

## LUIS LAMAS

*Engineering, Computing und Design*

ARDUINO-TUTORIALS FORTGESCHRITTENE ARDUINO-TUTORIALS

# NICHT-INVASIVER ELEKTRISCHER STROMSENSOR MIT ARDUINO UND SCT-013

24. JANUAR 2017



## WAS IST EIN SCT-013-SENSOR?

Die SCT-013-Familie besteht aus nicht-invasiven Stromsensoren, die die Intensität messen, durch die ein Leiter läuft, ohne dass der Leiter geschnitten oder modifiziert werden muss. Wir können diese Sensoren mit einem Prozessor wie Arduino verwenden, um die Intensität oder Leistung einer Last zu messen

SCT-013-Sensoren sind Stromwandler, Instrumentierungsgeräte, die eine Messung liefern, die proportional zum durch einen Stromkreis

fließenden Strom ist. Die Messung erfolgt durch elektromagnetische Induktion.

SCT-013-Sensoren haben einen geteilten ferromagnetischen Kern (wie eine Klemme), mit dem sie geöffnet werden können, um einen Leiter einer elektrischen Anlage zu wickeln, ohne ihn abschneiden zu müssen.

Innerhalb der SCT-013-Familie gibt es Modelle, die Messungen als Strom- oder Spannungsausgang ermöglichen. Es ist normal, dass wir den Spannungsausgang bevorzugen, da die Verbindung einfacher ist.

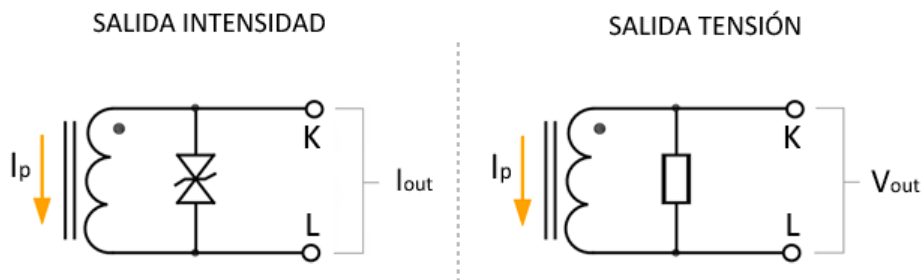
Anzeige:

### 3D Print Strong as

Anuncio Markforged 3  
print 23x stronger tha

Markforged

Get Quote

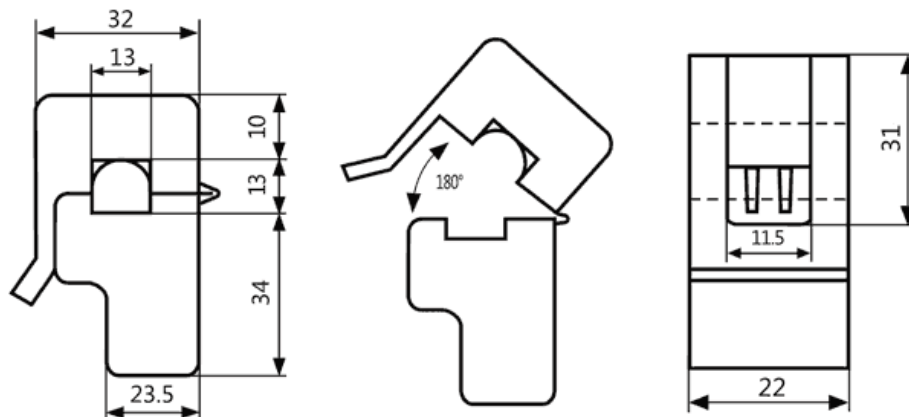


Die Genauigkeit des Sensors kann 1-2% betragen, aber dafür ist es sehr wichtig, dass der ferromagnetische Kern richtig schließt. Schon ein kleiner Luftspalt kann zu Abweichungen von 10% führen.

