas besoin de programmer à tâtons le code, suivant le type de pupitre. On pourra le faire en direct à partir du boîtier de configuration.

Le programme peut fonctionner sans ce boîtier, mais il est indispensable pour configurer l'Arduino.

Ne pas dépasser 1,5 m de long pour ce câble, ce qui doit suffire à le poser sur le pupitre, le temps de réaliser les réglages. Le bus I2C n'est pas adapté à des distances de plus de 1,5 mètre.

#### Le boîtier de configuration :



Il permet de paramétrer la course des servomoteurs et des sorties pwm.

Pour limiter le nombre de fil vers ce boîtier, les boutons poussoirs sont tous reliés à un seul fil, avec des résistances.

Suivant le bouton poussé, la tension en sortie varie. Ce plot est relié à une entrée de lecture analogique de l'Arduino.

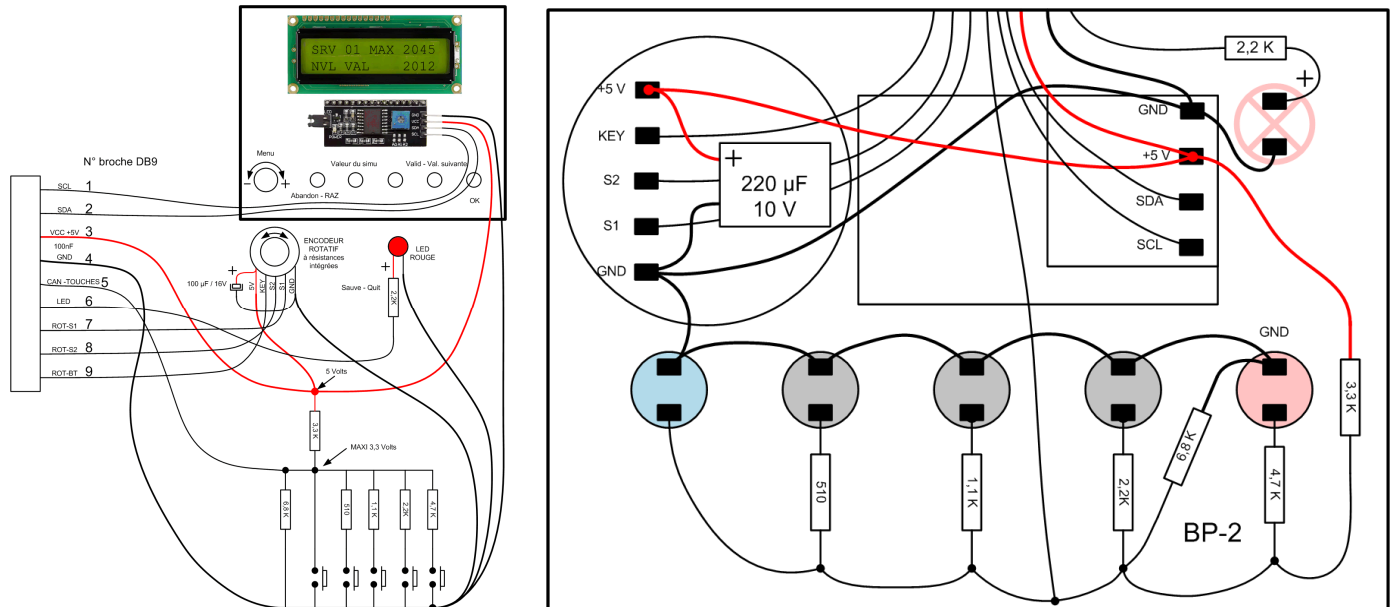
L'affectation des boutons est la suivante :

(Raz ou abandon) (Valeur du simulateur) (Servo, Galva suivant) (Sauve + quitte) (Validation)

La led rouge est là pour signaler une erreur.

Le bouton rotatif permet de naviguer dans les menus.

