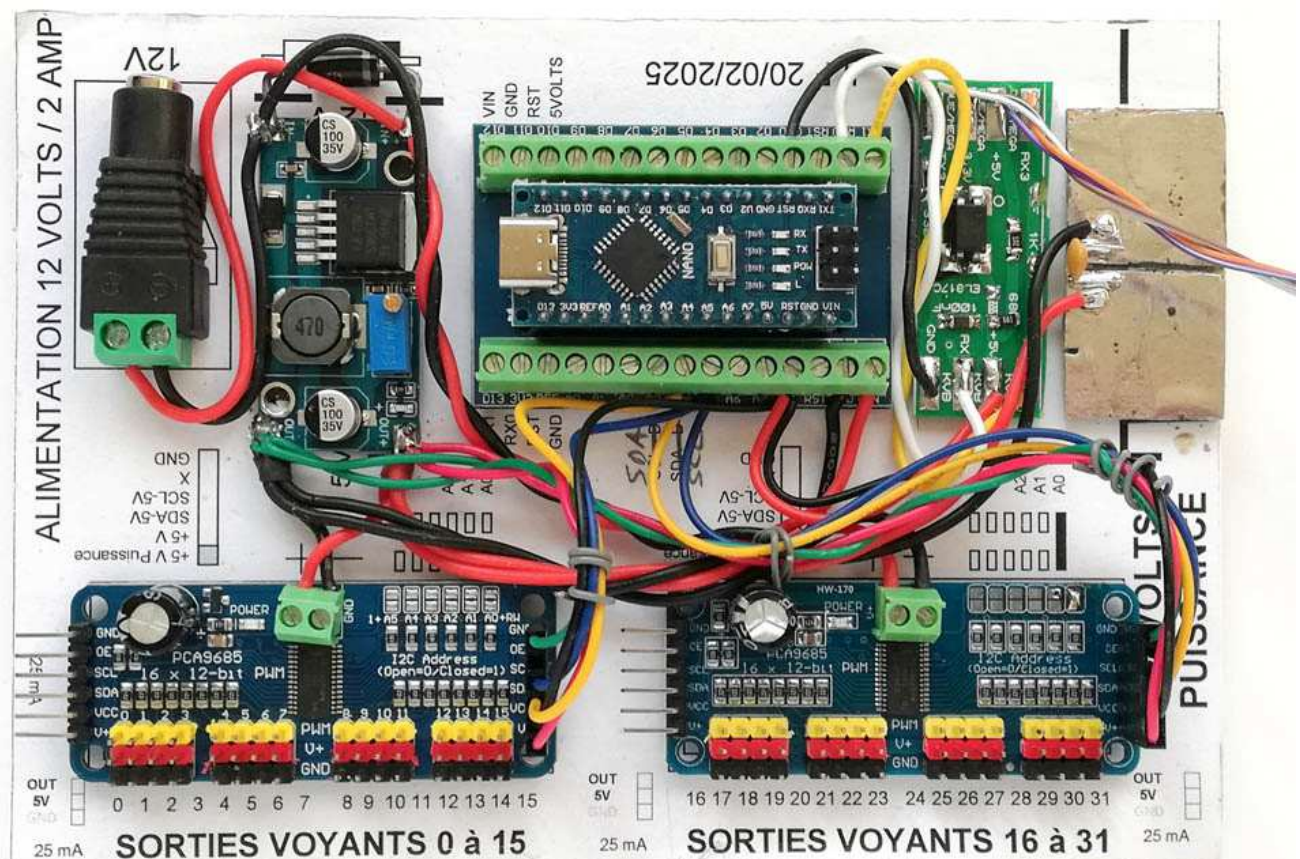


12/05/2025

Ce document se trouve ici : [http://www.la-tour.info/uts/uts\\_page15.html](http://www.la-tour.info/uts/uts_page15.html)

Le montage pour 32 sorties tient sur une plaque de 12 x 18 cm.



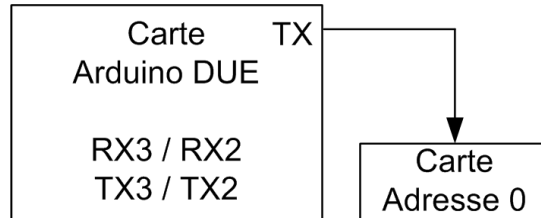
## 1 / Fonctionnement

La liaison série est à 9600 bps, car l'optocoupleur utilisé est assez lent.