Da es sich um eine induktive Last handelt, führt der SCT-013 eine Variation des Phasenwinkels ein, dessen Wert eine Funktion der durch ihn hindurchtretenden Last ist und bis zu 3 ° betragen kann.

Stromwandler sind häufige Komponenten in der industriellen Welt und in der elektrischen Verteilung und ermöglichen die Überwachung des Verbrauchs von Verbrauchspunkten, an denen eine andere Art der Messung unmöglich wäre. Sie sind auch Teil mehrerer Messinstrumente, selbst bei tragbaren Geräten wie Begrenzungsklemmen oder Netzwerkanalysatoren.

In unseren Elektronik- und Hausautomationsprojekten können wir SCT-013-Stromsensoren verwenden, um beispielsweise den Stromverbrauch des Geräts zu messen, den Status einer elektrischen Installation zu überprüfen, und sie sind häufige Komponenten in Hausenergiemonitoren, um den Verbrauch von zu erfassen eine Installation oder sogar Zugriff über das Internet in Echtzeit.



## PREIS

Innerhalb der SCT-013-Familie gibt es mehrere Modelle, die den Messbereich und die Form der Ausgabe variieren. Physikalisch sind sie identisch, obwohl es möglich ist, sie anhand des auf dem Gehäuse geschriebenen Textes zu identifizieren.

Der Preis von allen ist ähnlich, wir können sie für 4-4,50 € finden, suchen in internationalen Verkäufern von eBay oder AliExpress.



Am häufigsten sind der SCT-013-000 für einen maximalen Strom von 100 A und einen Stromausgang von 50 mA (100 A: 50 mA) und der SCT-013-030 für maximale Ströme von 30 A (30 A / 1 V) und einen Spannungsausgang von 1V.

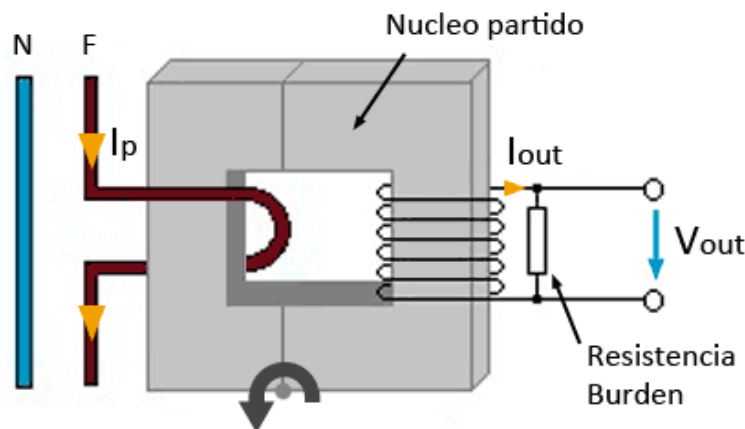
Obwohl der SCT-013-000 in Online-Shops sehr häufig ist, werden wir normalerweise Modelle mit Spannungsausgang bevorzugen.

Schließlich ist es wichtig, einen weiten Messbereich zu haben, aber es muss berücksichtigt werden, dass ein Modell mit einer höheren maximalen Intensität zu einer geringeren Präzision führt. Eine Intensität von 30 A bei 230 V entspricht einer Last von 6.900 W, was für die meisten Heimanwender ausreichend ist.

## WIE FUNKTIONIERT EIN SCT-013?

Die SCT-013-Sensorfamilie besteht aus kleinen Stromwandlern oder Stromwandlern. Stromwandler sind weit verbreitete Instrumente als Messelemente.

Ein Stromwandler ähnelt einem [Spannungswandler](#) und basiert auf denselben Funktionsprinzipien (tatsächlich ist er formal identisch). Sie verfolgen jedoch unterschiedliche Ziele und sind daher unterschiedlich konzipiert und konstruiert.



Ein Stromwandler versucht, eine Intensität in der Sekundärseite zu erzeugen, die proportional zum Strom durch die Primärwicklung ist. Hierzu ist es erwünscht, dass die Primärwicklung mit einer reduzierten Anzahl von Windungen gebildet wird.

