

# référence de tension avec Arduino Pro Mini

linéariser et calibrer les entrées analogiques

Giovanni Carrera (Italie)

Si vous utilisez les entrées analogiques de vos microcontrôleurs préférés, vous aurez tôt ou tard besoin de vérifier leur linéarité et leur précision sur toute la plage des valeurs d'entrée. Ce calibrateur de tension précis et compact basé sur Arduino, capable de générer des tensions de référence stables et précises dans la plage de 0 à 5 V, résoudra ce problème.

Vous vous demandez peut-être ce qu'est un calibrateur de tension et à quoi sert-il ? Un calibrateur de tension est un outil qui délivre des tensions connues avec une grande précision et qui est utilisé pour calibrer ou vérifier la classe des voltmètres, des CAN et CNA, des systèmes d'acquisition, etc.

Lors de mesures de tension effectuées avec des microcontrôleurs, il est souvent nécessaire de connaître la constante de conversion ou de vérifier la linéarité du convertisseur analogique-numérique alimenté. Si on se contente d'une précision autour de 5 % il suffit de diviser la tension  $V_{ref}$  par le nombre maximum que l'on obtient en sortie, par exemple pour l'Arduino UNO c'est  $2^{10} - 1 = 1023$ . Si nous utilisons la tension d'alimentation (5 V) comme  $V_{ref}$ , celle-ci peut varier selon que nous alimentons la carte par USB ou par une autre source d'alimentation et est également bruyante. Si nous utilisons la tension  $V_{ref}$  interne (cas préférable), par exemple 1,1 V, nous ne connaissons pas exactement sa valeur. D'autres microcontrôleurs tels que l'ESP32 n'ont même pas d'entrées analogiques fournissant une tension de 0 V, mais de 80 à 150 mV, et la pleine échelle est entre 950 et 1100 mV.

Pour pallier ces inconvénients, il faut procéder à un calibrage, c'est-à-dire utiliser une source de tension stable et peu bruyante ainsi qu'un voltmètre numérique d'une précision supérieure à celle du convertisseur du microcontrôleur. Il faut ensuite effectuer plusieurs mesures de différentes valeurs dans la plage d'entrée du CAN, noter les valeurs sur une feuille de calcul de Excel et procéder à une régression linéaire pour obtenir la pente et l'ordonnée à l'origine. Nous aurons besoin de ces valeurs pour obtenir une sortie en V ou en mV. Ce projet combine



: un générateur de tension très stable et peu bruyant et un voltmètre numérique très précis avec une plage de mesure de 0 à 5 V et une résolution d'environ 0,2 mV.

## La source de tension

Pour avoir une tension de référence stable, c'est-à-dire à faible dérive en température, il ne suffit pas d'utiliser une diode Zener ou un régulateur de tension ordinaire, mais un composant conçu pour fournir une tension de sortie qui reste aussi constante que possible au cours du temps, lorsque la tension d'alimentation, et la température varient. Pour ce projet, nous avons utilisé un LM236H-2.5, dont la dérive en température n'est que de 3,5 mV entre -25 et +85°C lorsqu'il est alimenté par un courant inverse de 1 mA. Si son comportement était linéaire, il y aurait une dérive d'environ 32 µV / °C (environ 12,8 ppm / °C). En maintenant la température et le courant constants, la variation de tension est d'environ 20 ppm sur une période de 1 000 heures.

Si vous souhaitez obtenir des valeurs légèrement plus élevées, vous pouvez utiliser un AD680, et apporter quelques modifications au circuit. Si, au contraire, vous n'avez besoin que de puces aux caractéristiques inférieures, vous pouvez remplacer ce circuit intégré par un TL431, plus répandu, sans avoir à modifier le circuit. La **figure 1** montre le schéma du circuit. Les deux diodes D2 et D3 sont utilisées pour réduire encore la dérive thermique, et le trimmer Rp1 doit être calibré pour avoir une tension de sortie de 2,49 V, la valeur avec laquelle il atteint les meilleures caractéristiques. L'amplificateur opérationnel (nous reviendrons plus loin sur sa fonction) utilisé est un MCP6002,

