

# MIGRATION DU COMPTEUR DE VITESSE LCD

Le 12/05/2025

Ce document explique comment animer le compteur de vitesse LCD, d'une ancienne cabine de conduite d'une locomotive.

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

C'est pour animer un véritable pupitre de locomotive, utilisé avec un programme de simulateur sur ordinateur.



Poser vos questions sur le forum RMF, par exemple ici : <https://www.rmfmagazine.com/phpBB/viewtopic.php?t=203075>

**Attention, avec une tension d'alimentation de 72 Volts, (>48 Volts) ces manipulations sont dangereuses !**

Si l'on veut alimenter ce compteur en +12 Volts, on doit pouvoir remplacer le module d'alimentation interne par un 20 IMY 15-05-8R.

La notice de ce compteur se trouve sur le site Logiplus :

- <https://www.logiplus.com/fr/nos-produits/indicateurs-de-vitesse-sncf>
- <https://www.logiplus.com/wp-content/uploads/2020/05/FeuilleCatalogueIVAtessLCDRev7.pdf>

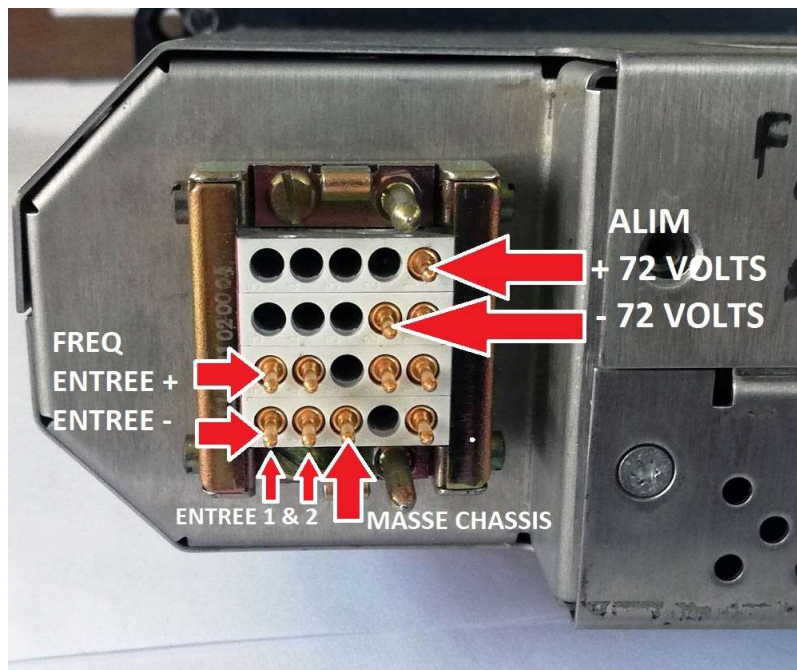
Il existe différents types de compteur de vitesse Logiplus. Les deux compteurs présentés ici, sont un modèle EQ-1112-008, pour l'application ATESS sous 72 Volts, et un modèle EQ-1112-001, pour l'application ATEC sous 72 Volts.

## 1 / MODELE : ATESS - 72 VOLTS - EQ-1112-008 : Connecteur d'entrée à l'arrière

Le compteur ATESS comporte un connecteur avec une série de broches.

L'alimentation 72 Volts utilisera deux broches (+)(-).

Il y a deux entrées pour les signaux de vitesse. Chaque entrée est complètement isolée électriquement de l'autre entrée et du reste du montage. Chaque entrée est polarisée et utilise deux broches (+)(-).



## Branchement des interfaces du boîtier

### Alimentation

On alimente le boîtier avec une tension adaptée au modèle, ici 72 Volts sous 100 mA.

**En fait, on peut alimenter ce compteur avec une alimentation 48 Volts sous 150 mA, ça fonctionne.**

### Entrées pour les signaux de vitesse.

On amène un signal d'amplitude 0-15 Volts et de fréquence variable, sur les deux entrées.

La masse du signal sur les deux broches (-), et le signal sur les deux broches (+) (*Voir doc Logiplus*).

## Fonctionnement du compteur

### Alimentation

Attention, avec une tension d'alimentation de 72 Volts, (>48 Volts) ces manipulations sont dangereuses !

Ce boîtier peut être alimenté entre  $7 \cdot U_{nominal}$  et  $1,25 \cdot U_{nominale}$ , soit entre 50,4 Volts et 90 Volts.

**En fait, on peut alimenter ce compteur avec une alimentation 48 Volts sous 150 mA, ça fonctionne.**

Mais le compteur d'arrête si la tension descend en dessous de 45 Volts. Monter la tension à 50 Volts si on rencontre des problèmes.

**Pour le modèle 72 Volts, il est préférable d'alimenter ce boîtier sous 48 Volts, pour réduire les risques au contact électrique.**

Donc alimenter ce boîtier en 48 Volts, sous 150 mA minimum.

A ce moment là, le compteur reste éteint, rien ne s'allume, c'est son fonctionnement normal.

En appuyant plusieurs fois sur la touche (+), l'éclairage des boutons (-) et (+) augmente, mais l'écran restera noir.

C'est le signe que l'alimentation du boîtier est en bonne état.

#### Signaux d'entrées

Il faut amener sur les deux entrées un signal de 80 Hz 0-15 Volts, pour que l'écran affiche "0".

La résistance d'entrée est de 900 Ohms, soit un courant par entrée de 16 mA, ou 32 mA pour les deux entrées en parallèle.

Si le signal est < 80 Hz ou > 3880 Hz, l'écran restera noir.

Si le signal est trop faible en voltage, l'écran restera noir.

Les entrées sont isolées du reste du montage. La carte embarque des optocoupleur HP2232 sur les deux entrées.

Il y a une diode en inverse et une résistance de 2 K en parallèle sur la photodiode. Il y a une résistance de 1,7 K en série avec la photodiode.