Damit können wir einen nicht-invasiven Stromsensor bauen. In einem Stromsensor kann der ferromagnetische Kern geteilt werden, so dass ein

Leiter geöffnet und gewickelt werden kann.

Auf diese Weise haben wir einen Transformator, in dem:

- ➔ Das Kabel, durch das der zu messende Strom zirkuliert, bildet eine Primärwicklung
- ➔ Die "Klammer" ist der Magnetkern
- ➔ Die Sekundärwicklung ist als Teil der Sonde integriert.

Wenn der Wechselstrom durch den Leiter zirkuliert, wird im ferromagnetischen Kern ein magnetischer Fluss erzeugt, der wiederum einen elektrischen Strom in der Sekundärwicklung erzeugt.

Das Intensitätsumwandlungsverhältnis hängt vom Verhältnis zwischen der Anzahl der Windungen ab.

$$\frac{I_s}{I_p} = \frac{V_p}{V_s} = \frac{N_p}{N_s}$$

Die Primärwicklung besteht im Allgemeinen aus einer einzelnen Schleife, die von dem zu messenden Leiter gebildet wird. Obwohl es möglich ist, den Leiter so aufzuwickeln, dass er mehr als einmal in der "Klemme" vorbeigeht. Die Anzahl der in die Sonde integrierten Sekundärwindungen variiert je nach SCT-013-Modell zwischen 1000 und 2000.

Im Gegensatz zu Spannungswandlern sollte in einem Stromwandler der Sekundärkreis niemals offen sein, da induzierte Ströme das Bauteil beschädigen können. Aus diesem Grund verfügen die SCT-130-Sensoren über Schutzfunktionen (Widerstandsbelastung der Spannungsausgangssensoren oder Schutzdioden der Stromausgangssensoren).

## MONTAGESCHEMA

Um die Verbindung des SCT-013-Sensors zu verstehen, müssen wir drei Probleme verstehen und lösen:

- ➔ Sensorausgang in Intensität
- ➔ Spannungsbereichsanpassung

➔ Positive und negative Spannungen

## SENSORAUSGANG IN INTENSITÄT

Die SCT-013 sind Stromwandler, dh die Messung wird als Stromsignal erhalten, das proportional zum durch das Kabel fließenden Strom ist. Prozessoren können jedoch nur Spannungen messen.

Dieses Problem ist leicht zu lösen. Um den Stromausgang in einen Spannungsausgang umzuwandeln, müssen wir nur einen Widerstand (Widerstandsbelastung) einbeziehen.

Mit Ausnahme des Modells SCT-013-100 verfügen alle anderen Modelle über einen internen Belastungswiderstand, sodass der Ausgang ein 1-V-Spannungssignal ist. Wir müssen uns also nicht einmal darum kümmern.

Nur im Fall von SCT-013-100 fehlt der interne Belastungswiderstand, sodass der Ausgang ein  $\pm 50\text{-mA}$ -Signal ist. Ein  $33\Omega$  Widerstand parallel zum Sensor reicht aus.

## POSITIVE UND NEGATIVE SPANNUNGEN

Ein weiteres Problem, das wir lösen müssen, besteht darin, dass wir Wechselstrom messen und die in der Sekundärseite induzierte Intensität gleichermaßen abwechselnd ist. Nach dem Durchlaufen des Lastwiderstands (intern oder extern) wechselt auch der Spannungsausgang.

Wie wir jedoch wissen, können die analogen Eingänge der meisten Prozessoren, einschließlich Arduino, nur positive Spannungen messen.

Um die Transformatorausgangsspannungen zu messen, haben wir verschiedene Möglichkeiten, vom schlechtesten zum besten.

Korrigieren Sie das Signal mit einer Diodenbrücke und messen Sie die Welle als positive Werte. Nicht zu empfehlen, da wir die Information verlieren, ob wir uns in der negativen oder positiven Halbperiode befinden, auch weil der Spannungsabfall der Diode auftritt und die Diode, noch schlimmer, nicht unter eine Spannung fährt, sodass das Signal an den Verbindungsstellen verzerrt wird um Null.