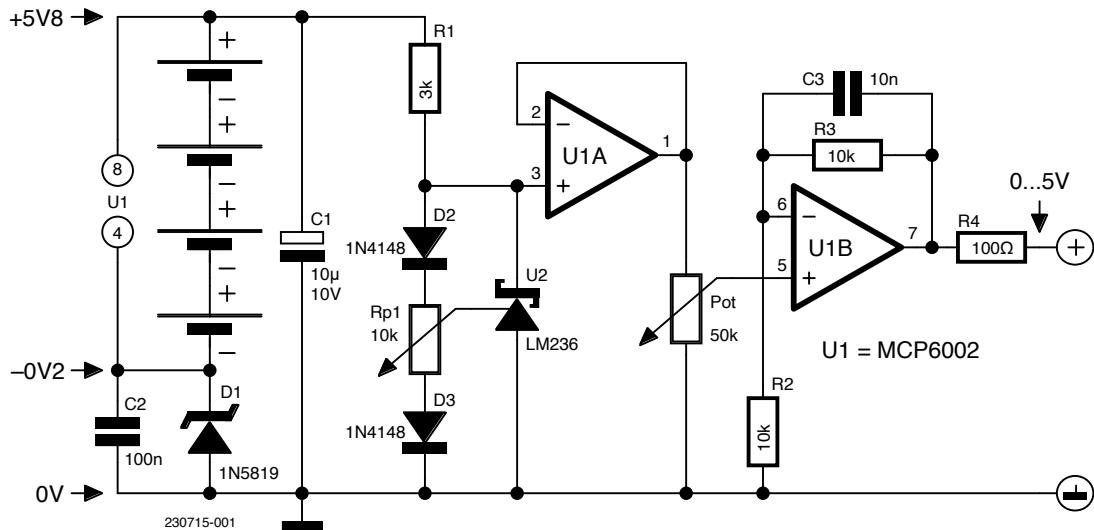


Figure 1. Schéma de la source de tension variable.

un ampli-op double rail à rail, capable de fonctionner à des niveaux de tension très proches de ceux des rails, contrairement aux ampli-op ordinaires qui saturent à quelques volts de différence contre quelques dizaines de mV de rail à rail.

La dérive en température de cette puce est très faible :  $\pm 2 \mu\text{V}/\text{C}^\circ$  et l'offset est de quelques mV. Pour se rapprocher encore plus de zéro, nous avons utilisé la diode schottky D1. Avec ces astuces, la tension négative des ampli-op est inférieure à zéro d'environ 100...200 mV (tension directe de ces diodes), et la tension minimale de sortie n'est influencée que par l'offset des ampli-op. Comme le montre la figure, le potentiel de masse ou GND ne correspond pas au pôle négatif des piles mais à l'anode de la diode D1, ce qui explique la nécessité d'une entrée différentielle du DVM.

Il est possible d'utiliser un LM358 ordinaire à la place du MCP6002. L'ampli-op U1A joue le rôle de séparateur de haute impédance d'entrée, et à sa sortie le potentiomètre qui fait varier le potentiel entre zéro et 2,5 V est connecté. Le potentiomètre utilisé est un 10-tours ; la valeur n'est pas très importante ; elle peut varier entre 5 kΩ et 100 kΩ. Le second ampli-op U1B fonctionne comme un amplificateur non inverseur avec un gain égal à

$$G = (1+R3/R2) = 2$$

La sortie sera donc comprise entre environ 0 V et 5 V. Le condensateur C3 sert de filtre passe-bas pour réduire le bruit thermique des semi-conducteurs. La résistance R4 sert à limiter le courant de sortie en cas de court-circuit accidentel. Une résistance de sortie de 100 Ω ne pose aucun problème car elle est négligeable par rapport aux résistances d'entrée des systèmes à calibrer. L'alimentation est sur batterie pour éviter le bruit du secteur, et il faut éviter à tout prix une alimentation à découpage qui présente un bruit de sortie de quelques centaines de kHz avec des pics allant jusqu'à 100 mV.

### Le voltmètre numérique haute précision

Pour construire un calibrateur, il faut avoir un voltmètre numérique dont la précision est supérieure de plusieurs ordres de grandeurs à celle du système que nous souhaitons calibrer, et dont l'exactitude est suffisante. Le DVM (Digital Volt Meter) proposé est doté d'un convertisseur A/N de 16 bits. Pour les valeurs positives seulement, il y

a  $2^{15} = 32768$  niveaux à partir de zéro, avec un nombre maximum de pleine échelle de 32767. Dans ce système, le CAN a été programmé à une pleine échelle de 6144 mV, la résolution est donc de  $6144 / 32767 = 0,1875$  mV. Le convertisseur n'accepte pas d'entrées avec des tensions supérieures de 0,3 V à la tension d'alimentation Vcc, soit 5,3 V, et la source de tension est conçue pour fournir une valeur maximale d'environ 5 V. Le nombre maximal correspondant à 5 V est donc  $5000 / 0,1875 = 26666$ .

Il s'agit d'une valeur très élevée par rapport à celle d'un DVM normal à 3 1/2 chiffres, qui est de 1999. Avec l'ajout d'un interrupteur, on