Schéma et câblage vu de l'intérieur :



Composants :

- 1 x Led rouge 3 mm
- 1 x 510 Ohms
- 1 x 1,1 K Ohms
- 2 x 2,2 K Ohms
- 1 x 3,3 K Ohms
- 1 x 4,7 K Ohms
- 1 x 6,8 K Ohms
- 5 x Boutons poussoirs
- 1 x Ecran LCD 1602 avec interface I2C
- 1 x Condensateur 220 µF / 10 Volts ou plus
- 1 x Contacteur rotatif type EC11 : <https://fr.aliexpress.com/w/wholesale-potentiom%C3%A8tre-rotatif-audio-rond-ec11.html?spm=a2g0o.productlist.search.0>

L'extrémité du câble sera reliée à une barrette mâle de 9 plots, suivant le câblage suivant :

- SCL1
- SDA1
- +5 V
- GND
- BP-A
- LED-
- POT-S1
- POT-S2
- POT-SW

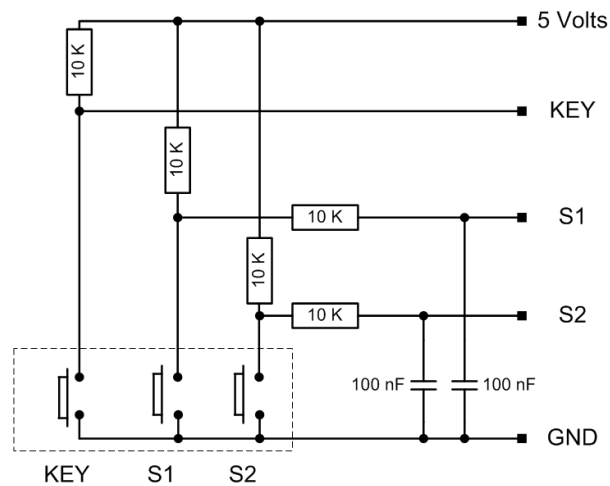


### Schéma équivalent du bouton rotatif :

Le modèle acheté du contacteur rotatif modèle EC11, doit être solidaire d'un **circuit imprimé rond**, avec des résistances et condensateurs soudés dessous. C'est le filtre de sortie indispensable.



Si vous avez un codeur rotatif, sans ces résistances et condensateurs, il faut obligatoirement ajouter ces composants. Ces composants sont obligatoires pour filtrer les signaux et avoir ensuite 3,3 Volts aux bornes de l'Arduino DUE.



## 7 / Le grand circuit imprimé posé sur l'Arduino

J'ai réalisé une seule grande plaque qui regroupe toutes les interfaces. Ca réduit nettement le câblage, tout en améliorant la fiabilité.

Le fichier pour faire fabriquer ce circuit imprimé est fourni. Il suffit d'envoyer le fichier **Arduino DUE - Bouclier PDC - JLF - CADCAM.ZIP'**, à un site comme 'JLPCB'. <https://jlpcb.com>

Le circuit imprimé final amélioré est légèrement différent de celui présenté ici.

Le montage est facile, toutes les références des composants sont marquées sur le circuit.

Il y a deux optocoupleurs EL817c.

Il y a deux types de supports pour une eeprom 24C04, mais un seul support doit être peuplé.

Pour les barrettes, on en trouve de bonne qualité ici :

TME : <https://www.tme.eu/fr/details/zl262-40sg/batis-de-prises-et-prises-de-courant/connfly/ds1023-1-40s21/>

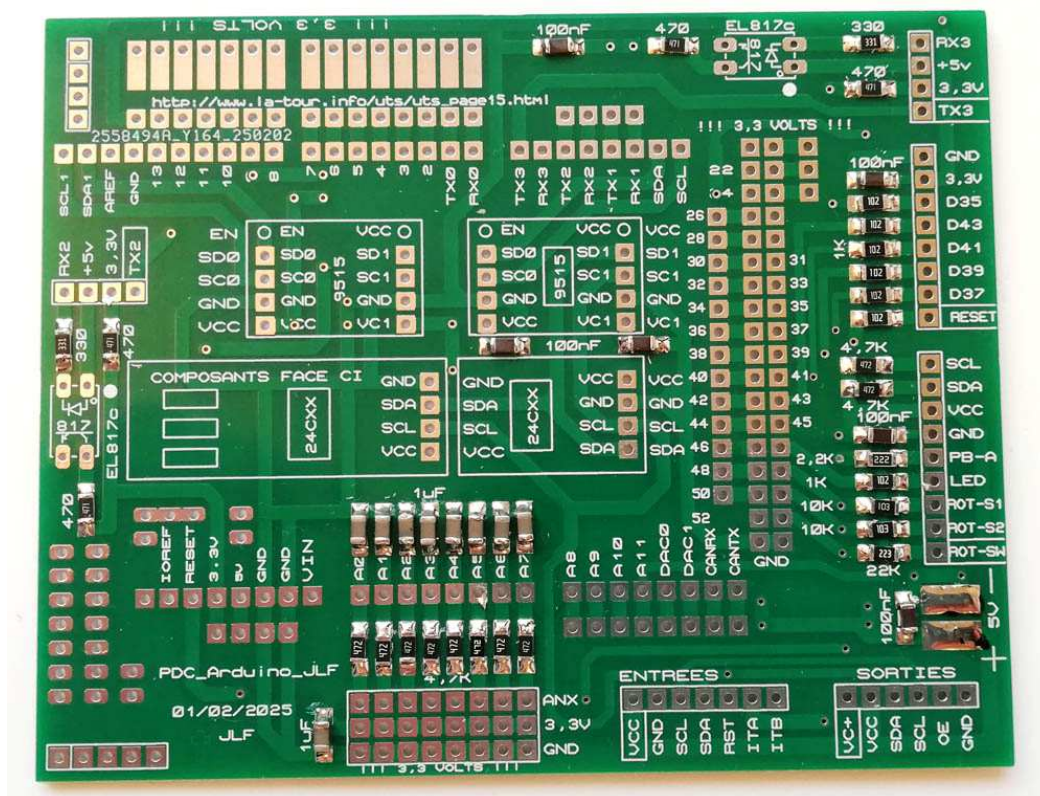
TME : <https://www.tme.eu/fr/details/zl201-40g/batis-de-prises-et-prises-de-courant/connfly/ds1021-1-40sf11-b/>