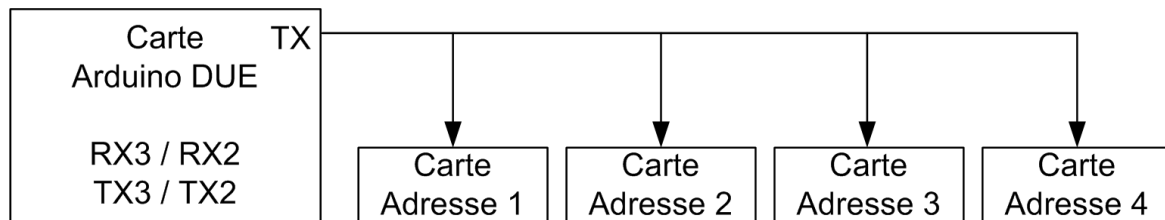
Les voyants sont branchés entre les sorties et le + 5 Volts. Les sorties peuvent fournir 25 mA sous 5 Volts.

Le code est prévu pour brancher, jusqu'à quatre cartes sur une seule sortie TX de l'Arduino principal.

Par défaut, si on utilise qu'une seule carte satellite, elle aura l'adresse 0 et sera toujours sélectionnée.



Si on utilise plusieurs cartes satellite, elles auront une adresse comprise entre 1 et 4 et seront sélectionnées à tour de rôle.



Avant de télécharger le code sur la carte, il faut définir son adresse.

Par défaut, il faut la ligne `const char adresse_carte = 0; "`.

## 2 / Déclaration de l'adresse de la carte dans le code de l'Arduino NANO

Si utilise une seule carte, on déclare le numéro de carte "0" dans le code :

```
const char adresse_carte = 0; // Carte toujours sélectionnée.
```

Quelque soit les caractères envoyés à cette carte, elle restera toujours sélectionnée.

A l'initialisation de l'Arduino, la carte sera automatiquement sélectionnée.

Si l'on veut utiliser plusieurs cartes branchées sur une seule sortie TX, il faut pouvoir les adresser séparément.

On déclare dans le code un numéro de carte différent, pour ces cartes :

```
const char adresse_carte = 1; // Adresse de carte = 1.
const char adresse_carte = 2; // Adresse de carte = 2.
const char adresse_carte = 3; // Adresse de carte = 3.
const char adresse_carte = 4; // Adresse de carte = 4.
```

## 3 / Sélection de la carte depuis l'Arduino principal

Pour sélectionner une des cartes connectées, on envoie un caractère spécial pour sélectionner la carte :

- 17 - pour sélectionner la carte n° 1
- 18 - pour sélectionner la carte n° 2
- 19 - pour sélectionner la carte n° 3
- 20 - pour sélectionner la carte n° 4

Exemple de code pour sélectionner la carte n° 1 : `Serial2.write(17);`

Quand on sélectionne une carte, les autres cartes sont désélectionnées.

Quand on sélectionne une carte, cette carte reste sélectionnée jusqu'à ce que l'on sélectionne une autre carte.

A l'initialisation de l'Arduino, les cartes sont automatiquement désélectionnées.

Pour allumer une sortie, on envoie N° de la sortie + 32. Exemple 34 pour activer la sortie n° 2.

Exemple de code pour allumer le voyant n° 0 : `Serial2.write(32);`

Pour éteindre une sortie, on envoie N° de la sortie + 80. Exemple 83 pour éteindre la sortie n° 3.

Exemple de code pour éteindre le voyant n° 0 : `Serial2.write(80);`

## 4 / Toutes les commandes pour activer les sorties d'une carte

Actions à la réception d'un caractère :

23 => Eteint toutes les 48 lampes, annule le clignotement, sorties = 5 Volts.

24 => Allume toutes les 48 lampes, annule le clignotement, sorties = 0 Volt.

25 => Fait clignoter les 48 lampes.

26 => Positionne le drapeau de clignotement. La prochaine sortie activée sera clignotante.

32 à 79 = Allume la sortie x-32.

32 à 47 = Sorties sur out1.

48 à 63 = Sorties sur out2.

64 à 79 = Sorties sur out3.

80 à 127 = Eteint la sortie x-80.

80 à 95 = Sorties sur out1.

96 à 111 = Sorties sur out2.

112 à 127 = Sorties sur out3.