### Liste des composants de la source de tension

#### Résistances

R1 = 3 kΩ, 1%, 1/4 W

R2, R3 = 10 kΩ, 1%, 1/4 W

R4 = 100 Ω, 5% 3 W, bobiné

Rp1 = 10 kΩ, trimmer multi-tour

Pot = 50 kΩ, potentiomètre, linéaire, 10 tours

#### Condensateurs

C1 = 10 μF, 25V, tantale

C2 = 100 nF, céramique

C3 = 10 nF, céramique

#### Semi-conducteurs

U1 = MCP6002, ampli-op

U2 = LM236H-2,5, référence de tension 2,49 V

D1 = 1N5819, diode schottky

D2, D3 = 1N4148, diode

#### Divers

SW1 = interrupteur, SP

B1...B4 = 4 x piles alcalines AA, 1,5 V avec porte-piles

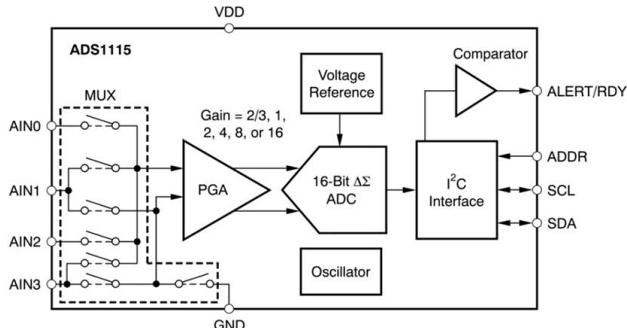


Figure 2. Principaux composants du convertisseur ADS1115.  
(Source : <https://ti.com/lit/ds/symlink/ads1115.pdf>)

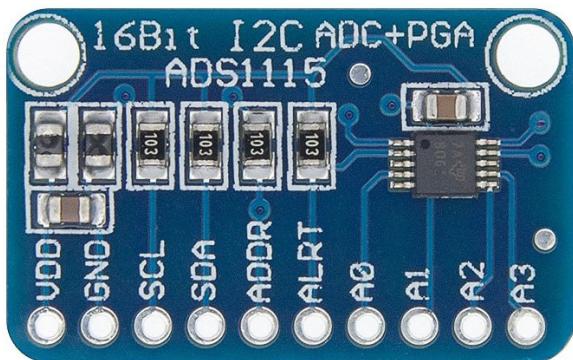


Figure 3. Module ADS1115.

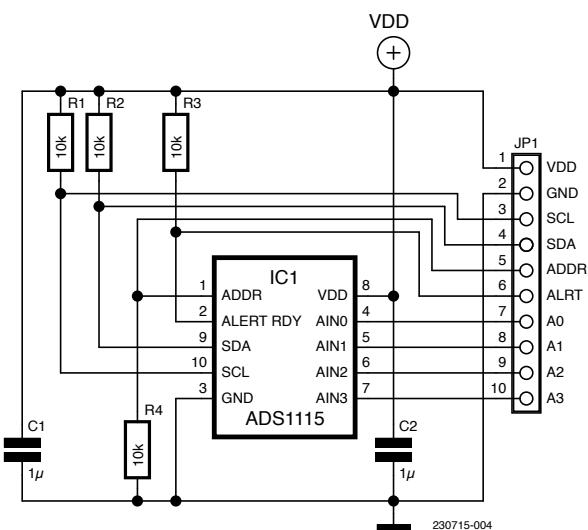


Figure 4. Schéma du module ADS1115.

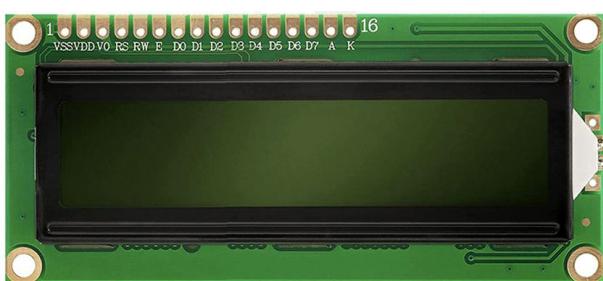


Figure 5. LCD 16×2.

peut débrancher le voltmètre pour l'utiliser séparément. Ce qui ne fonctionne pas, c'est de mettre un diviseur d'entrée à haute impédance pour mesurer des tensions plus élevées, quelque chose qui avait été essayé pour d'autres applications et qui créait des problèmes de bruit qui faisaient que même deux chiffres variaient avec une tension d'entrée stable.

### Le convertisseur ADS1115

Le cœur du DVM est la puce ADS1115, un convertisseur A/N à 16 bits Delta-Sigma avec des caractéristiques de précision et de fiabilité élevées. Il peut avoir jusqu'à quatre entrées. Sur les 16 bits, le bit de poids fort est réservé au signe ; en fait, la puce peut mesurer des valeurs négatives, surtout lorsqu'elle est configurée en mode différentiel, mais avec certaines limitations liées à la tension d'alimentation. Cette puce intéressante, fabriquée par Texas Instruments, présente des caractéristiques remarquables :

- Une résolution de 16 bits, comparée aux 10 bits de l'Arduino.
- Quatre voies simples (référencées à la masse) ou deux voies différentielles.
- Taux d'échantillonnage de 8 à 860 échantillons/seconde.
- Utilisation d'un amplificateur programmable (PGA) permettant de mesurer même de faibles tensions.
- Source de référence interne à faible dérive en température.
- Possibilité d'être alimenté par des tensions de 2 à 5,5 V avec une consommation de seulement 150 µA, encore réduite en mode single-shot.
- Interface I<sup>2</sup>C avec plusieurs adresses sélectionnables.
- Fonction d'alerte (ALERT/RDY) pour surveiller efficacement les tensions. Elle peut réduire la consommation d'énergie, réveiller et activer le microcontrôleur lorsque certains événements se produisent, ou générer des interruptions.

