



# Capteurs - Intensité du courant

[Mise à jour le 24/6/2024]

## 1. Généralités

Un courant électrique est un mouvement d'ensemble de porteurs de charges électriques, généralement des électrons, au sein d'un matériau conducteur. Ces déplacements sont imposés par l'action de la force électromagnétique, dont l'interaction avec la matière est le fondement de l'électricité. [Wikipédia](#)



## 2. Capteurs analogiques



### 2.1 ±5A - (ACS714)

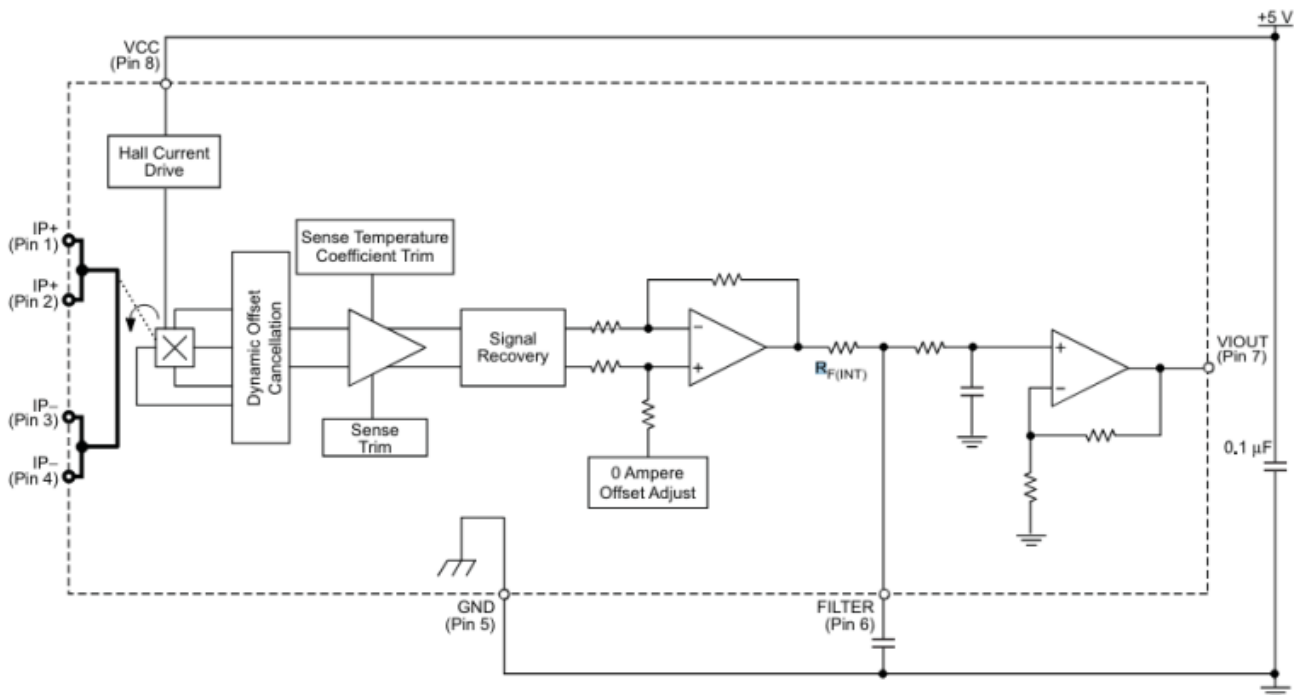
#### 2.1.1 Présentation

Le module MR362 est un capteur d'intensité basé sur un composant ACS714. Il mesure un courant compris entre -5A et +5A. La sortie analogique est proportionnelle au courant mesuré (2,5V pour 0A).