## 5 / Numérotation des sorties

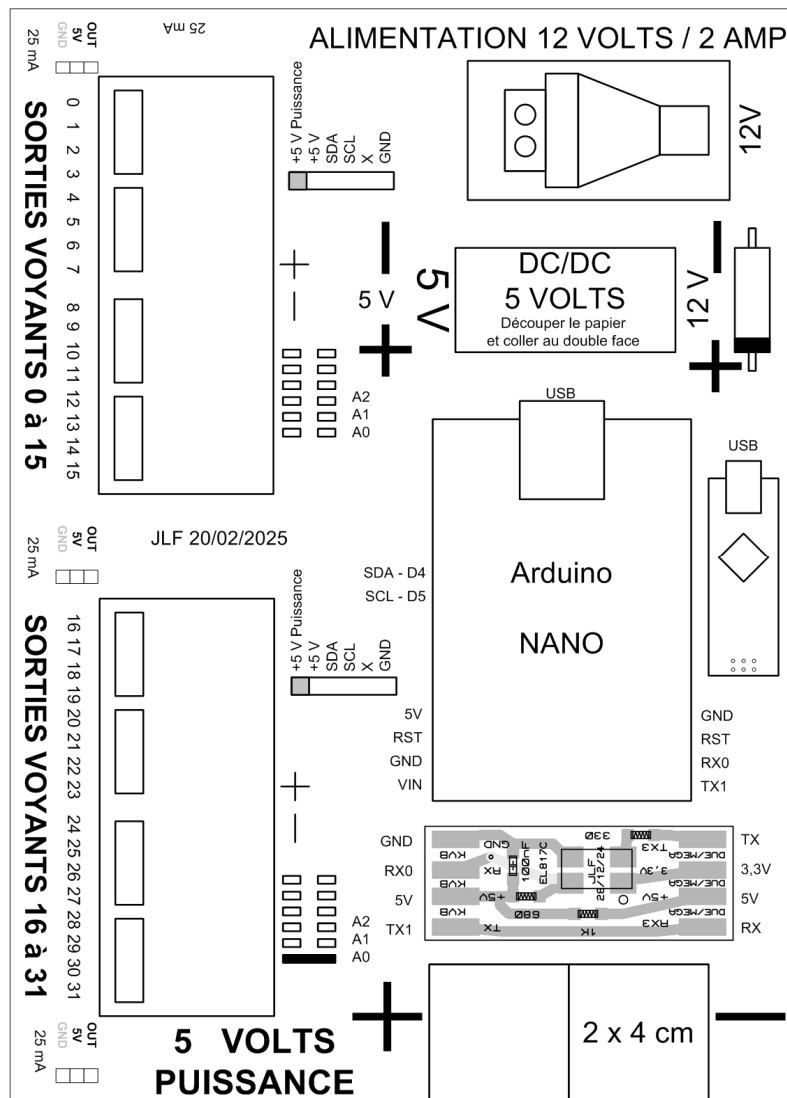
Sorties répartie sur 3 cartes au maximum : OUT0 : 00 à 15    OUT2 : 16 à 31    OUT3 : 32 à 47

## 6 / Divers

Quand le programme fonctionne, la led 'L' clignote sur l'Arduino.

## 7 / Construction

Imprimer le fichier : Voyants déportés JLF - 32 Sorties.pdf



Coller cette feuille sur une plaque. Placer le connecteur 12 Volts, le convertisseur DC/DC 5 Volts et une diode de protection 1N5403.