La **figure 2** montre le schéma fonctionnel de la puce ADS1115, qui offre deux modes de conversion :

- **Conversion continue** : l'ADS1115 effectue des conversions en continu. Une fois qu'une conversion est terminée, l'ADS1115 entre le résultat dans le registre de conversion et commence immédiatement une autre conversion.
- **Conversion single-shot** : l'ADS1115 attend que le bit OS soit mis à l'état haut. Une fois affirmé, le bit est mis à 0, indiquant qu'une conversion est en cours. Une fois que les données de conversion sont prêtes, le bit OS est réaffirmé et l'appareil s'éteint. L'écriture d'un 1 sur le bit OS pendant une conversion n'a aucun effet.

Dans ce projet, nous utiliserons le mode single-shot avec une conversion toutes les demi-secondes.

### Module ADS1115

Des modules similaires à celui de la **figure 3**, utilisé dans le projet, sont disponibles sur le marché. Ce module comporte quelques autres composants, comme le montre la **figure 4**. Pour utiliser cette puce avec l'EDI Arduino, nous avons utilisé la bibliothèque *ADS1115\_WE* de Wolfgang Ewald, que vous pouvez télécharger à partir de l'EDI lui-même (Gestionnaire de bibliothèques) ou sur le web [1]. Elle offre

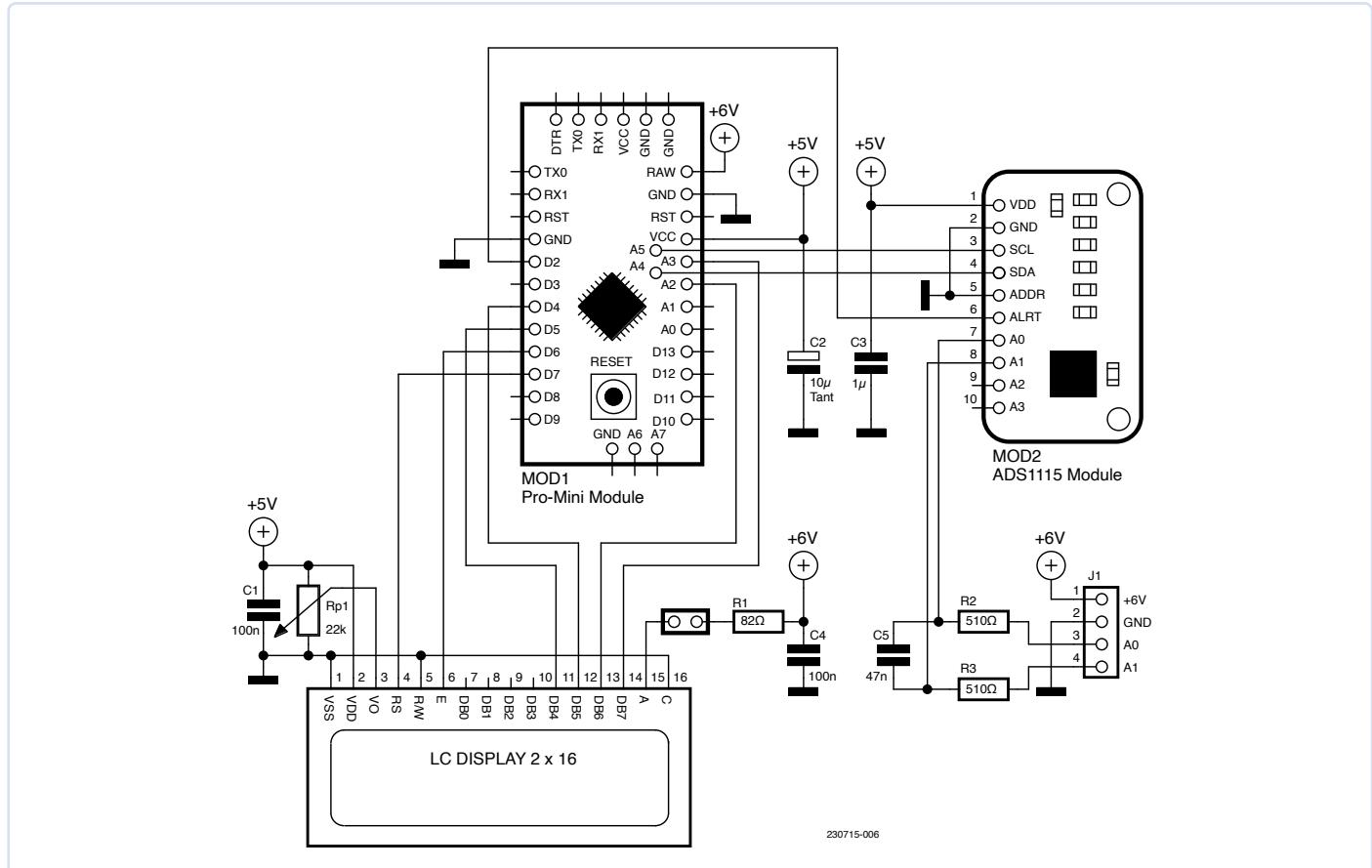


Figure 6. Schéma du DVM.