Fügen Sie einen Offset in DC hinzu, indem Sie zwei Widerstände und einen Kondensator verwenden, die einen Mittelpunkt zwischen GND und  $V_{cc}$  liefern. Viel besser, wenn wir auch einen Operationsverstärker als Spannungsfolger hinzufügen.

Fügen Sie einen ADC mit Differenzeingang hinzu, mit dem positive und negative Spannungen wie der ADS1115 gemessen werden können. Es ist die Option, die wir verwenden werden.

## SPANNUNGSBEREICHSANPASSUNG

Das letzte Problem ist, dass wir den Spannungsbereich am Sensorausgang anpassen müssen. Arduino kann nur Messungen zwischen 0 und  $V_{cc}$  durchführen. Je kleiner der Bereich, desto mehr Präzision verlieren wir. Daher sollten wir uns auf diesen Bereich einstellen.

Andererseits müssen wir berücksichtigen, dass bei Wechselspannung normalerweise RMS-Werte verwendet werden. Erinnern wir uns kurz an die Spannungsgleichungen von Spitze zu Spitze.

$$V_{pico} = \sqrt{2} \cdot V_{RMS} = R \cdot I_{pico} = \sqrt{2} \cdot R \cdot I_{RMS}$$

$$V_{picoapico} = 2 \cdot V_{pico} = 2 \cdot \sqrt{2} \cdot V_{RMS} = R \cdot I_{picoapico} = 2 \cdot \sqrt{2} \cdot R \cdot I_{RMS}$$

Daher beträgt bei Sensoren mit  $\pm 1$  V RMS-Ausgang die Spitzenspannung  $\pm 1.414$  V und die Spitze-Spitze-Spannung 2.828 V.

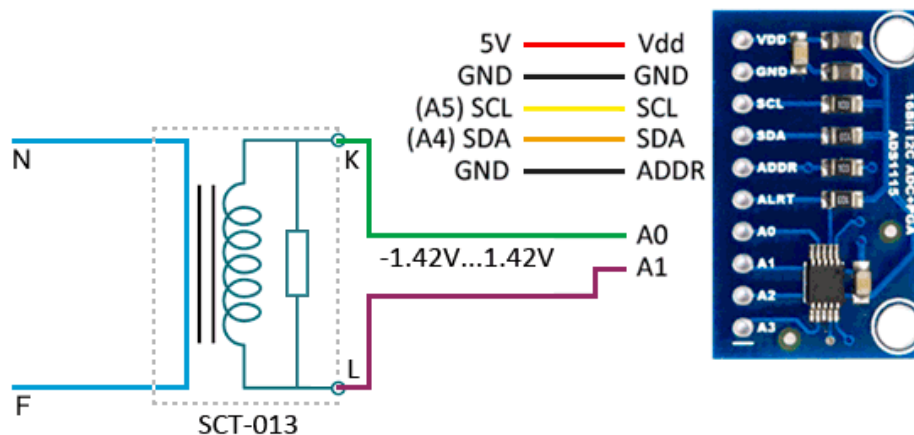
Im Fall von SCT-013-100 beträgt der Ausgang  $\pm 50$  mA. Bei einem externen Belastungswiderstand von  $33 \Omega$  beträgt die Ausgangsspannung  $\pm 1,65$  V RMS und daher eine Spitzenspannung von  $\pm 2,33$  V und eine Spitze-Spitze-Spannung von 4,66 V.