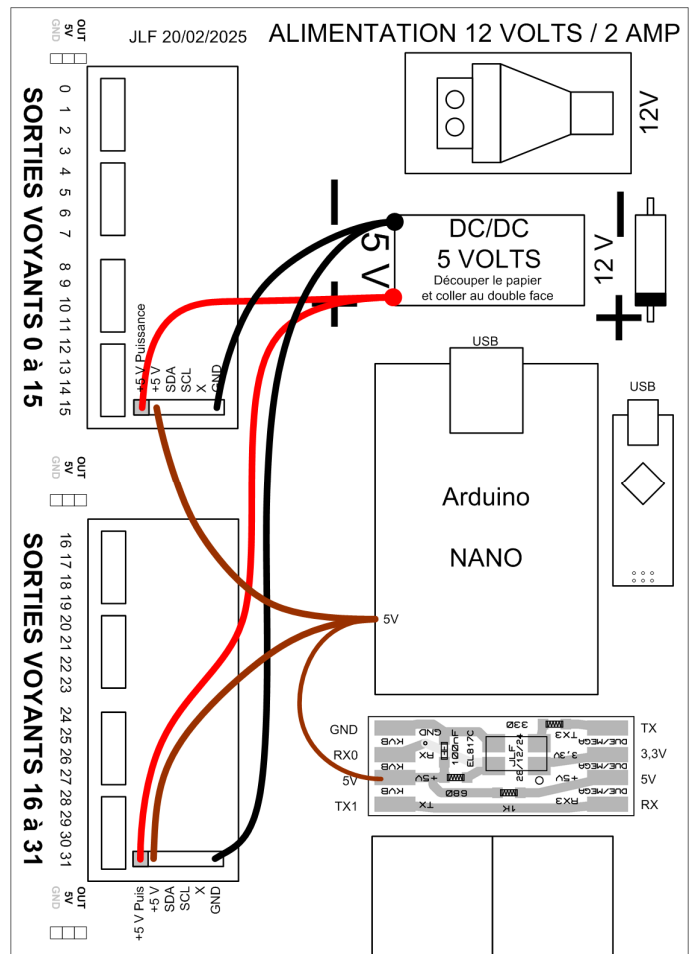
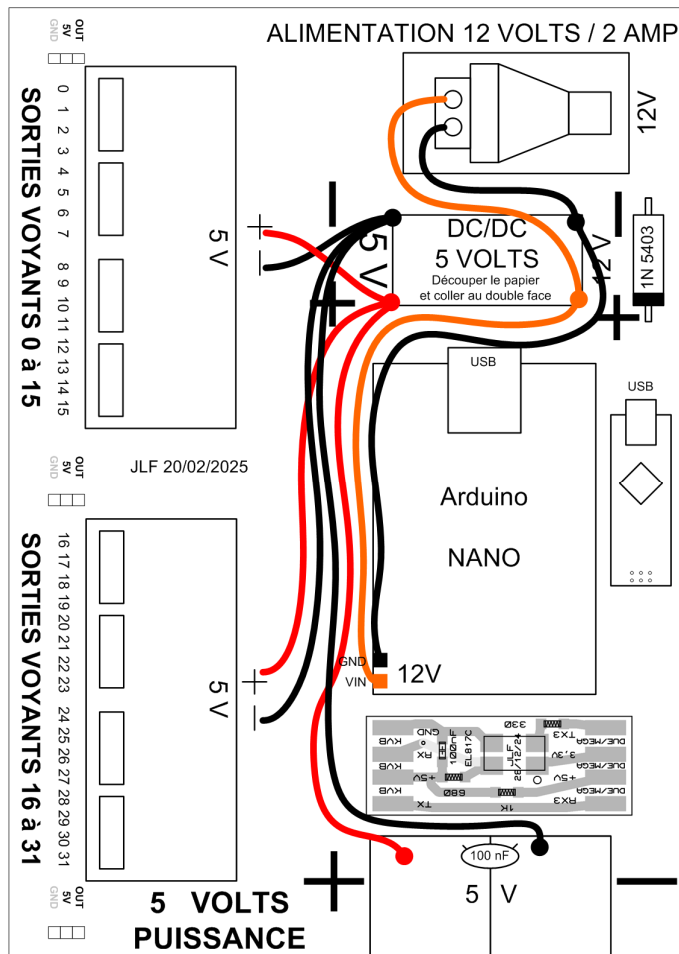
Sur les cartes des sorties pour configurer les adresse I2C, souder les plots **A0** pour la deuxième carte, et souder les plots **A1** pour la troisième carte.

Régler le module DC/DC à 5 Volts, avant de souder les fils vers les autres cartes.

Souder les fils d'alimentation 5 Volts et 12 Volts.

Acheter un câble 6 broches femelle-femelle de 30 cm.

Couper le en deux à bonne distance, pour installer les fils aux bons emplacements.