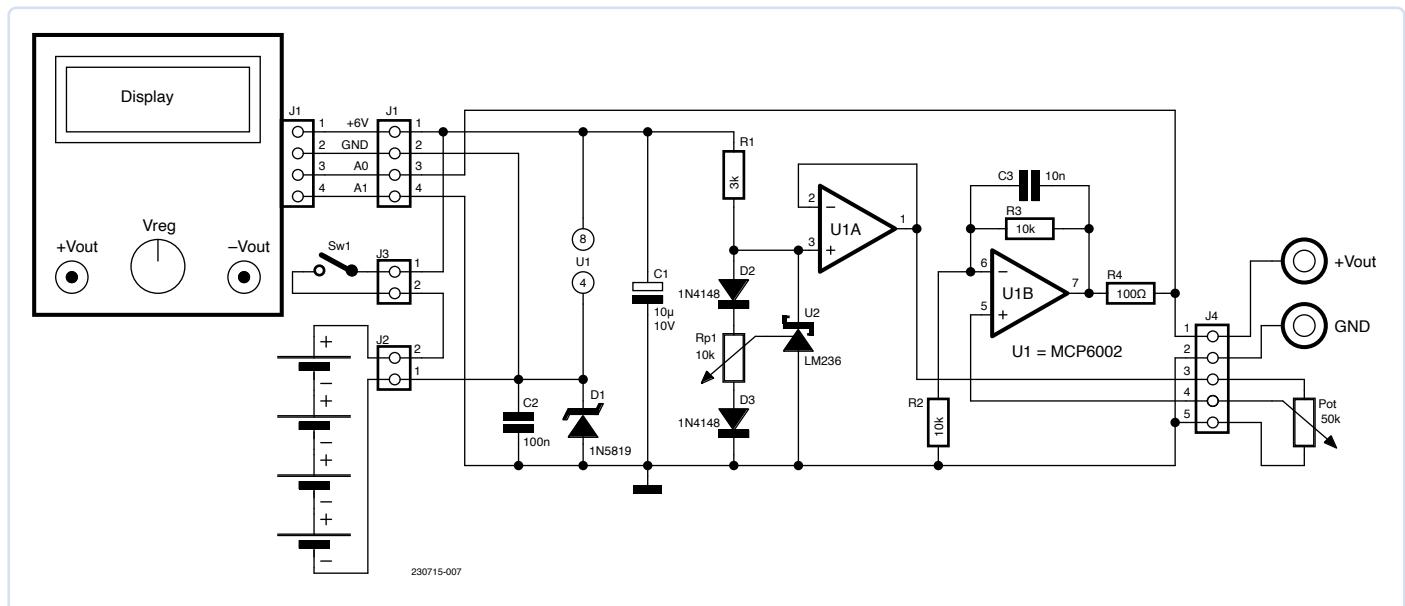


Figure 7. Câblage entre le DVM et la source de tension.

de nombreuses fonctions qui permettent d'exploiter toutes les caractéristiques de la puce.

## LCD

Nous avons choisi un écran LCD compatible avec le contrôleur Hitachi HD44780. En particulier, pour afficher des caractères plus grands, nous avons utilisé l'afficheur classique à deux lignes de 16 caractères similaire à celui de la **figure 5**. Nous avons fait ce choix car ces afficheurs sont très lisibles, consomment peu d'énergie (sans

rétroéclairage) et ont une alimentation et des niveaux logiques de 5 V, ce qui est parfaitement compatible avec Arduino Pro Mini. La bibliothèque Arduino nécessaire est disponible ici [2].

## Schéma de câblage du DVM

La **figure 6** montre le schéma de câblage du DVM. Nous avons utilisé le microcontrôleur Arduino Pro Mini, une carte très compacte, qui fournit également les 5 V pour alimenter l'écran LCD et le convertisseur. Dans ce projet, nous avons utilisé les entrées A0 et A1 du module CAN

configurées en entrée différentielle. La consommation de l'ensemble du système est d'environ 20 à 25 mA lorsque le rétroéclairage de l'écran LCD n'est pas utilisé. La tension idéale à appliquer serait de 6 V ; le système supporte également des tensions plus élevées (12 V max) mais le petit régulateur intégré risque de surchauffer. Le courant de rétroéclairage ne doit pas être fourni par le régulateur interne, mais par le régulateur externe avec la résistance R1 en série qui doit être calculée en fonction du courant requis (voir figure 6).