Sur le connecteur externe du boîtier, on trouve deux broches qui seront reliées ensemble en cas de panne du boîtier (Contact d'un relais).

On trouve aussi deux broches pour commander l'éclairage. A priori, ça ne sert pas.

On a deux entrées pour deux capteurs de vitesse. Par sécurité, le boîtier calcule la vitesse pour les deux entrées, et n'affichera une vitesse, que si les deux signaux d'entrée indiquent la même vitesse. Le même signal peut être appliqué sur les deux entrées.

La fréquence du signal d'entrée est égal à  $= 80 \text{ Hz} + 10\text{Hz par km/h}$ .

Soit 80 Hz à l'arrêt, 1080 Hz à 100 Km/h, 3280 Hz à 320 Km/h et 3880 Hz à 380 km/h.

La vitesse maximum affichable est 380 km/h.

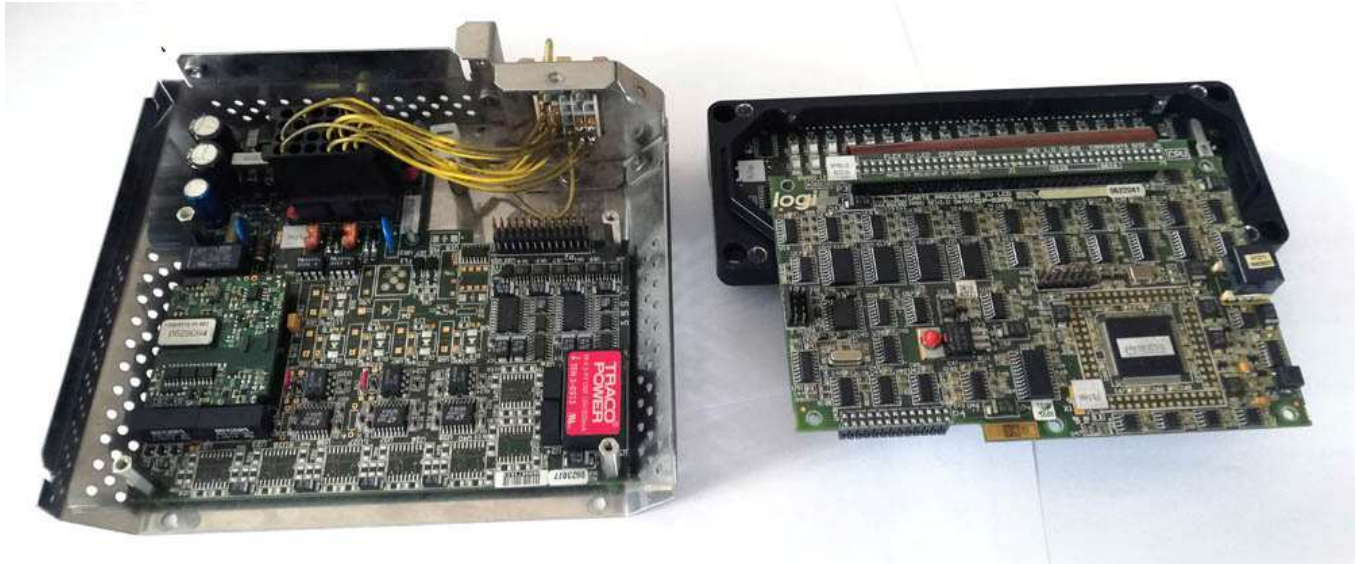


## Modification optionnelle pour réduire la tension de 15 à 5 Volts les signaux d'entrée

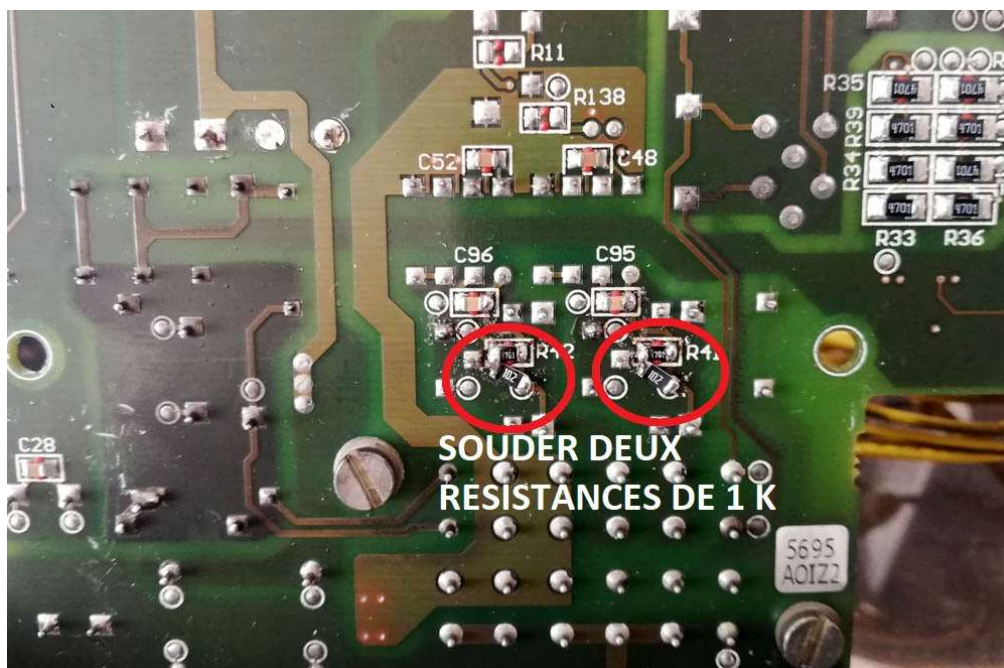
J'ai ajouté deux résistances de 1 K Ohms, soudé sur le circuit imprimé, pour que le compteur fonctionne avec un signal 0-5 Volts, au lieu de 0-15 Volts.