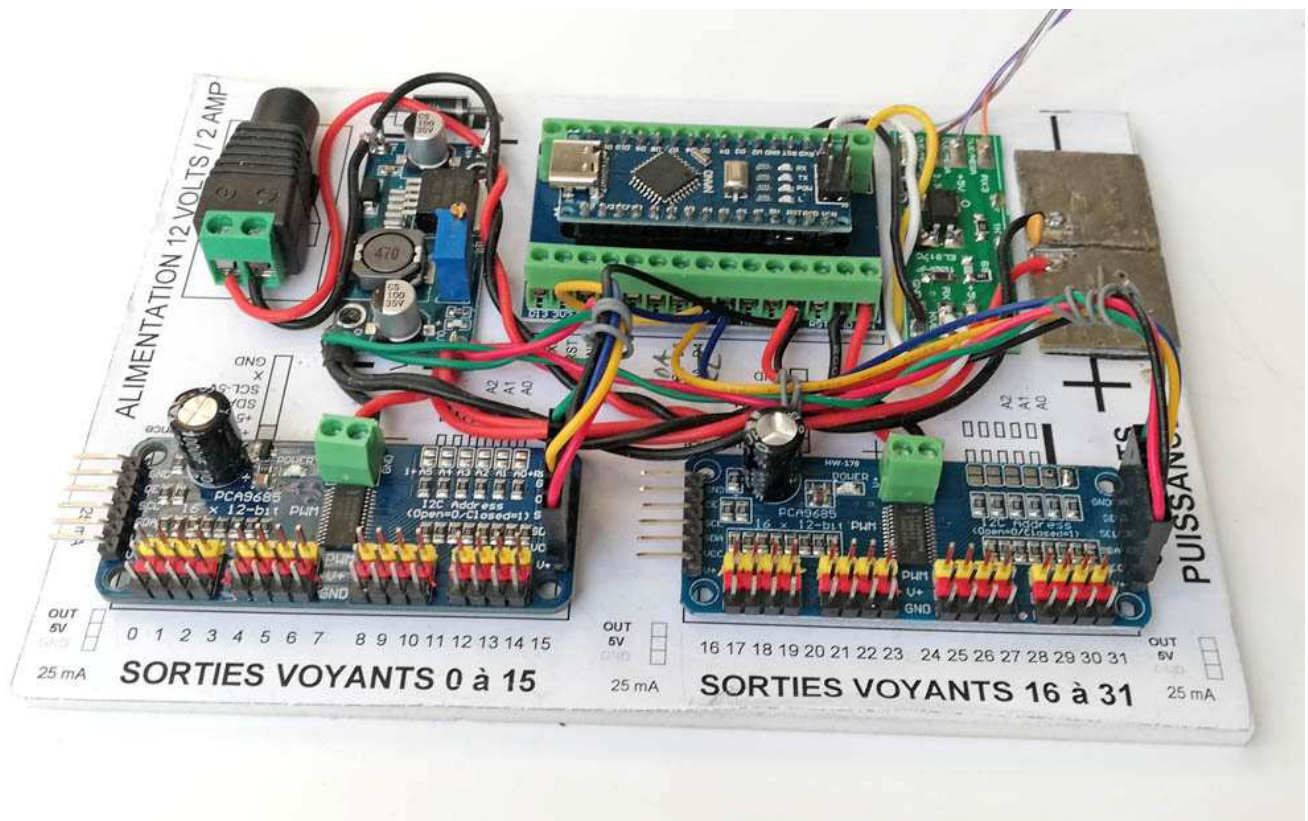
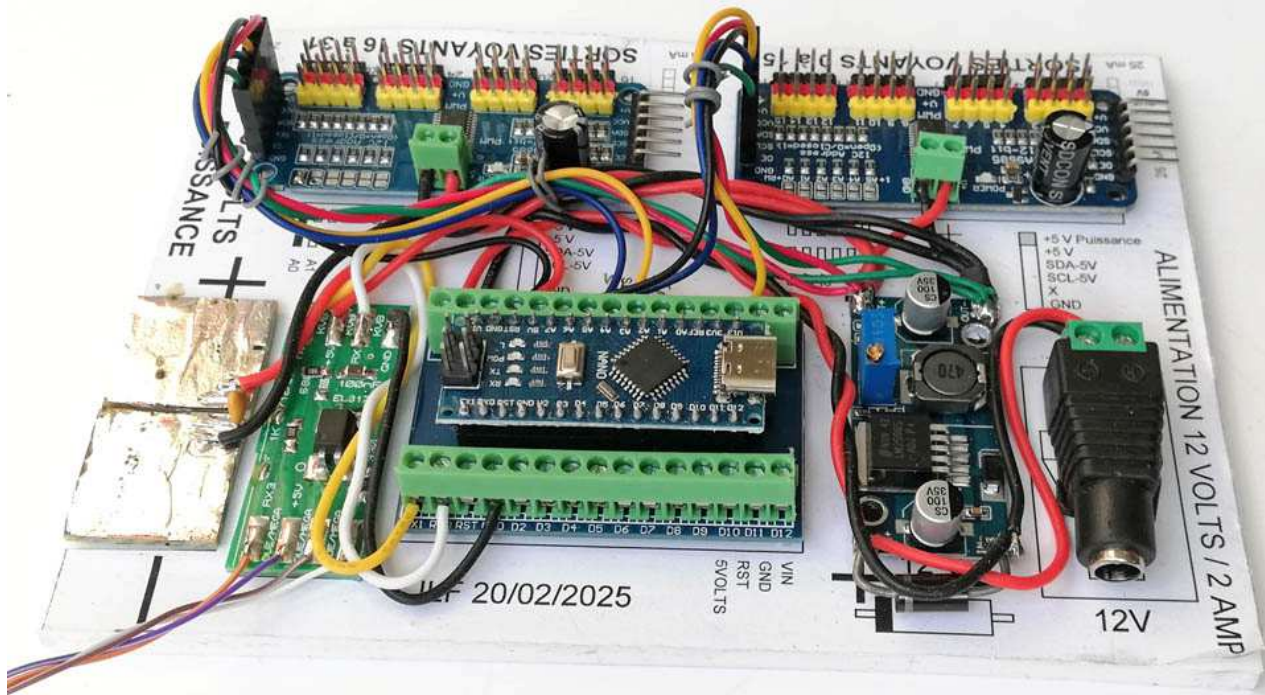


Souder un condensateur de 100 nF sur la plaque 5 Volts.