Nous avons utilisé une résistance de  $82\ \Omega$  dans le prototype, mais il est possible d'utiliser une valeur de  $100\ \Omega$  pour économiser quelques mA. Les anciens afficheurs dont les LED étaient de technologie ancienne nécessitaient un courant allant même jusqu'à 100 mA. Les plus récents nécessitent des courants maximums autour de 20 mA car ils disposent de LED à haute luminosité. Un connecteur à 4 broches relie le DVM à la carte de source de tension variable. Le condensateur C5 et les résistances R2 et R3 de  $510\ \Omega$  ont été utilisés pour réduire le bruit. Ils constituent un filtre passe-bas de premier ordre. La constante de temps est  $\tau = 47.94\ \mu s$ , à laquelle correspond une fréquence de coupure de

$$f = 1/(\tau^2 \cdot \pi) = 3.320\ kHz.$$

## Schéma de câblage

Nous avons réalisé deux cartes pour le prototype, l'une pour le DVM et l'autre pour la source de tension (figure 7), mais il est possible de monter tous les composants sur une même carte. Comme l'alimentation des circuits numériques produit du bruit causé par les courants de commutation, nous avons utilisé des condensateurs de dérivation (tels que le condensateur au tantale C2) pour réduire les pics de bruit.

## Prototype

La **figure 8** montre le prototype. Les deux cartes sont simple-face, l'afficheur et le CAN sont connectés avec des connecteurs strip 16 et 10 broches, avec un pas de 0,1 pouce, et le module Arduino Pro est connecté avec deux connecteurs strip de 12 broches.

La **figure 9** montre la carte du DVM. Les broches A4 et A5 du bus I<sup>2</sup>C de l'Arduino Pro Mini ne sont pas disponibles sur les 12+12 broches et ne sont pas non plus sur une grille de 2,54 mm, donc nous avons soudé deux fils (blanc et marron, illustrés) et un connecteur à 2 broches pour acheminer les signaux vers le module ADS1115. Le connecteur en bande à 16 broches que l'on voit en haut à gauche est destiné à l'afficheur LCD, qui est connecté à 90 degrés. La carte est très compacte et est

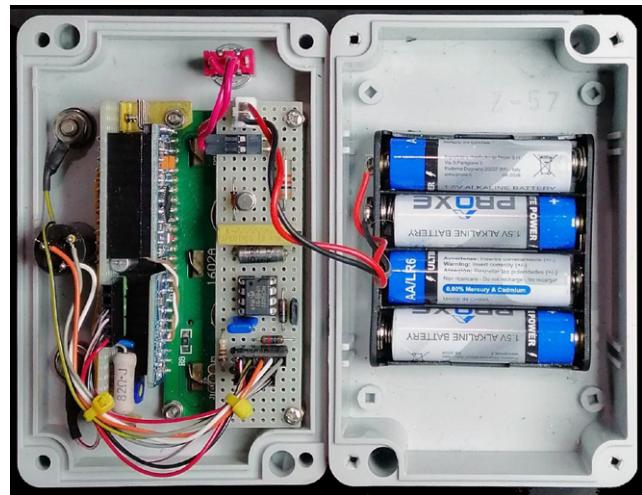


Figure 8. Vue intérieure du prototype terminé.

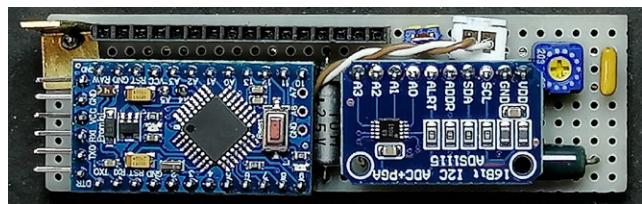


Figure 9. La carte DVM finale.

également fixée à l'écran avec un petit support en laiton, visible en haut à gauche de l'image.

## Programmation

Pour la programmation, nous avons besoin d'un adaptateur série USB TTL similaire à celui de la **figure 10**. Bien sûr, nous devons également charger le pilote de l'adaptateur qui, en plus des signaux Rx et Tx, doit avoir le DTR qui est utilisé pour réinitialiser le système. Il sert également d'interface série USB, mais dans ce cas, son rôle principal est de téléverser le croquis compilé de l'EDI Arduino à notre système.

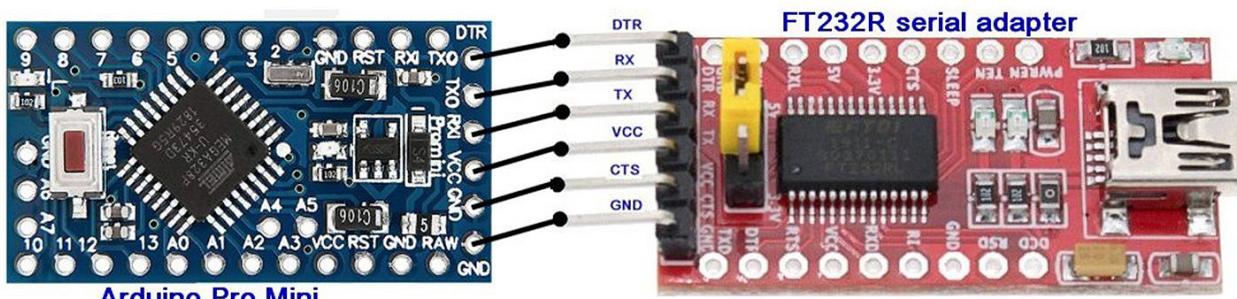


Figure 10. Connexions entre l'adaptateur série et l'Arduino Pro Mini.