- **Distributeur** : [Gotronic](#)
- **Caractéristiques**
  - Alimentation: 5 Vcc
  - Consommation: 10 mA
  - Plage de mesure: -5 A à +5 A en AC ou DC
  - Sensibilité: **185 mV/A**(à 25°C) (centrée sur 2,5V)

- précision: **+/-1,5%**
- Shunt: 1,2 mΩ
- Connecteur: JST au pas de 2 mm
- T° de service: -40 à +85 °C
- Dimensions: 28 x 21 x 13 mm

• **Schéma-blocs**




• **Documentation**

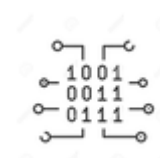
- Fichier Acrobat Reader à télécharger [ici](#)

• **Modèle**

- Sensibilité: 185 mV/A (centrée sur 2,5V)

• **Aide pour la simulation de la chaîne de mesure**

- Les équations de la chaîne de mesure sont téléchargeables [ici](#) 
- Le modèle à simuler est téléchargeable [ici](#)



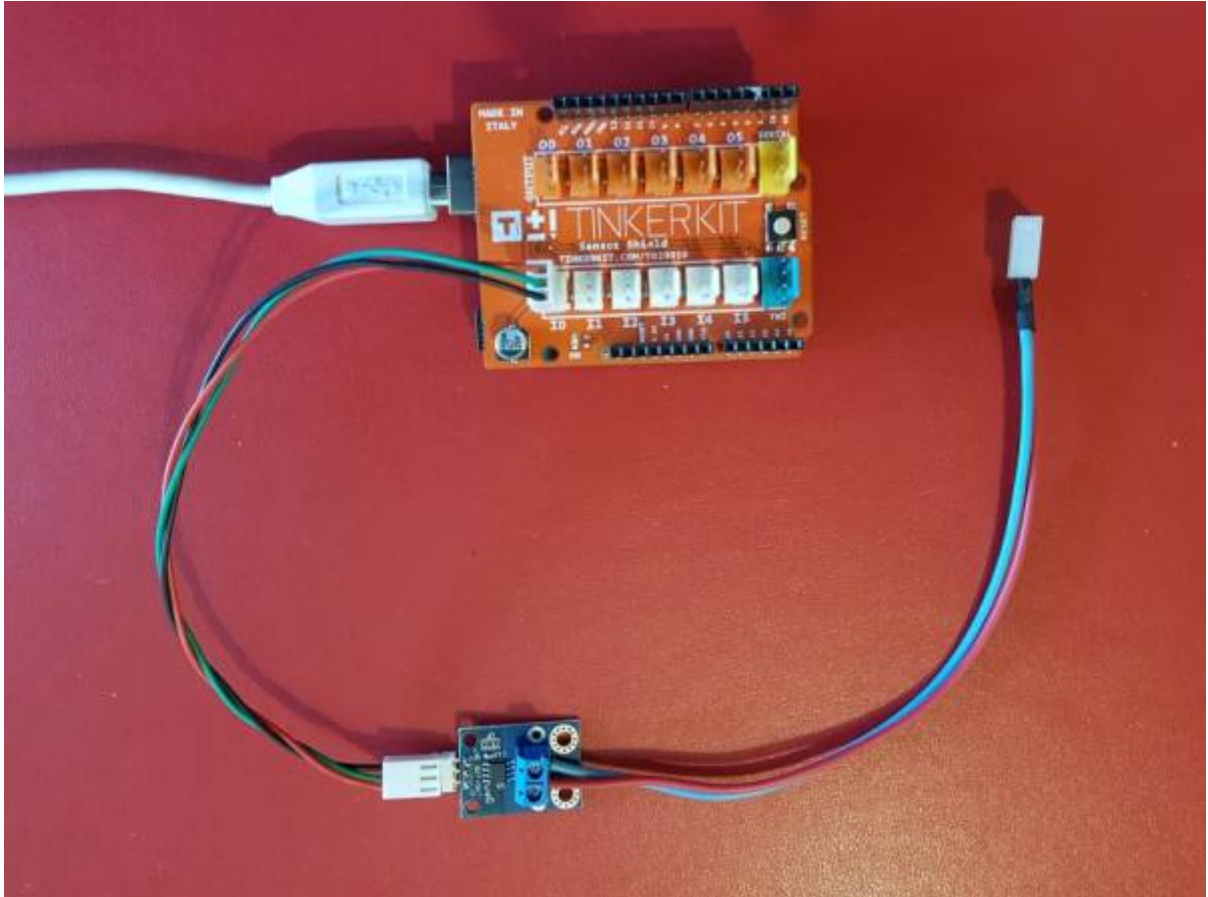
## 2.1.2 Exemples de code

- [Arduino UNO](#)
- [Rpi Pico \(µPython\)](#)
- [Netduino \(C#\)](#)