Souder les fils vers la carte d'isolation électrique.

[illegible]





## 8 / Le code source

Le code source est fourni : "Voyants\_deportes\_JLF.ino".

Il faudra installer la librairie "PCA9685" pour les modules I2C de sorties. PCA9685\_RT de Rob Tillaart (V0.7.2)  
URL : [https://github.com/RobTillaart/PCA9685\\_RT](https://github.com/RobTillaart/PCA9685_RT)

Ou depuis le \*.zip fourni. Menu : Croquis > Inclure une bibliothèque > Ajouter la bibliothèque .ZIP... >

- PCA9685\_RT-master.zip

On donne l'adresse de la carte.

Par défaut, la carte toujours sélectionnée = 0, ou sinon adresse de la carte = 1 à 4, déclarée dans le code :

```
const char adresse_carte = 0;
```

On peut installer une, deux ou trois cartes PCA9685.

Pour activer les cartes n° 2 ou n° 3, il faut supprimer les caractères // en début de ligne.

```
#define FONCTION_OUT2      // Permet d'activer la carte OUT2, si la ligne n'est pas en commentaire.  
// #define FONCTION_OUT3  // Permet d'activer la carte OUT3, si la ligne n'est pas en commentaire.
```

On peut modifier la fréquence de clignotement ici :

```
const long duree_timer_clignotant = 300; // Durée clignotement = 300 msec par défaut.
```

Exemple de code pour commander les voyants, depuis un Arduino DUE distant et le port série n° 2 :

```
Serial2.begin(9600);          // Vers la platine des voyants  
if (Serial2.availableForWrite() > 0) Serial2.write(17);    // Sélectionne la carte n° 1.  
for (x = 0; x < 32 ; x++) {  
    if (Serial2.availableForWrite() > 0) Serial2.write(32+x); // Allume les voyants un par un.  
    delay(200);  
}  
for (x = 0; x < 32 ; x++) {  
    if (Serial2.availableForWrite() > 0) Serial2.write(80+x); // Eteint les voyants un par un.  
    delay(200);  
}  
if (Serial2.availableForWrite() > 0) Serial2.write(21);    // Allume tous les voyants.  
delay(5000);  
if (Serial2.availableForWrite() > 0) Serial2.write(20);    // Eteint tous les voyants.  
delay(5000);  
if (Serial2.availableForWrite() > 0) Serial2.write(22);    // Fait clignoter tous les voyants.  
delay(5000);  
if (Serial2.availableForWrite() > 0) Serial2.write(20);    // Eteint tous les voyants.  
delay(5000);  
if (Serial2.availableForWrite() > 0) Serial2.write(23);  
if (Serial2.availableForWrite() > 0) Serial2.write(32);    // Fait clignoter le voyant n° 0.  
if (Serial2.availableForWrite() > 0) Serial2.write(33);    // Allume le voyant n° 1.  
delay(5000);
```

Pour tester la carte, le moniteur Arduino ne permet pas d'envoyer des caractères invisibles.

Utiliser **Putty**. Choisir la liaison série, à 9600 Bps et le bon port (*Exemple = COM10*).

Ensuite on peut envoyer n'importe quel code. Par exemple pour sélectionner la carte n° 1, taper [Alt]017, et ça envoie le code '17' à l'Arduino. Taper [Alt]032 allume le voyant n° 0.



## 9 / La connexion physique avec une carte Arduino DUE ou MEGA.

On utilisera le montage d'isolation électrique à base d'optocoupleur.

J'ai repris mes circuits d'isolation existants. On n'utilise que la partie Arduino DUE TX, vers Arduino Nano RX.