## ELEKTRISCHER ANSCHLUSS

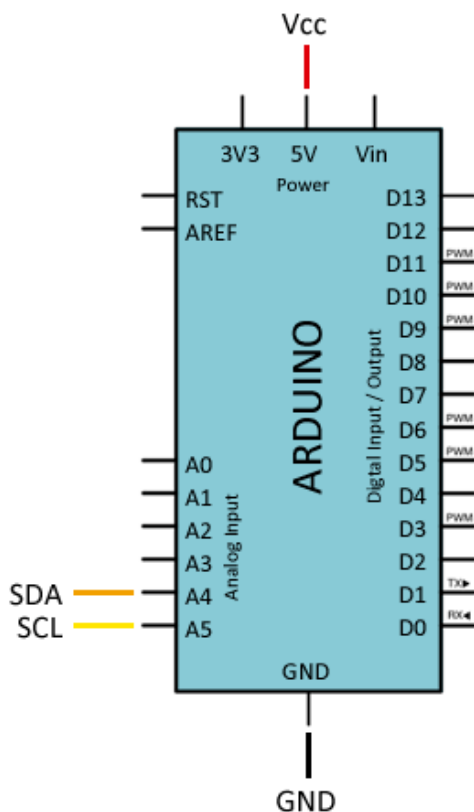
Wir haben bereits alle Komponenten, um die Netzwerkintensität mit einem SCT-013-Sensor zu messen. Wir werden einen Sensor mit  $\pm 1$  V RMS-Spannungsausgang und internem Belastungswiderstand sowie einen ADC wie den ADS1115 im Differentialmodus verwenden.

Durch Einstellen der Verstärkung des ADS1115 auf 2.048 V liegen wir im Bereich von  $\pm 1.414$  V. Im Fall eines 30A-Sensors haben wir eine Genauigkeit von 1,87 mA und 6,25 mA für einen 100A-Sensor.

Wenn Sie einen SCT-013-100 mit  $\pm 50$  mA Ausgang verwenden, müssen wir einen externen 33  $\Omega$ -Lastwiderstand hinzufügen und die Verstärkung des ADS1115 auf 4,096 V erhöhen, um den Bereich von  $\pm 2,33$  V zu erreichen.



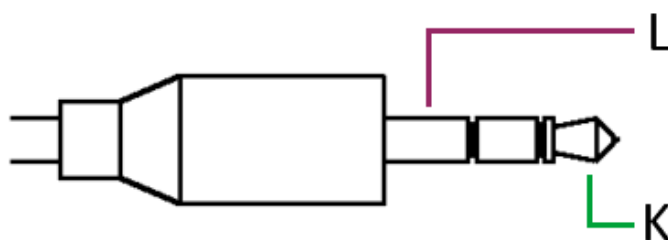
Die von Arduino aus gesehene Verbindung wäre nur die Stromversorgung des ADS1115-Moduls, wie wir im Eintrag zum ADS1115 gesehen haben .





Um die Messungen durchzuführen, ist es wichtig, dass wir nur einen der Leiter mit der "Klemme" nehmen. Wenn wir mehr als einen Leiter nehmen (zwei Leiter in einphasigen Installationen, drei in dreiphasigen Installationen), würde die Wirkung der Leiter aufgehoben. Dies erzeugt eine Nullinduktion und daher eine Nullmessung.

SCT-013-Sensoren verfügen über einen Jack 3.5-Anschluss, der in Audio sehr verbreitet ist, aber in unseren Elektronikprojekten nicht besonders komfortabel zu verwenden ist. Um es anzuschließen, müssen wir entweder das Kabel abschneiden oder eine Buchse erwerben, an die wir die Kabel anlöten. Glücklicherweise sind sie leicht zu erwerben, schließen aber auch das Abschneiden des Kabels nicht aus.