Dans l'EDI Arduino, il faut configurer la carte comme Arduino Pro à 16 MHz et 5 V. Une fois programmée, si tout va bien, vous pouvez retirer le programmeur et alimenter la carte par les accus.

Nous avons utilisé un connecteur femelle à 6 broches avec un pas de 0,1 pouce, soudé directement au connecteur de l'adaptateur, qui est mâle, mais vous pouvez aussi utiliser des fils femelle-femelle compatibles avec une plaque d'essai. Les adaptateurs série possèdent des dispositions de broches et des puces différentes. Avec certains adaptateurs série, comme le Prolific PL2303, des problèmes de pilotes ont été rencontrés avec Windows 10.

## Calibrage du DVM

Bien que les mesures de l'ADS1115 soient déjà très précises, le calibrage a été réalisé avec un multimètre de table HP 3478A de 5 1/2 digits avec une résolution de 0,1 µV. Pour cela, il faut remplacer la ligne, dans la fonction `printData()` :

```
// calibrated and result in millivolts
voltage = Ch0*1.000515+0.022;
```

avec celle-ci, c'est-à-dire supprimer la correction :

```
// not calibrated and result in millivolts
voltage = Ch0;
```

Environ dix valeurs ont été générées, les lectures du multimètre de référence et les lectures du calibrateur ont été insérées dans Excel, le *nuage de points* avec la *ligne de tendance* a été saisi avec les *options linéaires* et en cochant *Afficher l'équation* sur le graphique et *Afficher le R-carré* sur le graphique. Le résultat était excellent, comme le montre la **figure 11**. La valeur de R2 était de 1 avec pas moins de



## Liste des composants du DVM

### Résistances

R1 = 82 Ω, ±5%, 1 W , voir texte

R2, R3 = 510 Ω, ±1%, ¼ W

Rp1 = 22 kΩ, trimmer

### Condensateurs

C1, C4 = 100 nF, 50 V, céramique

C2 = 10 µF, 25 V, tantale

C3 = 1 µF, 63 V, céramique

C5 = 47 nF, 63 V, polyester

### Modules

1 x Arduino Pro Mini (ou compatible)

1 x Afficheur LCD 16x2, avec rétro-éclairage

1 x ADS1115 Convertisseur A/N

### Divers

12+12+10+16 Pin Strip, female

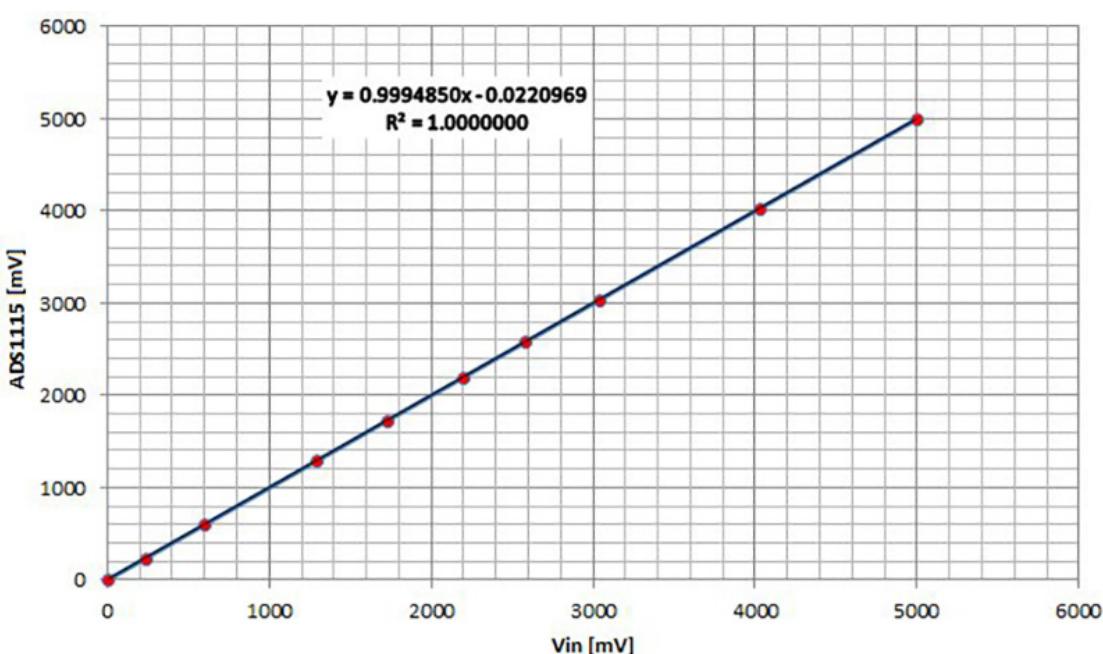


Figure 11. Graphique de calibrage (linéarisation) du DVM.

sept décimales égales à zéro : cela correspond à une valeur de linéarité extrêmement élevée. Pour corriger les données, il faut utiliser la formule inverse, c'est-à-dire inverser les axes. Nous obtenons donc :

➤ slope = 1.000515286  
➤ intercept = 0.022110099  
➤ error = 0.051941236

Comme vous pouvez le constater, l'erreur est minime. Si nous ne disposons pas d'un instrument très précis, nous pouvons nous passer de la correction, en nous contentant d'une erreur standard de 0,051941236 ; sinon, nous corrigons la ligne avec les valeurs que nous avons obtenues lors de notre calibrage. Dans le prototype, la tension minimale générée était de 1,7 mV, la maximale de 4 997,03. Le minimum n'est pas égal à zéro à cause de l'offset des ampli-op.