CHANZON : <https://fr.aliexpress.com/item/1005003610333849.html>

CHANZON : <https://fr.aliexpress.com/item/1005003642939564.html>

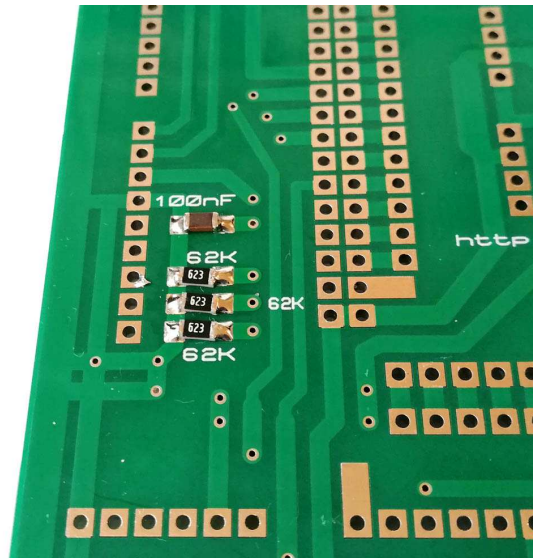
Commencer par souder les résistances et les condensateurs cms au format 1206.

Si la sérigraphie indique 470, c'est 470 Ohms. Ne pas confondre avec le marquage des résistances. Une résistance de 470, est marquée [471].