Cette opération n'est pas obligatoire, elle permet simplement d'abaisser la tension des signaux d'entrée de 15 à 5 Volts, pour commander ensuite ce compteur par un Arduino.

Le boîtier se démonte facilement. Il y a un fusible sur la carte du bas, près des gros condensateurs.



Sous le circuit imprimé du bas, on soude deux résistances cms format 1206 de 1 K Ohms.



Une fois soudé, on teste à l'ohmmètre ces résistances. Si elles ont bien soudées, on doit trouver environ 500 à 700 Ohms.

Attention une fois cette modification réalisée, et en mettant les deux entrées en parallèle, le courant d'entrée sera de alors de 13 mA sous 5 Volts.

On ne pourra pas brancher directement la sortie d'un Arduino sur cette entrée, il faudra passer par un transistor amplificateur.

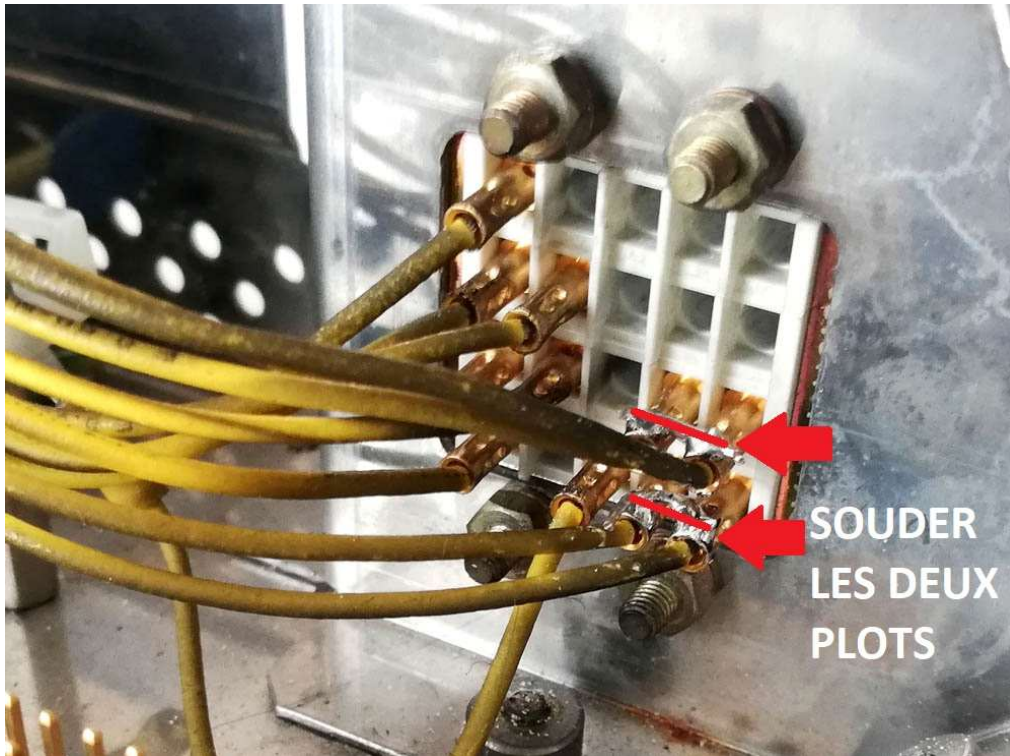
## Modification optionnelle pour relier les entrées ensemble

Pour une utilisation sur un simulateur de loisir, j'ai relié les deux entrées ensemble à l'intérieur du boîtier.

Cette modification permettra de n'utiliser qu'une entrée du connecteur sur les deux. Ca limitera le câblage externe.

J'ai relié par un fil, les deux plots du haut ensemble.

J'ai relié par un deuxième fil, les deux plots du bas ensemble.



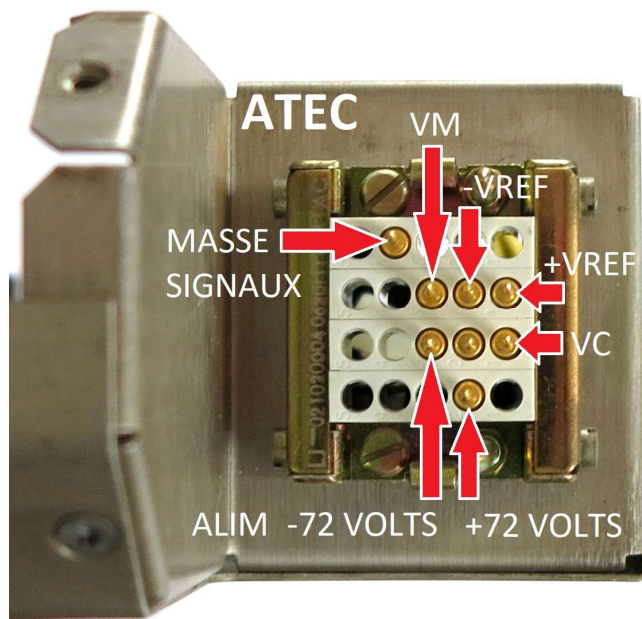
Comme cela, en envoyant le signal sur une seule entrée, le compteur fonctionnera à tous les coups.

## 2 / MODELE : ATEC 220-03/04 - EQ-1112-001 - 72 VOLTS : Connecteur d'entrée à l'arrière

Ce compteur ATEC comporte un connecteur avec une série de broches.

L'alimentation 72 Volts utilise les broches 72 V (+)(-).

Il y a quatre entrées pour les signaux de vitesse.



### Branchement des interfaces du boîtier

#### Alimentation

On alimente le boîtier avec une tension adaptée au modèle, ici 72 Volts sous 100 mA.

**En fait, on peut alimenter ce compteur avec une alimentation 48 Volts sous 150 mA, ça fonctionne.**

#### Entrées pour les signaux de vitesse.

+VREF = Tension continue 10,75 Volts.

-VREF = 0 Volt = Sur la patte "Masse des Signaux".

