

29 Ampèremètre Arduino

Giovanni Carrera (Italie)

La mesure d'un courant nécessite l'utilisation d'un ampèremètre placé en série avec la charge. Un ampèremètre idéal ne présente aucune chute de tension, c'est un court-circuit. En réalité, un bon ampèremètre, lorsqu'il est inséré, doit avoir une très faible chute de tension pour ne pas altérer le circuit à mesurer. Le capteur présenté ici a une chute de tension de seulement 50 mV. Il a une bonne précision et utilise des composants faciles à obtenir.

La technique de mesure utilisée dans ce projet est de type « low-side » (c'est-à-dire que la résistance de shuntage est connectée entre la charge et la masse). Le shunt est connecté à un amplificateur différentiel avec une sortie compatible avec Arduino (voir la **figure 1**). Ce projet utilise la référence interne de l'ATmega328P ; la tension pleine échelle du convertisseur est donc de 1,1 V. Grâce à cette solution, on obtient une référence plus stable et moins bruyante avec une résolution de $1100/1024 = 1,074$ mV.

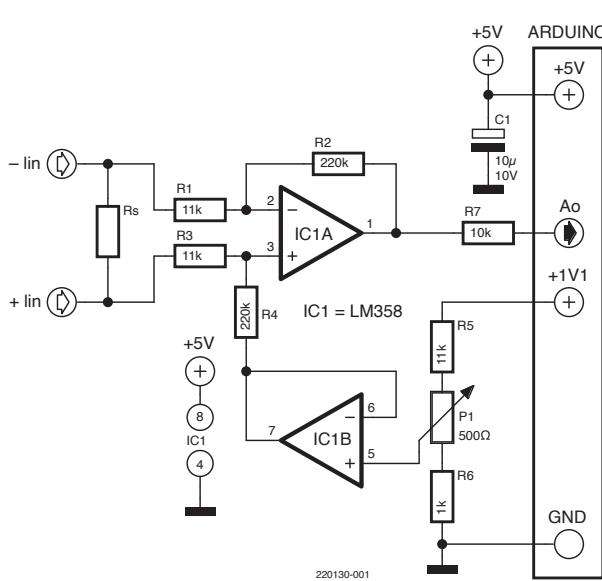
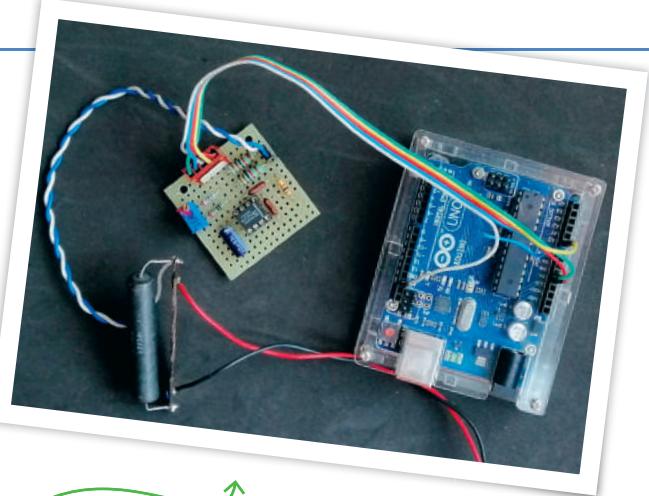


Figure 1. Schéma de l'ampèremètre Arduino.



Lorsque nous utilisons des amplificateurs opérationnels de puissance avec une seule alimentation, des problèmes se posent si les signaux s'approchent de zéro. Même en utilisant des amplificateurs opérationnels spéciaux à excursion totale (ou « rail to rail ») dont l'oscillation de sortie est plus proche des valeurs maximales de l'alimentation, ils ont rarement des tensions de sortie inférieures à 20 mV pour une entrée à zéro. Il est possible d'obtenir une sortie similaire en utilisant le fameux LM358, qui peut fonctionner même avec une seule alimentation. Un procédé consiste à travailler à une valeur supérieure à zéro, puis à soustraire le décalage (« offset ») par logiciel. Pour avoir un zéro en sortie et aussi pour éliminer l'offset, une constante V_{sh} d'environ 100 mV est ajoutée. Nous utilisons pour ce circuit un amplificateur différentiel IC1A, dont la sortie est égale à :

$$V_o = (V_{IN+} - V_{IN-}) \cdot \left(\frac{R_2}{R_1} \right) + V_{sh}$$

Le deuxième amplificateur opérationnel IC1B, connecté en tant que suiveur de tension, génère cette tension V_{sh} en utilisant un diviseur pour la tension V_{ref} de l'Arduino

Pour assurer un fonctionnement correct, les relations suivantes doivent être respectées : $R1 = R3$ et $R2 = R4$ avec de très faibles tolérances. Je les ai sélectionnées parmi des résistances à film métallique avec une tolérance de 1%, désormais facilement disponibles, en utilisant un bon ohmmètre. Comme l'amplificateur double LM358 n'a pas un offset très faible, j'ai préféré ne pas trop augmenter le gain, donc $R2/R1 = 20$.