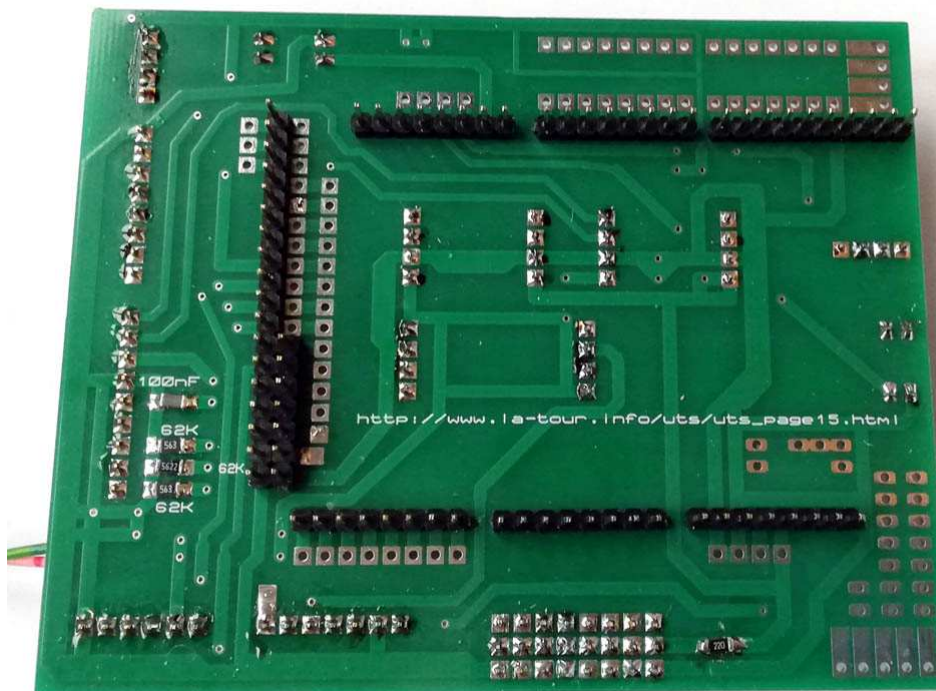




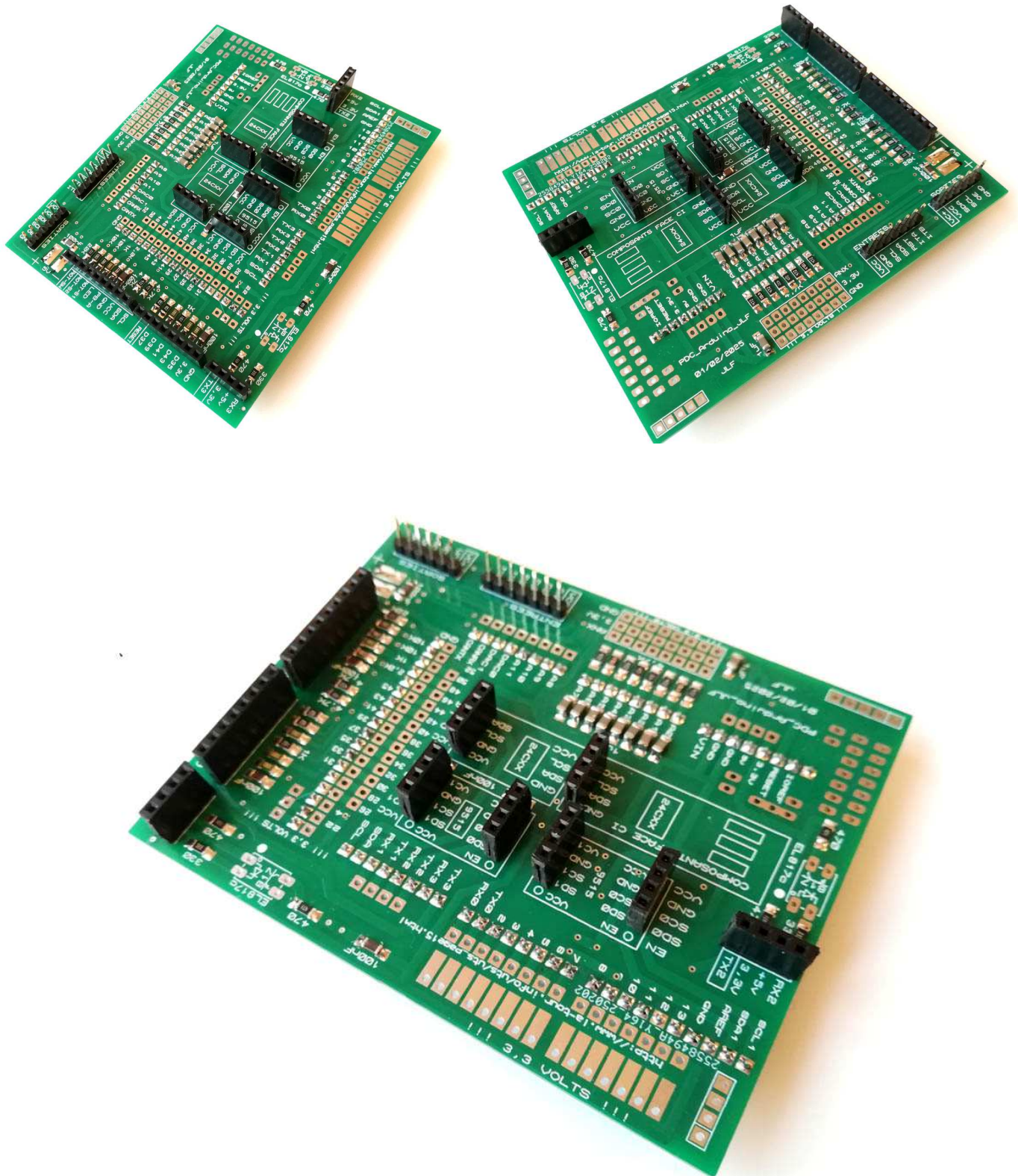
Souder les 4 composants en dessous



Souder les barrettes qui se connecteront sur la carte Arduino DUE



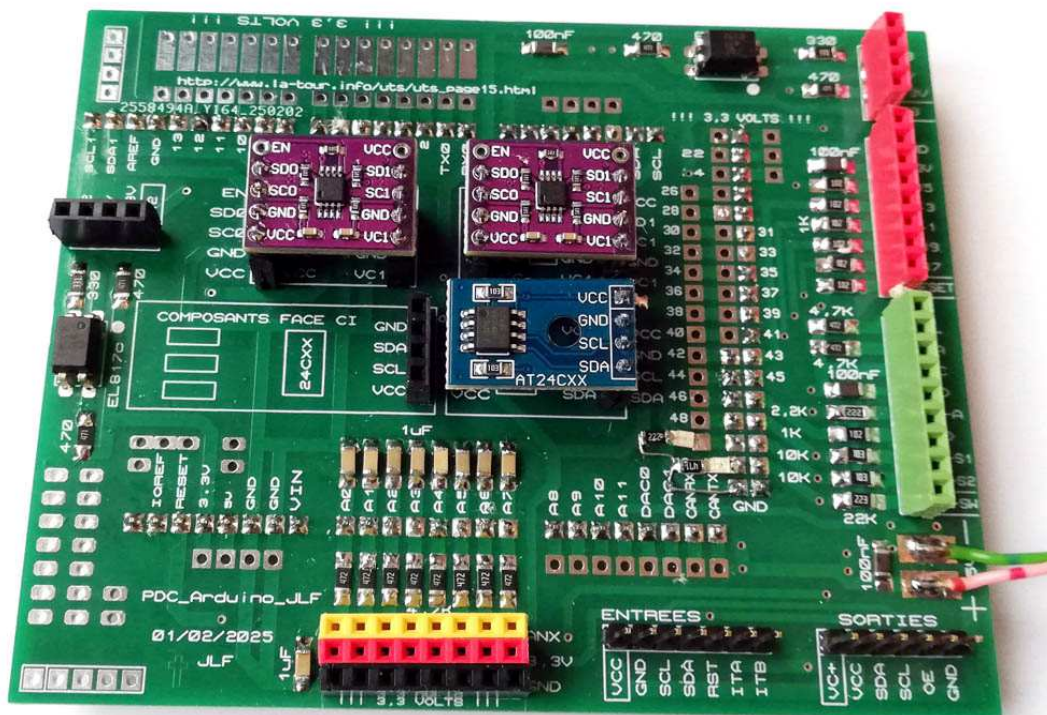