VC et VM sont branchées ensemble. Leur potentiel par rapport à la broche "Masse des Signaux", donne la vitesse.

- 0 Volt = 0 Km/h.
- 4 Volts = 100 Km/h.
- 10 Volts = 250 Km/h.

### Fonctionnement du compteur

#### Alimentation

Attention, avec une tension d'alimentation de 72 Volts, (>48 Volts) ces manipulations sont dangereuses !

Ce boîtier peut être alimenté entre  $7 \cdot U_{nominal}$  et  $1,25 \cdot U_{nominale}$ , soit entre 50,4 Volts et 90 Volts.

**En fait, on peut alimenter ce compteur avec une alimentation 48 Volts sous 150 mA, ça fonctionne.**

Mais le compteur s'arrête si la tension descend en dessous de 45 Volts. Monter la tension à 50 Volts si on rencontre des problèmes.

Alimenter ce boîtier en 48 Volts, sous 150 mA minimum.



A ce moment là, le compteur reste éteint, rien ne s'allume, c'est son fonctionnement normal.

En appuyant plusieurs fois sur la touche (+), l'éclairage des boutons (-) et (+) augmente, mais l'écran restera noir.

C'est le signe que l'alimentation du boîtier est en bonne état.

### Signaux d'entrées

Il faut amener une tension fixe de 10,75 Volts sur la broche +VREF, et 0 Volt sur la broche -VREF.

Ensuite, une tension continue sur les broches VC et VM donne la vitesse. Ce compteur compte jusqu'à 10 Volts = 250 Km/h.

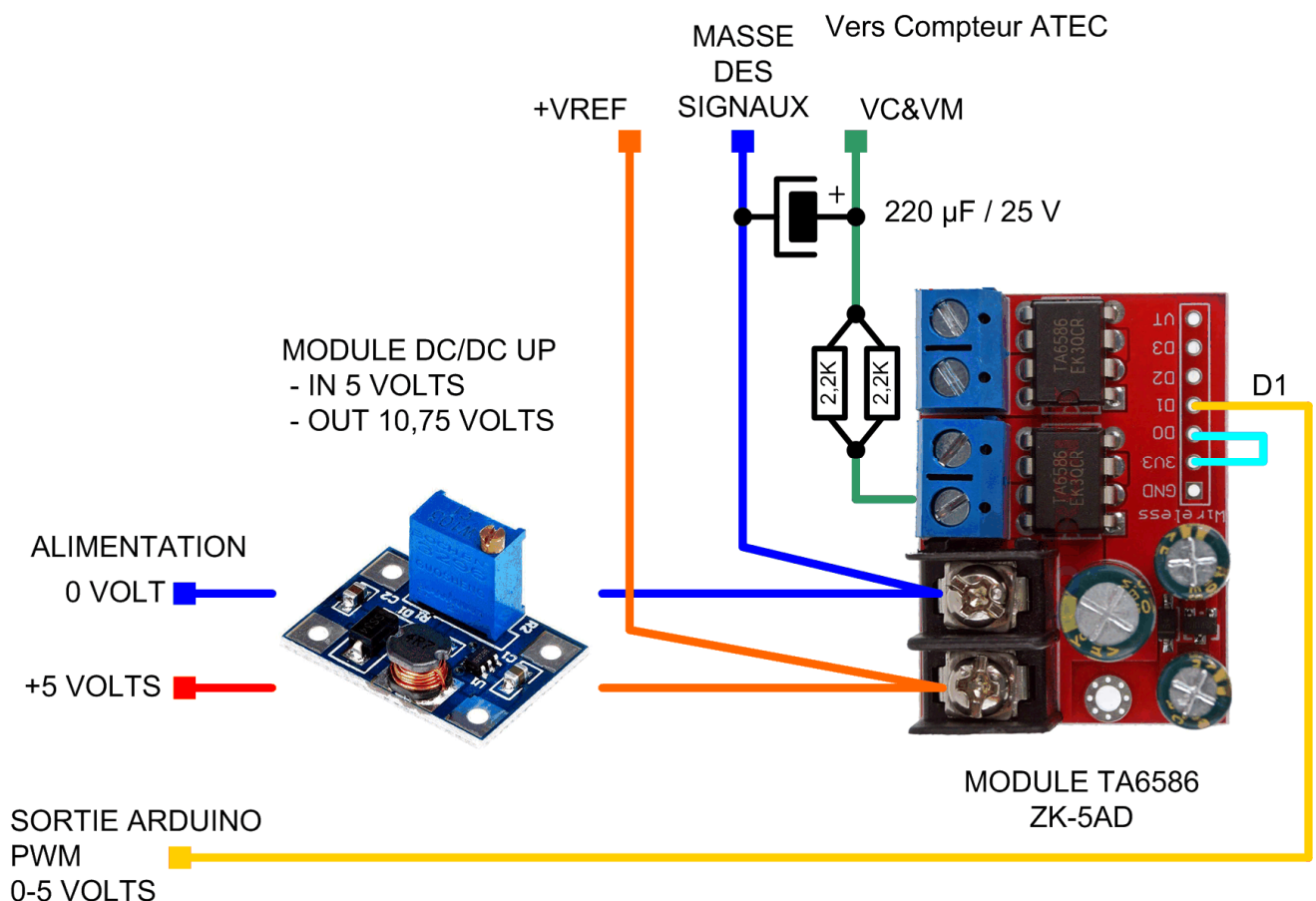
Voici un exemple de branchement à une carte Arduino, pour obtenir un signal 0 à 10 Volts vers le compteur.

Le signal en sortie d'Arduino est obligatoirement de type PWM, et non pas une tension continue.

Le signal est inversé, et 100 % de cycle donnera 0 km/h.

Avant de brancher le module TA6586, régler le module DC/DC élévateur de tension à **10,75** Volts en sortie.

