

# Utiliser un capteur de courant SCT013-30A

Pour nos installations de domotique, il peut être important de mesurer la quantité de courant qui circule dans un conducteur électrique. Le réseau utilise du courant alternatif avec une tension de 230 Volts. Le capteur SCT013, positionné au tour du conducteur électrique va nous permettre de mesurer le courant qui circule dans un conducteur.

Cependant, on ne peut pas câbler directement le capteur sur l'arduino, car le courant mesuré est alternatif, alors que l'arduino lui utilise du courant continu. Nous allons donc devoir réaliser un petit montage avec un condensateur et deux résistances pour redresser le courant. Le schéma ci-dessous nous montre comment câbler cela à l'arduino :

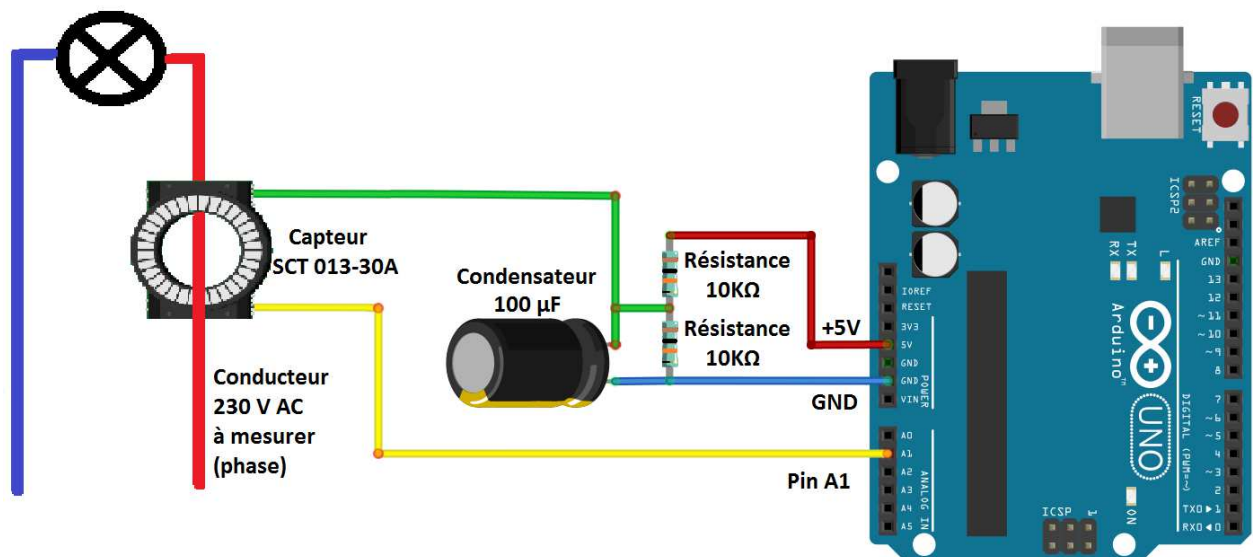


Figure 1.1 : Schéma de raccordement du capteur à l'arduino

Le capteur SCT013 à pour raccordement une prise jack avec 3 points de raccordement. Il faut utiliser les points de chaque extrémité. L'un est raccordé au pin A1 de l'arduino, l'autre est raccordé au montage de redressement.

Le montage de redressement se compose d'un condensateur 100 µFarad, et de deux résistances de 10 KΩ. Une résistance est raccordée au GND de l'arduino, l'autre au 5 Volts, et ces deux résistances sont raccordées ensemble. Le condensateur est positionné en parallèle de la résistance raccordée au neutre. On obtient ainsi un point "milieu" représenté par le fil vert. L'autre borne du capteur est raccordé sur ce point milieu. Le câblage est terminé.

Il ne nous reste plus qu'à écrire le code ci-dessous dans l'arduino :



```
code_bon_mesure_stc013 | Arduino 1.8.5
Fichier Édition Croquis Outils Aide

code_bon_mesure_stc013 $

#include <EmonLib.h> // On inclus la bibliotheque emonlib pour controler le capteur de courant
#include <Wire.h> //On inclus la bibliotheque wire pour controler l'ecran LCD
#include <LiquidCrystal_I2C.h> //On inclus la bibliotheque liquidcrystal I2C pour controler l'ecran en protocole I2C
LiquidCrystal_I2C lcd(0x27, 16, 2); //On declare l'adresse I2C de l'ecran, ainsi que sa taille
EnergyMonitor emon1;
int tension = 230.0; // on declare la tension secteur
int pin_sct = 1; //Capteur raccorde sur le pin A1

void setup() Vidéo complète disponible sur la chaine Youtube Jeremyrenove
{
  lcd.init();//on initialise l'ecran LCD
  lcd.backlight(); //On active le rétroéclairage de l'ecran
  Serial.begin(9600); //On active le moniteur série
  emon1.current(pin_sct, 29);//Pin, calibration - Courant Const= Ratio/Res. Burder. 1800/62 = 29.
  lcd.setCursor(0,0); //on positionne le curseur de l'ecran LCD en coordonnées x, y,
  lcd.print("I. (A):"); //on ecrit le texte qui se trouve entre " les guillemets "
  lcd.setCursor(0,1); //on positionne le curseur de l'ecran LCD en coordonnées x, y,
  lcd.print("P. (W):"); //on ecrit le texte qui se trouve entre " les guillemets "
}

void loop() Vidéo complète disponible sur la chaine Youtube Jeremyrenove
{
  double Irms = emon1.calcIrms(1480);//calcul du courant
  Serial.print("Courant : "); //on affiche dans le moniteur série le texte qui se trouve entre " les guillemets "
  Serial.print(Irms); //on affiche dans le moniteur série la variable qui se trouve entre les ( )
  lcd.setCursor(10,0); //on positionne le curseur de l'ecran LCD en coordonnées x, y,
  lcd.print(Irms); //on affiche sur l'ecran LCD la variable qui se trouve entre les ( )

  Serial.print(" Puissance : "); //on affiche dans le moniteur série le texte qui se trouve entre " les guillemets "
  Serial.println(Irms*tension); //on affiche dans le moniteur série la variable qui se trouve entre les ( )
  lcd.setCursor(10,1); //on positionne le curseur de l'ecran LCD en coordonnées x, y,
  lcd.print(" "); //on affiche dans le moniteur série le texte qui se trouve entre " les guillemets "
  lcd.setCursor(10,1); //on positionne le curseur de l'ecran LCD en coordonnées x, y,
  lcd.print(Irms*tension,1); //on affiche sur l'ecran LCD la variable qui se trouve entre les ( )

  delay(1000); //On attend 1000 milisecondes avant de recommencer le cycle du programme
}

15 Arduino/Genuino Uno sur COM1
```

Figure 1.2 : code de programmation du capteur pour l'arduino

A noter que sur ce code montage était monté un écran LCD qui n'est pas représenté sur le schéma.