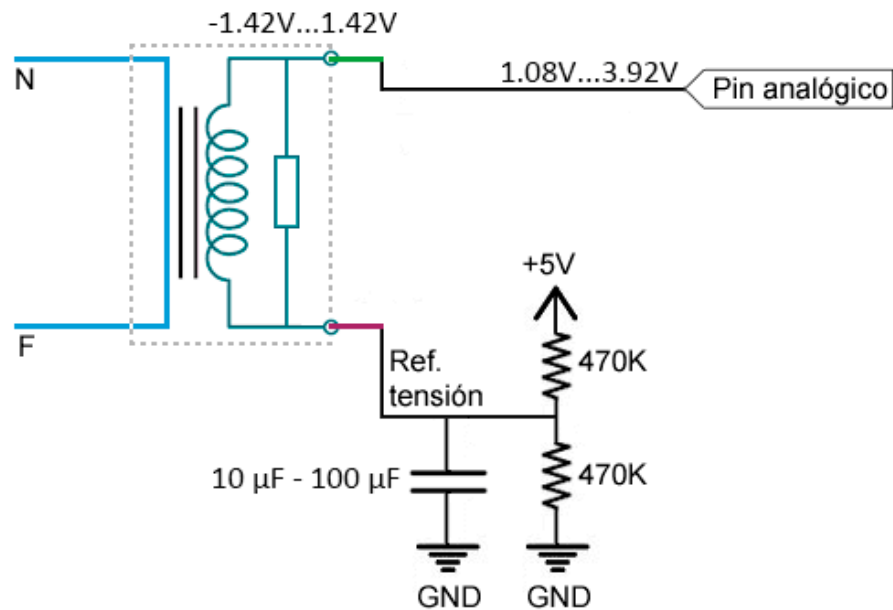


Wenn Sie keinen externen ADC verwenden möchten, können Sie die herkömmlichste Lösung verwenden, nämlich das Hinzufügen einer Schaltung, mit der wir einen zentralen Versatz hinzufügen können.



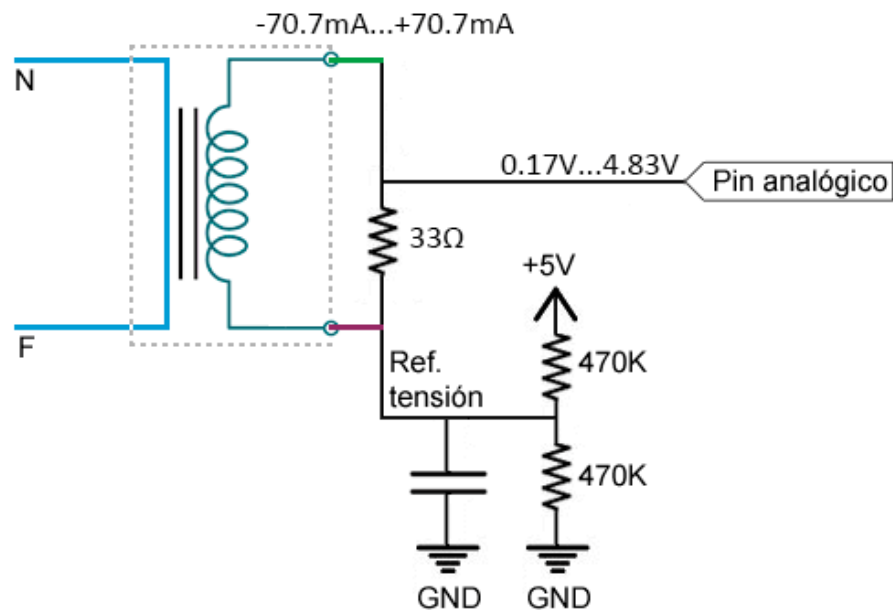
**Gehen vorwärts, gehen wir davon aus, dass Sie eine Vcc 5V**

**Arduino verwenden**. Wenn Sie einen anderen Prozessor oder ein Arduino-Modell mit einem anderen Vcc (z. B. 3,3 V) verwenden, müssen Sie die Baugruppe entsprechend korrigieren.



Wo wir einen 2,5-V-DC-Offset-Punkt hinzugefügt haben, liegt der endgültige Bereich zwischen 1,08 V und 3,92 V, innerhalb des Bereichs der analogen Eingänge wird ein Arduino mit 5 V betrieben.

Bei Verwendung des SCT-013-100 mit einem Ausgang von  $\pm 50$  mA müssen wir einen externen Belastungswiderstand von 33 Ohm hinzufügen, und der Endbereich liegt zwischen 0,17 V und 0,483 V, ebenfalls im Bereich der Analogeingänge eines Arduino bis 5V.