## Description du programme

Le programme est relativement simple ; après avoir initialisé les entrées/sorties, l'afficheur et le CAN, il effectue une mesure toutes les 500 ms. La fonction `readChannel(ADS1115_MUX channel)` est celle proposée pour le mode single-shot et effectue la mesure en millivolts. La fonction `LCDprintLine(String text, byte line)` imprime une chaîne sur la ligne 0 ou 1. La fonction `printData()` corrige la mesure avec les résultats du calibrage initial et l'imprime. ↗

230715-04

## Questions ou commentaires ?

Contactez Elektor ([redaction@elektor.fr](mailto:redaction@elektor.fr)).



### Listage 1.

```
/* program ArduCalibrator.ino for the ADS1115, 16 bit 4 ch ADC
a single differential input (A0,A1), Ch0 6144 mV (5000 generated)
read the precision voltage source
uses an Arduino Pro Mini, LCD 16x2 display
Giovanni Carrera 23/11/2022 with calibration
*/
#include <Wire.h>
#include<ADS1115_WE.h>
#include <LiquidCrystal.h>
ADS1115_WE adc = ADS1115_WE(); // uses Wire / ADC Address = 0x48
// LCD pins
#define rs 7
#define en 6
#define d4 5
#define d5 4
#define d6 A2
#define d7 A3
// initialize the library by associating any needed LCD interface pin
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
#define Alert 2 // AD Alert pin
const unsigned long deltat_ms = 500;
unsigned long cms, pms;
float Ch0,Ch1;
void setup(){
    Wire.begin();
    //Serial.begin(115200);
    pinMode(Alert,INPUT);
    // set up the LCD's number of columns and rows:
    lcd.begin(16, 2);
    lcd.print(F("ArduCalibrator"));
    lcd.setCursor(0, 1); // print on the second row
    lcd.print(F("GCar V231122"));
    delay(2000);
    if(!adc.init()){

```

```

lcd.setCursor(0, 1);
lcd.print(F("No ADS1115 found"));
while( true );// ends here
}
adc.setVoltageRange_mV(ADS1115_RANGE_6144); // 6144 mV input range
LCDprintLine("PRECISION SOURCE", 0);
}

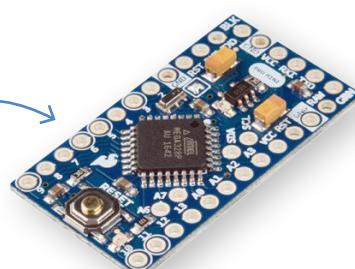
void loop() {
    cms = millis();
    // checks if passed deltat milliseconds
    if(cms - pms > deltat_ms) { // period timebase
        pms = cms;// update pms
        Ch0 = readChannel(ADS1115_COMP_0_1);// Ch0 differential A0(+) with A1(-)
        Ch0 = readChannel(ADS1115_COMP_0_1);// repeating helps
        printData();
    }
}

```



### À propos de l'auteur

Giovanni Carrera est titulaire d'un diplôme en ingénierie électronique. En tant que professeur d'université à la faculté d'ingénierie navale de Gênes, en Italie, il a enseigné de nombreux cours, tels que l'automatisation navale et la simulation des systèmes de propulsion des navires. Carrera a commencé à travailler à la fin des années 1970 avec le CPU 6502, avant de passer à d'autres processeurs. Aujourd'hui, il s'adonne à la conception et au développement de circuits électroniques analogiques et numériques, dont il a présenté un grand nombre sur ses blogs (ArduPicLab et GnssRtkLab) et dans divers magazines.



### Produits

- **SparkFun Arduino Pro Mini 328 (5 V, 16 MHz)**  
[www.elektor.fr/20091](http://www.elektor.fr/20091)
- **Écran LCD rétroéclairé standard de 2 x 16 caractères**  
[www.elektor.fr/16414](http://www.elektor.fr/16414)



### LIENS

- [1] Wolfgang Ewald's ADS1115 library: [https://github.com/wollewald/ADS1115\\_WE](https://github.com/wollewald/ADS1115_WE)
- [2] Bibliothèque LiquidCrystal : <https://github.com/arduino-libraries/LiquidCrystal>
- [3] Téléchargements pour ce projet: <https://elektormagazine.fr/Arducalibrator>
- [4] G. Carrera, « Calibrateur de CAN à faible bruit pour les microcontrôleurs modernes », circuits de vacances 2022 : <https://www.elektormagazine.fr/magazine/elektor-264/60886>
- [5] Ultra-Small, Low-Power, 16-Bit Analog-to-Digital Converter with Internal Reference: <https://ti.com/lit/ds/symlink/ads1114.pdf>