- **Fiche guide** à télécharger



- Connexion à un shield [Tinkerkit v2](#) monté sur une Arduino Uno



- Exemple pour tester le capteur (étude dans la fiche guide)



### acs714\_V1.cpp

```
/*
ACS714 sur Arduino Uno
*/

#define RefVal 5.0
#define Pin A0

// Take the average of 500 times
const int averageValue = 500;
long int sensorValue = 0;
float sensitivity = 1000.0 / 185.0; //1000mA per 185mV
```

```
// Output voltage with no current: 2500mV
float Vref = 2500;

void setup() {
  Serial.begin(9600);
}

void loop() {
  // Read the value 500 times:
  for (int i = 0; i < averageValue; i++) {
    sensorValue += analogRead(Pin);

    // wait 2 milliseconds before the next loop
    delay(2);
  }
  sensorValue = sensorValue / averageValue;

  // The on-board ADC is 10-bits
  // Different power supply will lead to different reference sources
  // example: 2^10 = 1024 -> 5V / 1024 ~= 4.88mV
  //          unitValue= 5.0 / 1024.0*1000 ;
  float unitValue = RefVal / 1024.0 * 1000;
  float voltage = unitValue * sensorValue;

  // When no load, Vref=initialValue
  Serial.print("initialValue: ");
  Serial.print(voltage);
  Serial.println("mV");

  // Calculate the corresponding current
  float current = (voltage - Vref) * sensitivity;

  // Print display current (mA)
  Serial.print(current);
  Serial.println("mA");

  Serial.print("\n");

  // Reset the sensorValue for the next reading
  sensorValue = 0;
  // Read it once per second
  delay(1000);
}
```

- Exemple à compléter (étude dans la fiche guide)

```

/*#####
#####
Connexions de l'ACS714 à une carte Arduino Uno
* Vcc    -> 5V      # Alimentation du module
* GND    -> GND
* OUT    -> PIN A0  # Signal de sortie du capteur (tension)
#####
#####*/

// Constantes et variables
const int Ucapteur = A0; // Sortie du capteur : tension (image de
l'intensité mesurée)
int N = 0;                // Valeur délivrée par le CA/N
float Intensite = 0;      // Valeur de l'intensité mesurée par l'ACS714

void setup() {
  Serial.begin(9600);
}

void loop() {
  N = analogRead(Ucapteur);
  Intensite = ((float)(N - a) * b) / c; // Calculer a,b et c pour un
ACS714 à partir de l'étude de la chaîne d'information

  Serial.print("Intensité = " ); // Affichage dans la console
  Serial.print(Intensite);
  Serial.println("A");

  delay(2000);
}

```

A venir



### Archives

Le projet contenant l'exemple pour la carte **Netduino**, à programmer avec l'IDE **Visual Studio Community 2015**, est téléchargeable [ici](#)