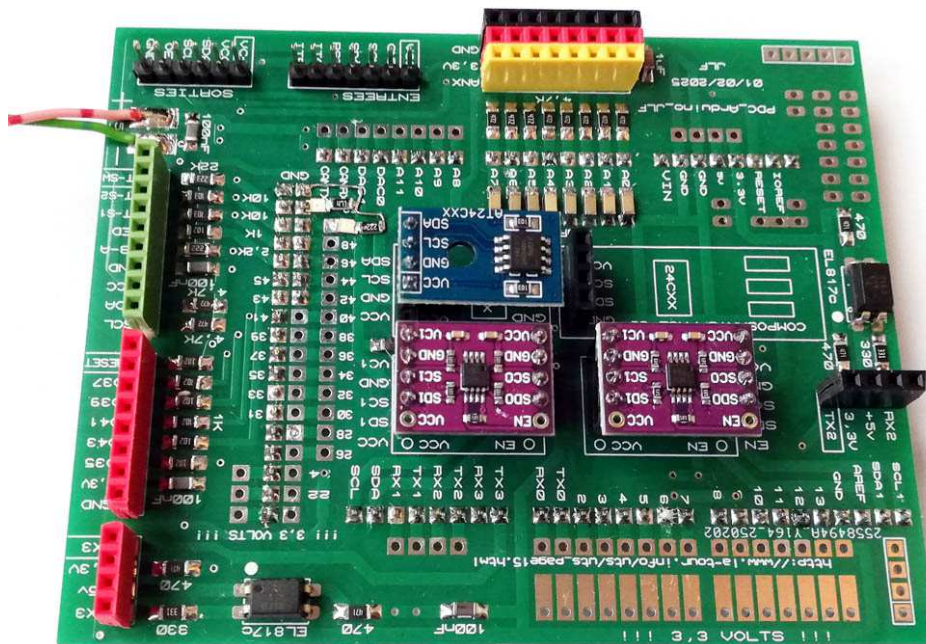
Souder les barrettes mâles et femelles du dessus.





Souder les optocoupleur EL817C et placer les modules sur leurs supports.

Sur la version finale du circuit imprimé, il n'y a pas de leds à souder près du plot n° 48.



A+