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

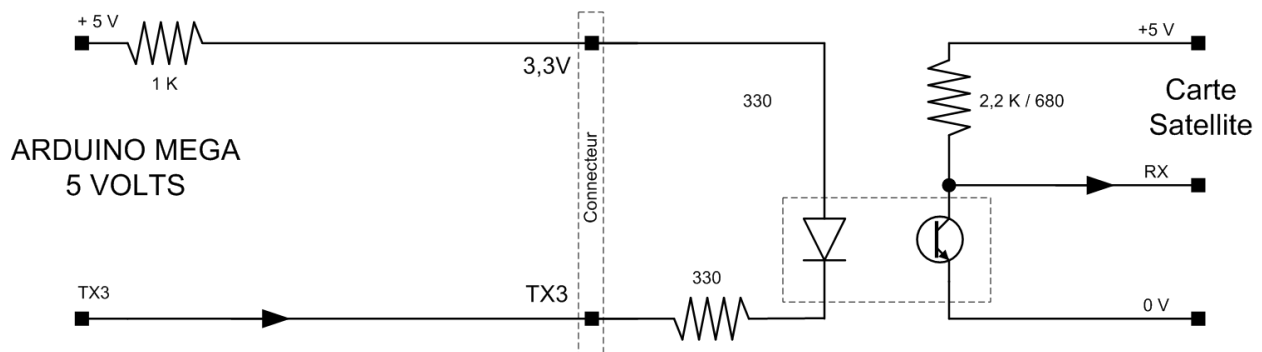
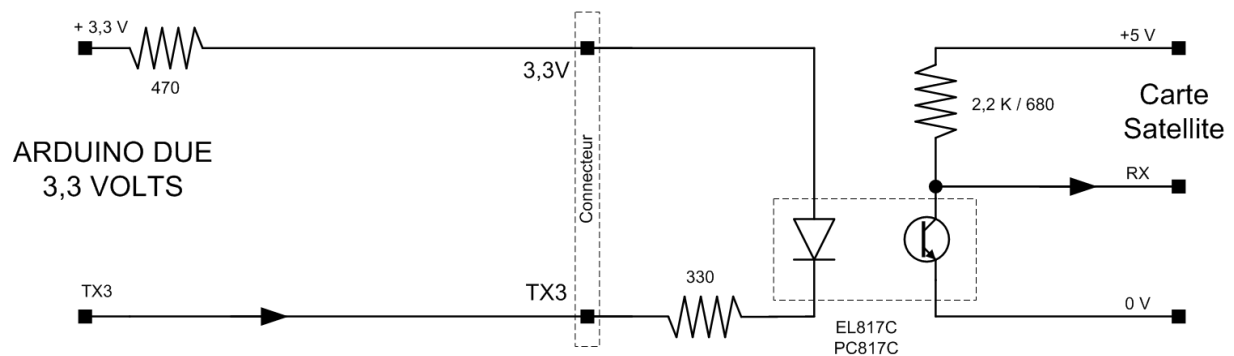
Pour d'autres explications, voir le document KVB : [http://www.la-tour.info/uts/uts\\_page15.html#kvb](http://www.la-tour.info/uts/uts_page15.html#kvb)

Schémas suivant la tension de fonctionnement :

Sur l'entrée RX, souder une résistance de 2,2 K Ohms pour un Arduino NANO (A la place de la 680 Ohms).

Il y a déjà une résistance de 1 K Ohms en entrée sur une carte NANO, ce qui donne 680 Ohms en parallèle avec la 2,2K.

Par contre sur une entrée RX, qui n'a pas de résistance de rappel (Entrée RX3 d'un Arduino MEGA), il faudrait une résistance de 680 Ohms.

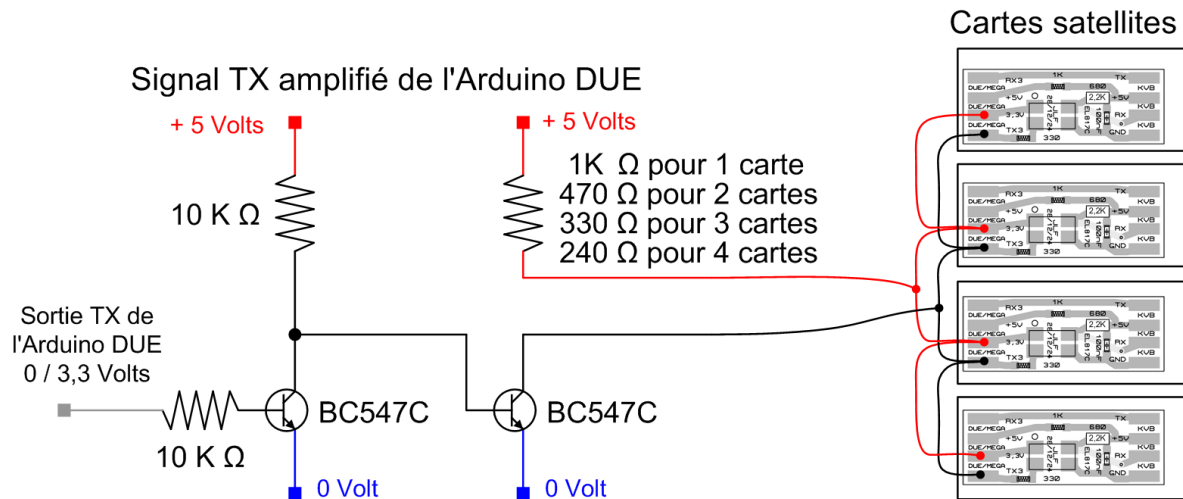


Pour une seule carte satellite, il n'y a pas besoin de ce montage amplificateur.