## 2.2 ±5A - (ACS70331)

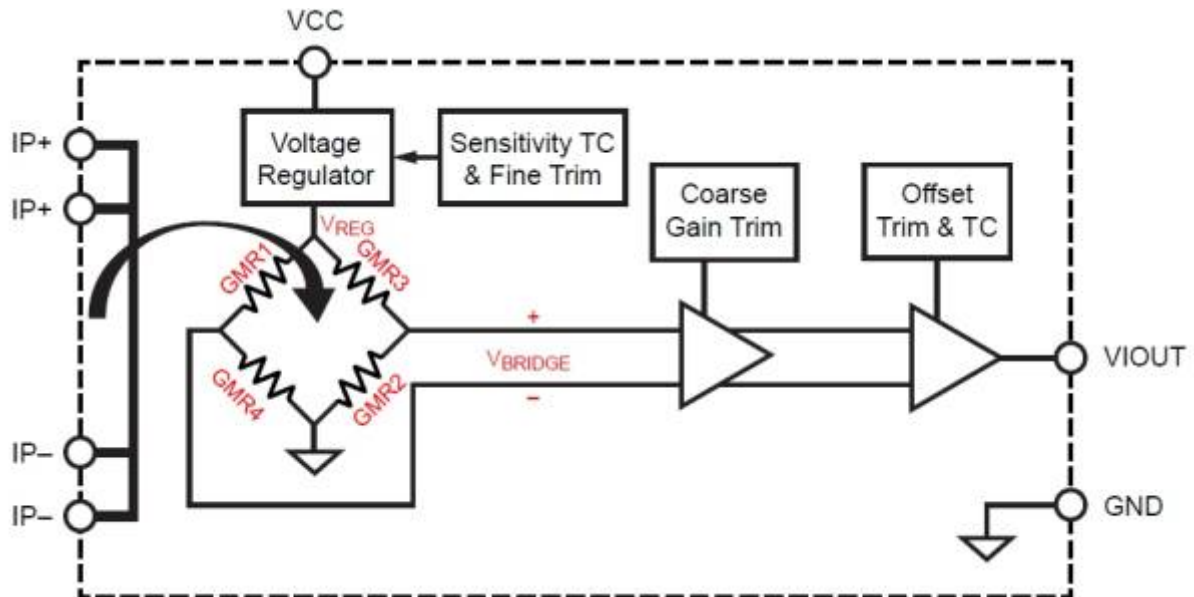


### 2.2.1 Présentation

- **Source** : [wiki](#) Seeed studio

Ce module Grove, basé sur un capteur ACS70331, permet la mesure de l'intensité d'un **courant AC et DC** jusqu'à 5 A. Ce capteur s'insère en série dans un montage électrique et délivre une tension analogique exploitable par une carte Arduino ou compatible.

- **Distributeur** : [Gotronic](#)
- **Caractéristiques**:
  - Alimentation: 3,3 ou 5 Vcc
  - Interface: compatible Grove
  - Sortie: analogique
  - Sensibilité: 200 mV/A
  - Tension d'entrée: 100 Vcc maxi
  - Plage de mesure: 0 - 5 A alternatif ou continu
  - Dimensions: 20 x 40 mm
  - Température de travail: -40 °C à +85 °C
- **Schéma-blocs**



- **Documentation**
  - Datasheet à télécharger [ici](#)
- **Modèle**
  - Sensibilité: **200 mV/A**
- **Aide** pour la simulation de la chaîne de mesure
  - Des données de mesure sont téléchargeables [ici](#)
- **Schéma**
  - A télécharger [ici](#)

### 2.2.2 Bibliothèques

- [Arduino UNO](#)
- [Rpi Pico \(µPython\)](#)
- Téléchargez la bibliothèque de [capteurs de courant Grove](#) depuis Github et l'installez dans le dossier ./Documents/Arduino/libraries.

A venir



### 2.2.3 Exemples de code

- [Arduino UNO](#)
- [Rpi Pico \(µPython\)](#)
- **Ressource à consulter** : [wiki](#) Seeed studio.
- **Exemple de l'IDE Arduino pour tester le capteur** et corriger le décalage à zéro  
Dans l'IDE Arduino, sélectionner : Fichiers → Exemples → Grove current sensor → **Grove\_5A\_DC\_Current\_Sensor**
- **Résultat** dans la console

```
COM36  
  
initialValue: 595.70mV  
-4561.48mA
```

A venir



## 2.3 ±12,5A - (ACS711)

### 2.3.1 Présentation

Le module MR392 est un capteur d'intensité basé sur un composant ACS711. Il mesure un courant compris entre -12,5A et +12,5A. La sortie analogique est proportionnelle au courant mesuré (1,65 V pour 0 A sous 3,3 V).