## CODEBEISPIELE

## MONTAGE MIT ADS1115

Si habéis usado el montaje con un SCT-013 con salida de  $\pm 1V$  RMS y ADS1115, el código necesario es similar al que vimos en la entrada sobre el ADS1115. Necesitaréis la [librería de Adafruit para el ADS1115](#).

Para que el ADS1115 muestree a una velocidad superior, deberemos modificar la siguiente línea del fichero 'Adafruit\_ADS1015.h'

```
1 | #define ADS1115_CONVERSIONDELAY      (8)
```

Por,

```
1 | #define ADS1115_CONVERSIONDELAY      (1)
```

Con esto conseguiremos bajar el tiempo de muestreo de unos 8-9 ms (unos 100 Hz) a 1.8 aprox (unos 500 Hz). Con eso nos alejamos de la frecuencia de Nyquist, y mejoramos el comportamiento de la medición.

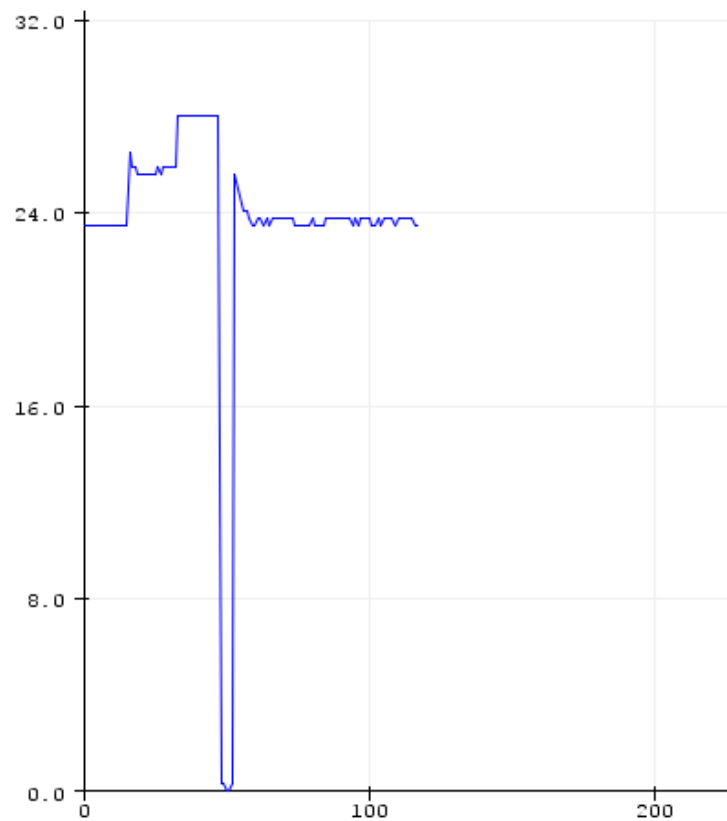
```
1 | #include <Wire.h>
2 | #include <Adafruit_ADS1015.h>
3 |
4 | Adafruit_ADS1115 ads;
5 |
6 | const float FACTOR = 30; //30A/1V
7 |
8 | const float multiplier = 0.0625F;
9 |
10 | void setup()
11 | {
12 |   Serial.begin(9600);
13 |
14 |   ads.setGain(GAIN_TWO);           // ±2.048V  1 bit = 0.0625mV
15 |   ads.begin();
16 | }
17 |
18 | void printMeasure(String prefix, float value, String postfix)
19 | {
20 |   Serial.print(prefix);
21 |   Serial.print(value, 3);
22 |   Serial.println(postfix);
23 | }
24 |
25 | void loop()
26 | {
27 |   float currentRMS = getCorriente();
28 |   float power = 230.0 * currentRMS;
29 |
30 |   printMeasure("Irms: ", currentRMS, "A ,");
31 |   printMeasure("Potencia: ", power, "W");
32 |   delay(1000);
33 | }
34 |
35 | float getCorriente()
36 | {
37 |   float voltage;
38 |   float corriente;
39 |   float sum = 0;
40 |   long tiempo = millis();
41 |   int counter = 0;
42 |
```

```
43 while (millis() - tiempo < 1000)
44 {
45     voltage = ads.readADC_Differential_0_1() * multiplier;
46     corriente = voltage * FACTOR;
47     corriente /= 1000.0;
48
49     sum += sq(corriente);
50     counter = counter + 1;
51 }
52
53 corriente = sqrt(sum / counter);
54 return(corriente);
55 }
```