Les résistances de 2,2 K et le condensateur de 220  $\mu$ F lisse la tension en entrée du compteur. C'est obligatoire, car c'est un signal rectangulaire qui sort du module, et sans ce filtre le compteur affichera une valeur fluctuante.



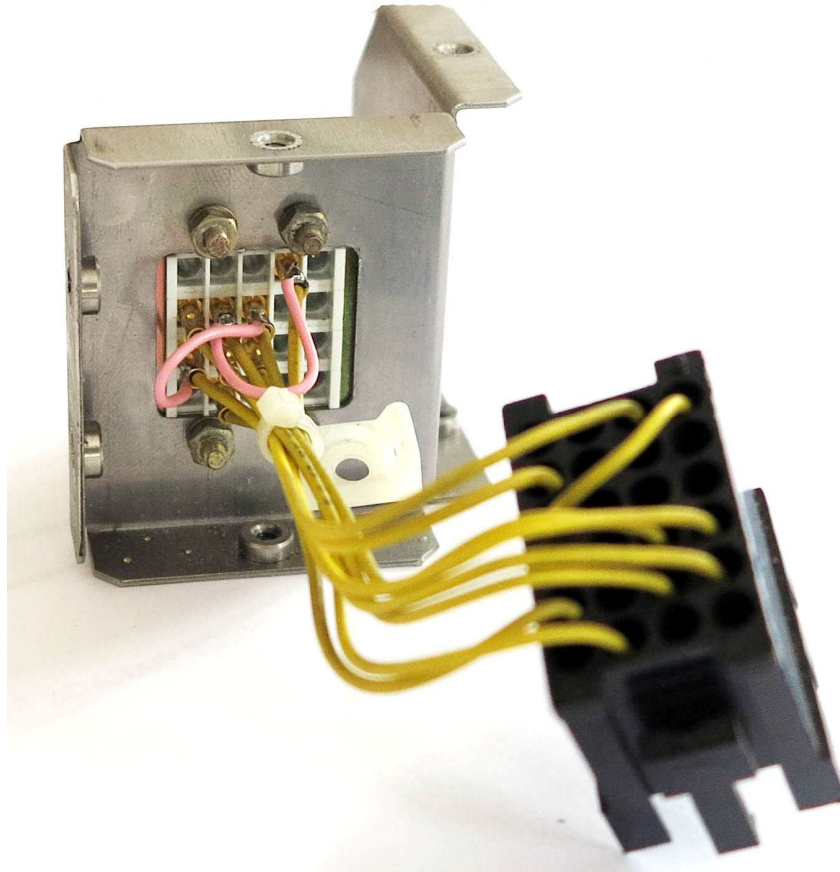
## Modification optionnelle pour relier des entrées ensemble

Pour une utilisation sur un simulateur de loisir, j'ai relié des entrées ensemble à l'intérieur du boîtier.

Cette modification permettra d'utiliser moins de broche du connecteur, et facilite les branchements.

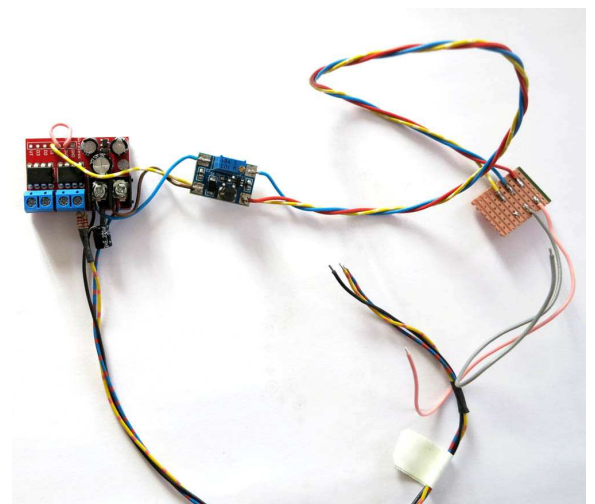
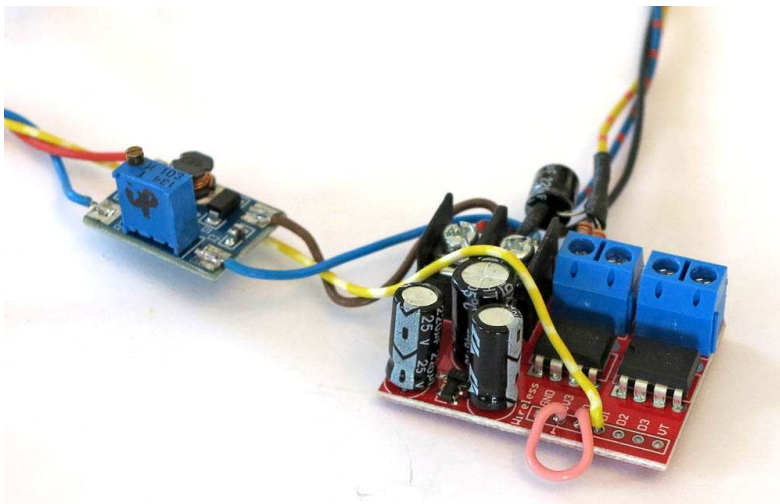
J'ai relié ensemble les plots "Masse des signaux" et -VREF.

J'ai relié ensemble les plots VC et VM.



Comme cela, en envoyant 10,75 Volts sur +VREF et une tension de 0 à 10 Volts sur VC, le compteur fonctionnera à tous les coups.

Exemple de montage, pour le brancher le compteur ATEC en sortie PWM d'Arduino.