- **Distributeur** : [Gotronic](#)
- **Caractéristiques**
  - Alimentation: 3,3 à 5 Vcc
  - Consommation: 5,5 mA maxi
  - Plage de mesure: -12,5A à +12,5A en AC ou DC
  - Sensibilité: 110 mV/A



- Shunt: 1,2 mΩ
- T° de service: -40 à +85 °C
- Dimensions: 28 x 21 x 13 mm



- **Documentation**

- Datasheet à télécharger [ici](#)

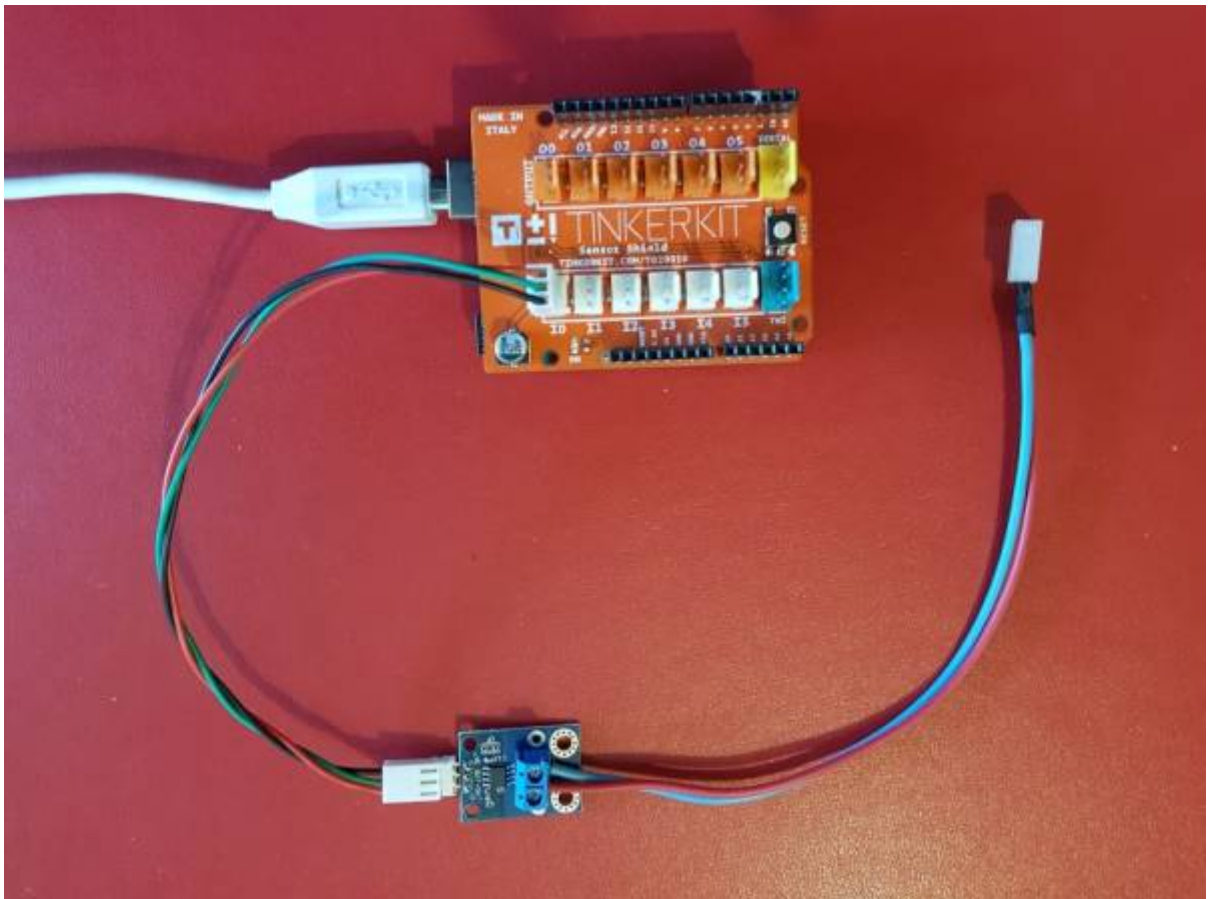
- **Modèle**

- Sensibilité: 110 mV/A



### 2.3.2 Exemples de code (A compléter)

- [Arduino UNO](#)
- [Rpi Pico \(μPython\)](#)
- *Connexion à un shield [Tinkerkit v2](#) monté sur une Arduino Uno.*