Otra versión es emplear el máximo y el mínimo de la medición, y calcular la medición a partir del valor de pico. Los resultados deberían ser similares a los vistos en el ejemplo con la suma al cuadrado. Para ello, podéis sustituir la función por la siguiente.

```
1 float getCorriente()
2 {
3     long tiempo = millis();
4     long rawAdc = ads.readADC_Differential_0_1();
5     long minRaw = rawAdc;
6     long maxRaw = rawAdc;
7     while (millis() - tiempo < 1000)
8     {
9         rawAdc = ads.readADC_Differential_0_1();
10        maxRaw = maxRaw > rawAdc ? maxRaw : rawAdc;
11        minRaw = minRaw < rawAdc ? minRaw : rawAdc;
12    }
13
14    maxRaw = maxRaw > -minRaw ? maxRaw : -minRaw;
15    float voltagePeak = maxRaw * multiplier / 1000;
16    float voltageRMS = voltagePeak * 0.70710678118;
17    float currentRMS = voltageRMS * FACTOR;
18    return(currentRMS);
19 }
```

Podemos ver los resultados en el [monitor del puerto serie](#), graficarlo con el [serial plotter](#), o recogerlo en un proyecto mayor para mostrarlo en una página web, o registrarlo en una SD.



## MONTAJE CON RESISTENCIAS Y PUNTO MEDIO

En este caso el ejemplo es muy sencillo, únicamente tenemos que realizar la medición mediante una [entrada analógica](#).

```

1  const float FACTOR = 30; //30A/1V
2
3  const float VMIN = 1.08;
4  const float VMAX = 3.92;
5
6  const float ADCV = 5.0; //Para Vcc
7  //const float ADCV = 1.1; //Para referencia interna
8
9
10 void setup()
11 {
12     Serial.begin(9600);
13     //analogReference(INTERNAL);
14 }
15
16 void printMeasure(String prefix, float value, String postfix)
17 {
18     Serial.print(prefix);
19     Serial.print(value, 3);
20     Serial.println(postfix);
21 }
22
23 void loop()
24 {
25     float currentRMS = getCorriente();
26     float power = 230.0 * currentRMS;
27
28     printMeasure("Irms: ", currentRMS, "A ,");
29     printMeasure("Potencia: ", power, "W");

```

```
30     delay(1000);
31 }
32
33 float getCorriente()
34 {
35     float voltage;
36     float corriente;
37     float sum = 0;
38     long tiempo = millis();
39     int counter = 0;
40
41     while (millis() - tiempo < 500)
42     {
43         voltage = analogRead(A0) * ADCV / 1023.0;
44         corriente = fmap(voltage, VMIN, VMAX, -FACTOR, FACTOR);
45
46         sum += sq(corriente);
47         counter = counter + 1;
48         delay(1);
49     }
50
51     corriente = sqrt(sum / counter);
52     return(corriente);
53 }
54
55 // cambio de escala entre floats
56 float fmap(float x, float in_min, float in_max, float out_min, float out_max)
57 {
58     return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
59 }
```



Si te ha gustado esta entrada y quieres leer más sobre Arduino puedes consultar la sección **tutoriales de Arduino**

**Anuncio:**

🔧 Arduino, Arduino intermedio, Electricidad, Sensores

---

PREVIOUS

**MEDIR INTENSIDAD Y CONSUMO ELÉCTRICO CON ARDUINO Y ACS712**