### 3 / Dépannage

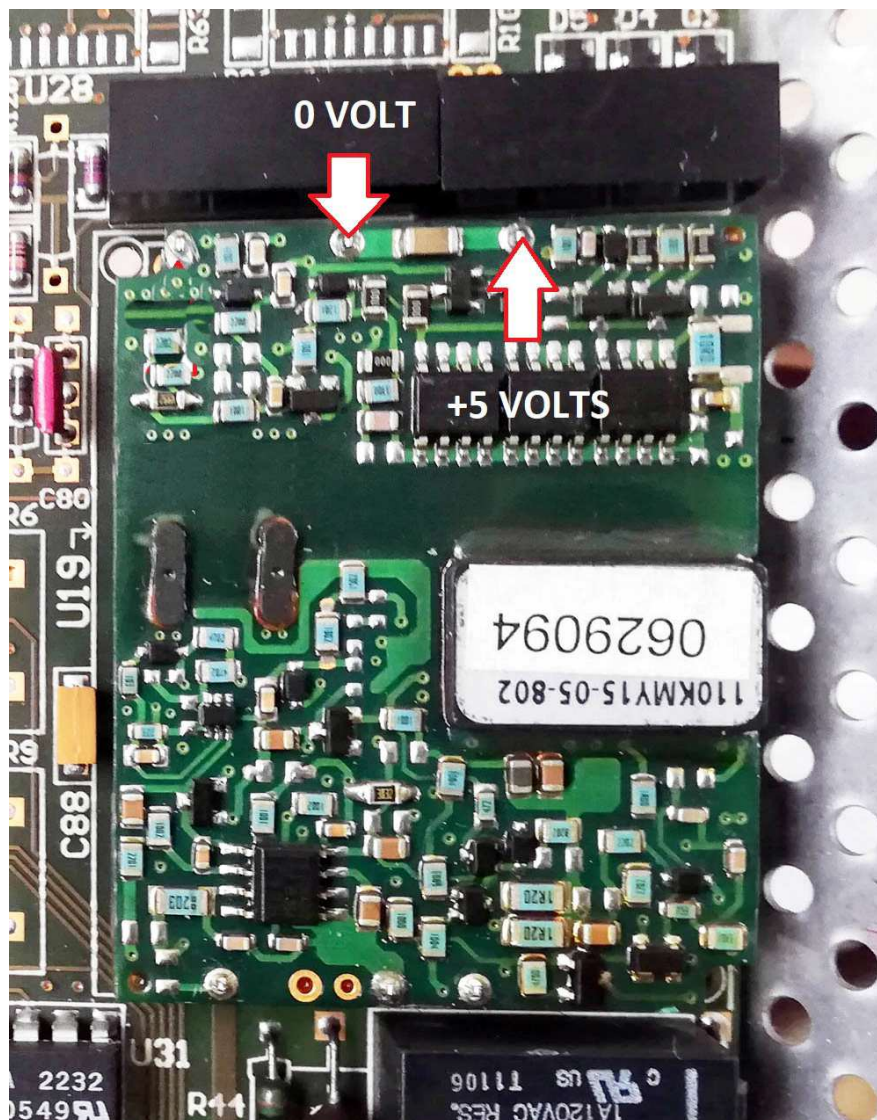
Attention, avec une tension d'alimentation de 72 Volts, (>48 Volts) ces manipulations sont dangereuses.

**En fait, on peut alimenter ce compteur avec une alimentation 48 Volts 150 mA, ça fonctionne.**

Remarque : Le boîtier n'affiche rien tant qu'un signal de 80 Hz ou 10,75 V ne sera pas branché sur les entrées !

Si l'appui sur la touche (+) éclaire les touches, le boîtier n'est peut être pas en panne, mais les signaux d'entrée ne sont peut être pas corrects. Sur un boîtier ATESS non modifié, ce signal carré fait 15 Volts d'amplitude.

On laisse uniquement la carte du bas sur ses supports. En alimentant cette carte, on trouve du + 5 Volts en sortie de la carte d'alimentation.



En cas de panne, racheter un module : 110 IMY 15-05-8R (Alimentation 50 à 150 Volts), ou 40 IMY 15-05-8R (Alimentation 16 à 75 Volts), ou 20 IMY 15-05-8R (Alimentation 9 à 36 Volts).

Ou acheter un module DC/DC 12 Volts vers 5 Volts 3 Ampères, et alimenter le boîtier en 12 Volts.

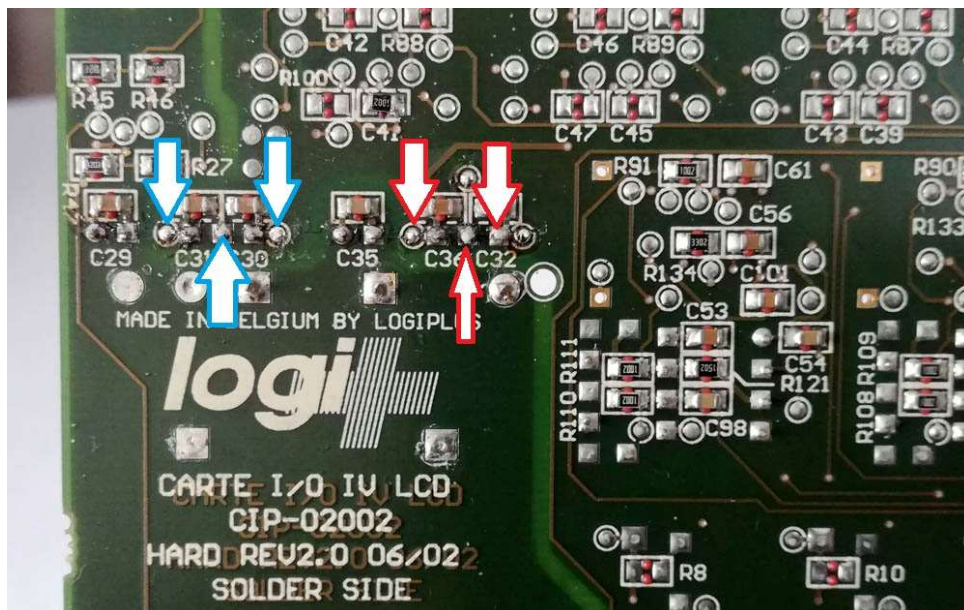
Et il y a aussi des alimentations annexes.