- Exemple pour tester le capteur



```
/*#####  
###  
Author:  
* Mirko Prosseda (06-2013)  
* email: mirko.prosseda@gmail.com  
  
Description:  
* 5A Linear Current Sensor test sketch v1.0  
* Read current value from the sensor and print its value on the Serial  
Monitor  
  
Connections:  
* BOARD -> ARDUINO  
* Vcc -> 5V  
* GND -> GND  
* OUT -> PIN A0  
#####  
##*/  
  
// Define constants and variables  
const int analogInPin = A0; // Analog input pin that the current sensor is  
attached to  
int sensorValue = 0; // value read from the sensor  
float outputValue = 0; // converted value of the sensor reading  
  
// Initialization  
void setup() {  
  Serial.begin(9600); // Serial Port initialization  
}  
  
// main loop  
void loop() {  
  sensorValue = analogRead(analogInPin); // reads  
the sensor value and convert it  
  outputValue = ((float)(sensorValue - 338) * 5 / 0.11) / 1024; // A  
vérifier  
  
  Serial.print("Current Sensor value= "); // print  
results  
  Serial.print(outputValue);  
  Serial.println("A");  
  
  delay(200);  
}
```

A venir



## 2.4 ±30A - (ACS712)

### 2.4.1 Présentation

- **Source** : [wiki](#) Phidgets

Ce module Phidgets doit être inséré en série dans le circuit à mesurer. Il est équipé de 2 sorties qui permettent de mesurer séparément le courant continu et alternatif pour les ondes de forme complexe..

- **Distributeur** : [Gotronic](#)
- **Caractéristiques**
  - Alimentation: 4,5 à 5,5 Vcc
  - Consommation maxi: 10 mA
  - Impédance de sortie: 1 K
  - Plage de mesure:
    1. 0 à 30 A en alternatif (10 kHz maxi)
    2. -30 à +30 A en continu
  - Précision typique: ±1,5 % à 25°C
  - Dimensions: 46 x 36 x 25 mm.
- **Documentation**
  - Site Phidgets : [Guide De L'utilisateur 1122](#)

## 3. Capteurs numériques

### 3.1 INA219 Current Sensor



### 3.1.1 Présentation

- **Source** : [wiki](#) Adafruit.

Capteur de courant basé sur le circuit INA219B permettant de mesurer un courant continu jusqu'à 3,2 A avec une résolution de 0,8 mA (jusqu'à 26 Vcc). Il communique avec un microcontrôleur Arduino ou compatible via le bus I2C et est équipé d'un shunt de 0,1  $\Omega$  (chute de tension de 320 mV pour 3,2 A).

- **Distributeur** : [Gotronic](#)
- **Caractéristiques**
  - Alimentation: 5 Vcc
  - Plage de mesure: jusqu'à 3,2 Acc
  - Résolution: 0,8 mA
  - Précision:  $\pm 1\%$
  - Shunt: 0,1  $\Omega$  1% 2 W
  - Bus série I2C : **SLA = 0x40** (default) configurable
  - Dimensions: 26 x 22 x 13 mm



- **Documentation**
  - Datasheet à télécharger [ici](#)

### 3.1.2 Bibliothèques

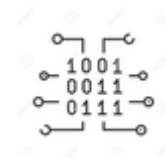
- [Arduino UNO](#)
- [Rpi Pico \( \$\mu\$ Python\)](#)
- *A partir du gestionnaire de bibliothèques de l'IDE Arduino, installer :*

**Adafruit INA219** par Adafruit

1.2.3 installed

INA219 Current Sensor INA219 Current Sensor

A venir



### 3.1.3 Exemples de code

- [Arduino UNO](#)
- [Rpi Pico \(µPython\)](#)
- **Ressource** : [wiki](#) Adafruit.
- **Exemple** de l'IDE Arduino pour tester le capteur  
Dans l'IDE Arduino, sélectionner : Fichiers → Exemples → Adafruit INA219 → **getcurrent**

A venir

From:  
<https://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link:  
<https://webge.fr/dokuwiki/doku.php?id=materiels:capteurs:intensite:intensite&rev=1719310109>

Last update: **2024/06/25 12:08**