Si l'on veut brancher plusieurs cartes satellites à une sortie TX d'un Arduino, il faut amplifier ce signal.

Ne pas mettre du 5 Volts sur une entrée de l'Arduino DUE, sous peine de le griller.

La résistance comprise entre 240 et 1 K Ohms, dépend du nombre de carte branché.



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://jlcpcb.com>

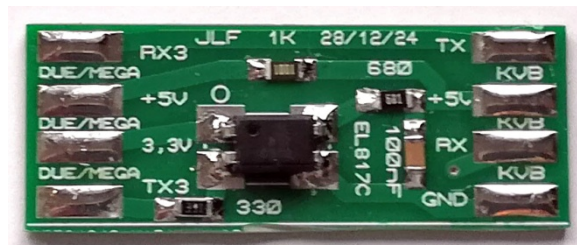
#### Composants :

- 1 x Optocoupleur EL817c
- 330, 470, 1K, 2,2K Ohms cms (format 1206)
- 1 x 100 nF cms (format 1206)

#### Fabrication du circuit d'isolation :

Il y a déjà une résistance de 1 K en entrée RX sur l'Arduino NANO. Ajouter une résistance de 2,2 K suffit en entrée RX d'un Arduino NANO. Souder une 2,2 K à la place de la 680 Ohms.

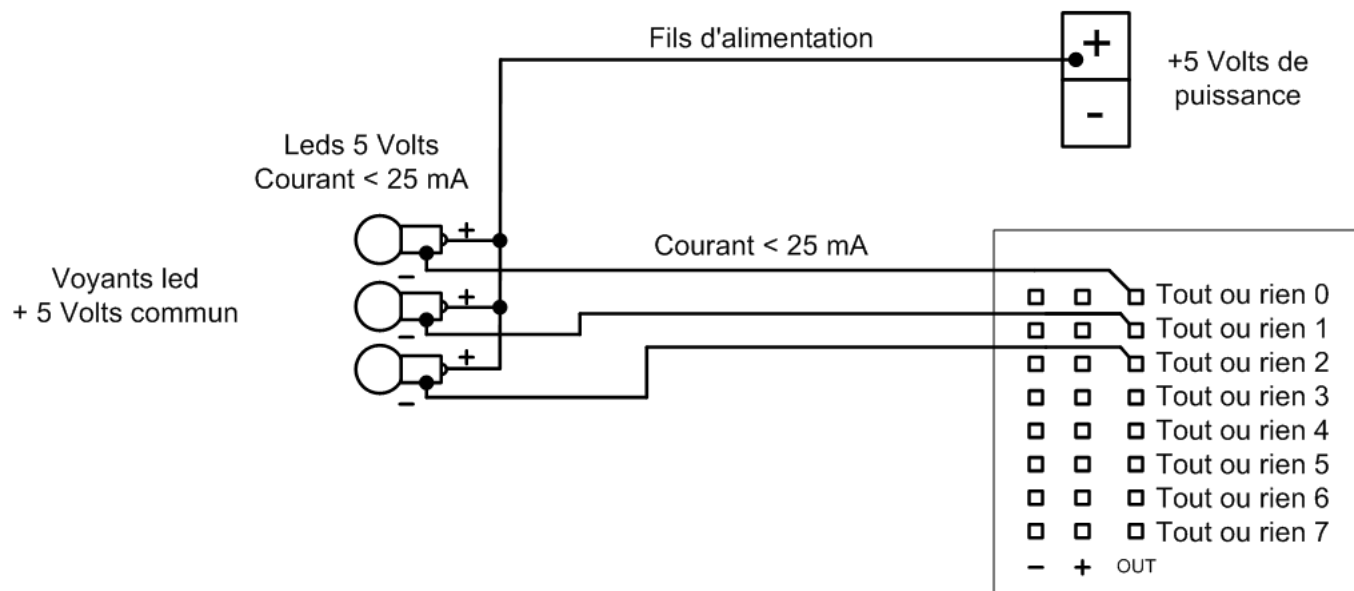
Souder tous ces composants. Repérer le rond '●' sur l'optocoupleur, pour l'aligner avec celui sur le circuit imprimé.



#### 10 / Branchement des voyants.

Les voyants ou autres équipements ne doivent pas consommer plus de 25 mA sous 5 Volts.

Si l'on commande des relais, il faut obligatoirement placer une diode anti-retour sur leur bobine.



A+