Ces boîtiers RECOM RD-051D alimentés en 5 Volts, fournissent en sortie du -15 et +15 Volts.



Sous le circuit imprimé du bas, aux bornes de :

- C29 on a 5 Volts
- C30 on a 15 Volts
- C31 on a 15 Volts
- C35 on a 5 Volts
- C32 on a 15 Volts
- C36 on a 15 Volts



En cas de panne, racheter des modules : NMH0515SC.

#### 4 / L'affichage en fonctionnement

On a aussi l'affichage d'un repère à droite de l'écran pour l'accélération en cours.



En cas d'anomalie, ou de signal hors spécification, un carré hachuré est allumé à gauche de l'écran.

Sur ces compteurs, il est très faiblement allumé.





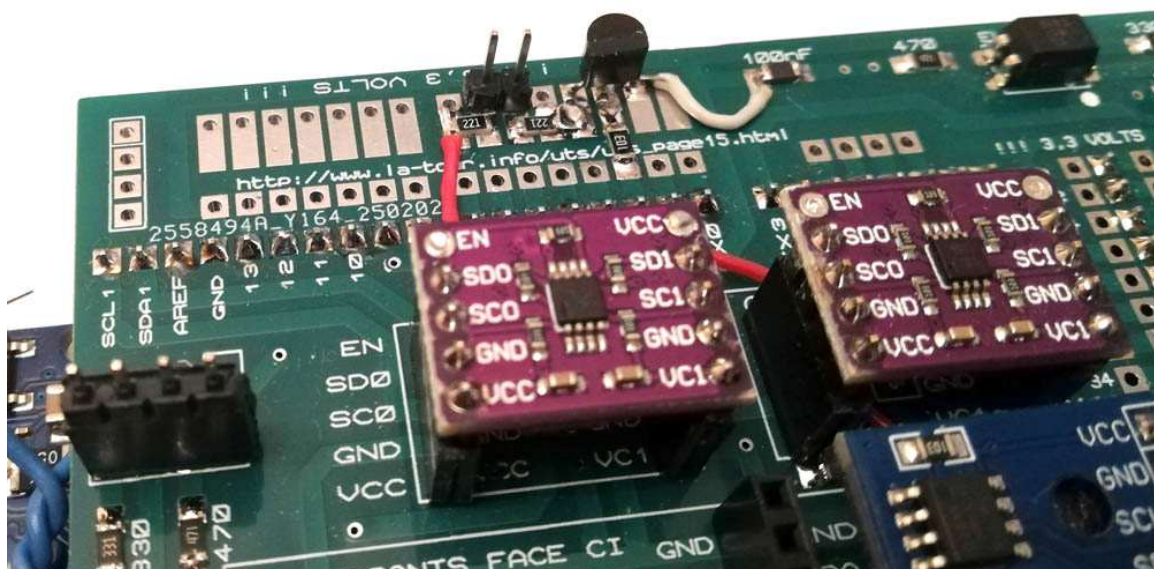
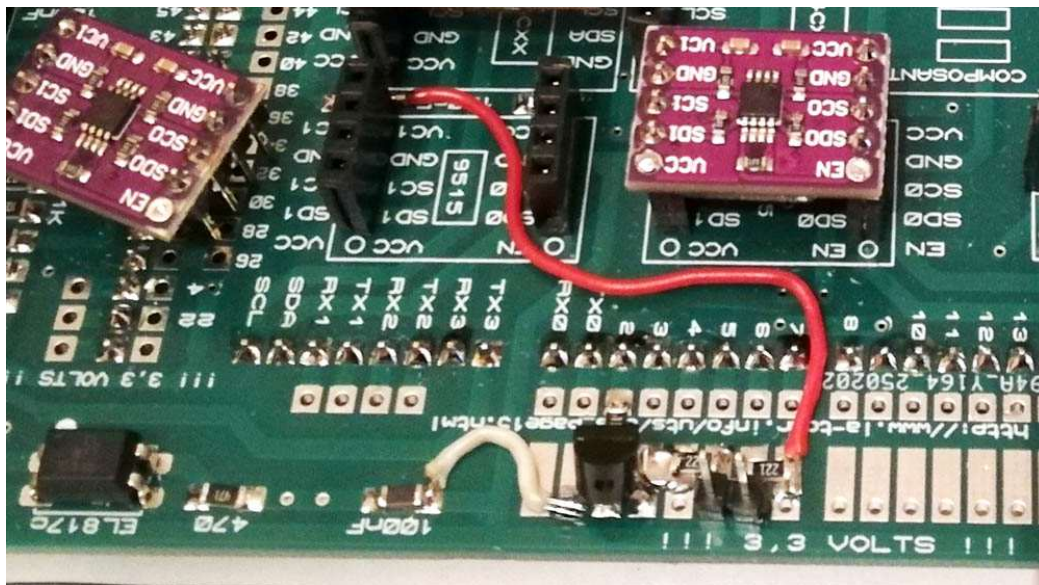
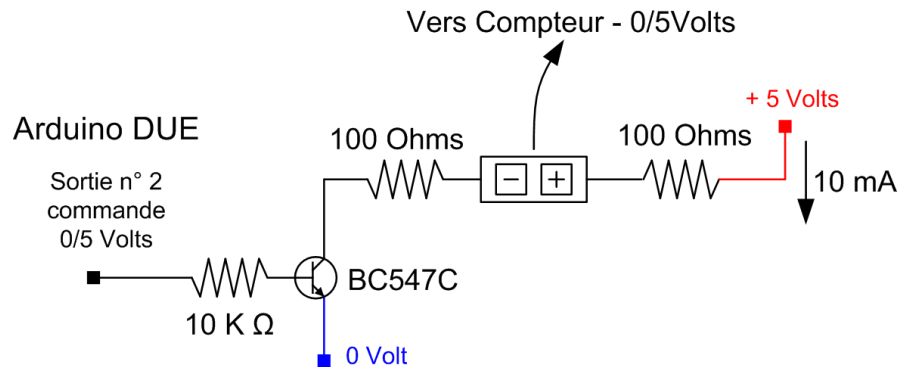
## 5 / Exemple de branchement du modèle ATESS sur un Arduino DUE

Sur un montage comme le mien : [https://www.la-tour.info/uts/uts\\_page15.html#conduite](https://www.la-tour.info/uts/uts_page15.html#conduite)

On ajoute un transistor en sortie. Les résistances de 100 Ohms protègent le montage.