Conversion de tension

Le convertisseur A/N a une pleine échelle de 1100 mV. Je déduis 100 mV pour obtenir le zéro, ce qui me permet d'avoir 1000 mV pour la mesure. Cela signifie que la tension d'entrée maximale doit être de $1000/20 = 50$ mV. Le **tableau 1** montre quelques valeurs de shunt et les taux de courant correspondants.

Pour les courants élevés, il est préférable d'utiliser des shunts commerciaux. Beaucoup d'entre eux ont une chute de tension de 75 mV, nous pouvons donc faire varier le gain à $1000/75 = 13,33$.

Nous obtenons des améliorations significatives en remplaçant le LM358 par une entrée à faible offset tels que les AOP OP290, OPA2196, OPA2277 et autres puces similaires. Nous pouvons supprimer le décalage en utilisant un trimmer sur le diviseur qui fournit V_{sh} . Le logiciel fera le reste en soustrayant une valeur constante. Pour $V_{sh} = 100$ mV, ceci est égal à :

$$Nsh = 100 * 1023 / 1100 = 93$$

Table 1. Valeurs des shunts et intensités du courant.

I max [A]	Shunt [Ω]
0,5	0,1
1	0,05
5	0,01
10	0,005

 Télécharger le projet



www.elektormagazine.fr/summer-circuits-22

C'est la constante à soustraire. Le trimmer P1 doit être ajusté jusqu'à ce que l'on obtienne zéro pour un courant nul. Au cours de la phase de test, j'ai augmenté cette valeur à 98 pour mieux travailler avec le trimmer.

Programme

Le programme d'exemple est très simple. Les valeurs utilisées sont relatives à mon système (voir **Listage 1**, le code source peut être téléchargé [1]). Pour calibrer le trimmer P1, il suffit d'exécuter le programme et de tourner le trimmer pour lire zéro sur le moniteur série ou l'IDE Arduino avec un courant d'entrée nul. S'il n'est pas possible de mettre le trimmer à zéro, il est nécessaire de modifier légèrement la constante `Nsh` à soustraire de la mesure. La constante `mVtomA` est dérivée des mesures expérimentales, en utilisant un bon ampèremètre de précision, une alimentation et quelques charges résistives comme des ampoules de voiture ou des résistances filaires de puissance adéquate. 

220130-04

Listage 1. Code source [1].

```
/* program ArduAmmeter.ino Arduino current meter
Giovanni Carrera, rev. 11/07/2019 */

float NtomV;
const float VREF = 1095;      // in mV, this value can be read on VREF pin
const int Nsh = 98;           // shift value corresponding to about 100 mV
const float mVtomA = 2.17;    // value obtained by a calibration

void setup() {
  Serial.begin(9600);
  analogReference(INTERNAL); // internal ADC reference input = 1100 mV
  NtomV = VREF/1023;        // constant of conversion into millivolts
}

void loop() {
  int val = analogRead(A0)-Nsh; // read the current sensor
                               // and remove the shift
  float mvolt = NtomV*val;   // convert to millivolt
  float mamp = mvolt/mVtomA; // convert to milliampere
  Serial.print("Vo = ");
  Serial.print(mvolt,1);
  Serial.print(" mV - Current = ");
  Serial.print(mamp,0);
  Serial.println(" mA");
  delay(1000);
}
```

À propos de l'auteur



Giovanni Carrera est titulaire d'un diplôme d'ingénieur en électronique. Professeur d'université à la faculté d'ingénierie navale de Gênes, en Italie, il a notamment enseigné l'automatisation navale et la simulation des systèmes de propulsion des navires. Il a commencé à travailler à la fin des années 1970 sur le processeur 6502, avant de se pencher sur d'autres unités centrales. Aujourd'hui, il aime concevoir et développer des circuits électroniques analogiques et numériques, dont beaucoup ont fait l'objet d'articles sur ses blogs (ArduPicLab et GnssRtkLab) et dans divers magazines.

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (g.carrera@elettronicaemake.it), ou contactez Elektor (redaction@elektor.fr).



Liste des composants

Résistances

R1,R3 = 11 k, $\pm 1\%$ à film métallique
R2,R4 = 220 k, $\pm 1\%$ à film métallique
R5 = 11 k, $\pm 2\%$ à film métallique
R6 = 1 k, $\pm 2\%$ à film métallique
R7 = 10 k, $\pm 5\%$
P1 = trimmer multi-tour 500 Ω
Rs = shunt 0,1 Ω

Condensateurs

C1 = 10 μ F, 25 V Aluminium
électrolytique

Semiconducteurs

IC1 = AOP double LM358

Divers

Arduino Uno ou Nano

LIENS

[1] Code source: <https://www.elektormagazine.com/summer-circuits-2022>