Attention, ne pas appliquer une tension > 3,3 Volts sur les broches de l'Arduino DUE sous peine de le griller instantanément.

Fréquence de sortie = 80 Hz à 3880 Hz. Fréquence générée = 80 Hz + 10 \* Vitesse (km/h).



J'ai soudé des 220 Ohms, mais des 100 Ohms c'est plus sûr pour faire fonctionner le compteur modifié pour 5 volts.



## 6 / Code de génération du signal pour un Arduino DUE pour le modèle ATESS

C'est un sous programme pour générer un signal de fréquence variable, pour animer le compteur de vitesse LCD.

Source : [https://github.com/aethaniel/ExperimentalCore-sam/blob/master/module/cores/arduino/port\\_sam/core\\_tone.cpp](https://github.com/aethaniel/ExperimentalCore-sam/blob/master/module/cores/arduino/port_sam/core_tone.cpp)

Fréquence de sortie = 80 Hz à 3880 Hz. Fréquence générée = 80 Hz + 10 \* Vitesse (km/h).

### Déclaration des variables :

```
unsigned int vitesse_lcd;           // De 0 à 380 km/h.
unsigned int vitesse_lcd_old;       // Vitesse sauvegardée.
unsigned int vitesse_lcd_atess;     // Vitesse au format ATESS = signal de 80 Hz à 3280 Hz.
```

### Dans la fonction setup() :

```
configureTimer();                 // Initialise la pin n°2.
vitesse_lcd = 0;
vitesse_lcd_old = 0;
vitesse_lcd_atess = 80;           // Signal ATESS 80 Hz pour 0 km/h.
setFrequencytone(vitesse_lcd_atess);
```

### Dans la fonction loop() :

```
if (vitesse_lcd != vitesse_lcd_old) {           // Si la vitesse a bougé, on rafraichi le compteur.
    vitesse_lcd_old = vitesse_lcd;
    vitesse_lcd_atess = 80 + (vitesse_lcd*10); // Fréquence envoyée au compteur LCD = 80 hz + 10 Hz par km/h.
    setFrequencytone(vitesse_lcd_atess);
}
```

### En réception des données du simulateur :

```
// < :SpeedometerKPH:160.00> ==> Sortie pwm 4
else if (!strcmp(buffer_rx[1],"SpeedometerKPH")) { //
    buffer_rx_fvaleur = atof(buffer_rx[2]);         // Conversion du dernier champ : Ascii-> Flottant.
    vitesse_lcd = int(0.996*(buffer_rx_fvaleur+0.5)); // Pour un compteur de vitesse de type LCD.
    Correction pour une fréquence de sortie exacte.
}
```

### Sous programme en fin de programme :

```
//-----
// Sous programme pour générer un signal de fréquence variable, pour animer le compteur de vitesse LCD.
// F = 80 Hz à 3880 Hz. Fréquence générée = 80 Hz + 10 * Vitesse(km/h).
// Source : https://github.com/aethaniel/ExperimentalCore-sam/blob/master/module/cores/arduino/port_sam/core_tone.cpp
#ifdef FONCTION_CPT_VITESSE_LCD // Compilation conditionnelle. Permet de sortir un signal à fréquence variable vers
le compteur de vitesse LCD.
#define TONE_PIN 2 // TIOA0 - Sortie du signal en patte n° 2.
static Tc *chTC = TC0;
static uint32_t chNo = 0;
void configureTimer(void)
{
    // Configure TONE_PIN pin as timer output, pmc_enable_periph_clk( ID_PIOB );
    int result = PIO_Configure( PIOB, PIO_PERIPH_B, PIO_PB25B_TIOA0, PIO_DEFAULT);
    pmc_set_writeprotect(false);
    pmc_enable_periph_clk(ID_TC0);
    TC_Configure(chTC, chNo,
        TC_CMR_TCCLKS_TIMER_CLOCK3 | // MCK/32 + TIMER_CLOCK3 + frequency << 5.
        TC_CMR_WAVE | // Waveform mode
        TC_CMR_WAVSEL_UP_RC | // Counter running up and reset when equals to RC
        TC_CMR_ACPA_SET | // RA compare sets TIOA
        TC_CMR_ACPC_CLEAR ); // RC compare clears TIOA
    chTC->TC_CHANNEL[chNo].TC_IER=TC_IER_CPCS; // RC compare interrupt
    chTC->TC_CHANNEL[chNo].TC_IDR=~TC_IER_CPCS;
}
void setFrequencytone(uint32_t frequency)
```

```

{
    if(frequency < 75 || frequency > 3885) { // Capacité d'affichage du compteur LCD = 0 à 380 km/h => 80 à 3880 Hz.
        TC_Stop(chTC, chNo);
        return;
    }
    frequency = frequency << 5; // frequency << 5;
    const uint32_t rc = VARIANT_MCK / frequency;
    const uint32_t ra = rc >> 1; // 50% duty cycle ra = rc >> 2;
    const uint32_t rb = ra >> 2; // 20% duty cycle rb = ra >> 2;
    TC_Stop(chTC, chNo);
    TC_SetRC(chTC, chNo, rc); // set frequency
    TC_SetRA(chTC, chNo, ra);
    TC_SetRB(chTC, chNo, rb);
    TC_Start(chTC, chNo);
}

```

**A+**