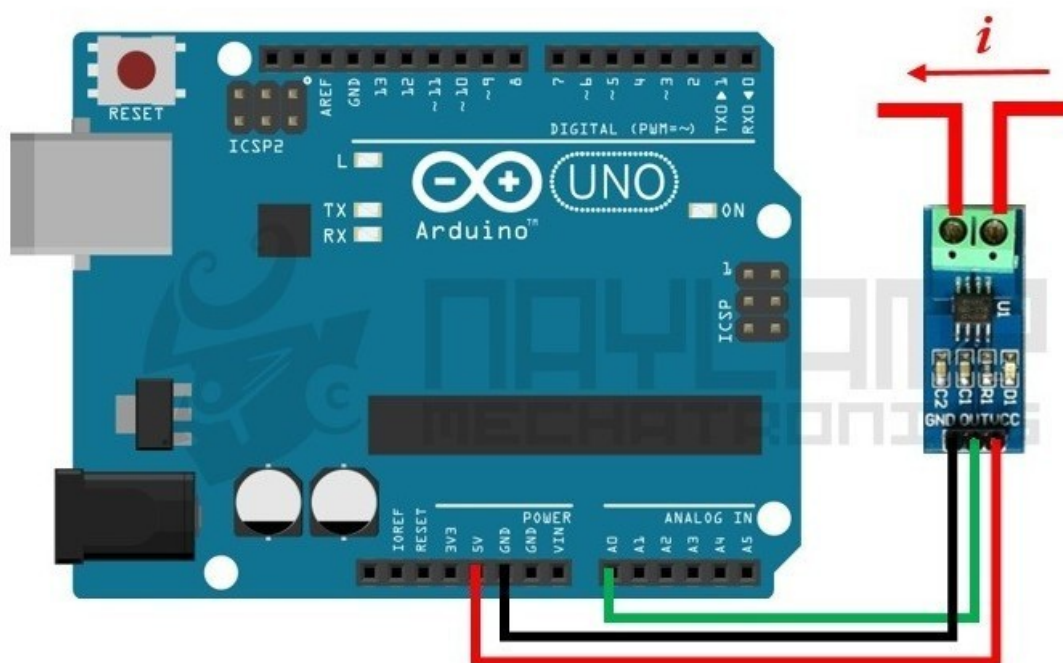


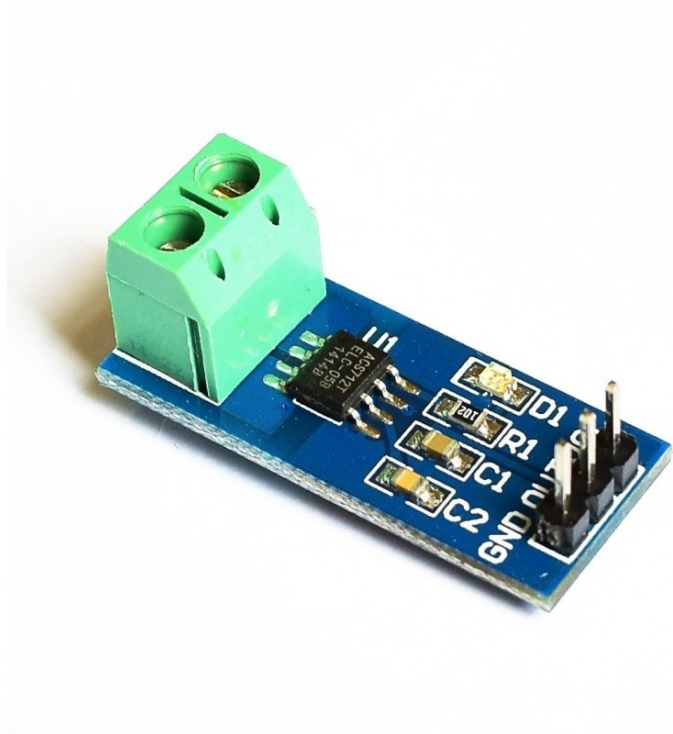
# TUTORIAL SENSOR DE CORRIENTE ACS712

263777



El sensor de corriente ACS712 es una solución económica para medir corriente, internamente trabaja con un sensor de efecto Hall que detecta el campo magnético que se produce por inducción de la corriente que circula por la línea que se está midiendo. EL sensor nos entrega una salida de voltaje proporcional a la corriente, dependiendo la aplicación podemos usar el ACS712-05A, ACS712-20A o el ACS712-30A, para rangos de 5, 20 o 30 amperios respectivamente

El ACS712 podemos encontrarlo en módulos, los cuales nos facilitan sus conexión, traen una bornera para conectar la línea que queremos medir y 3 pines, dos para conectar la alimentación y un pin para la salida analógica.



El rango de corriente que podemos medir y sensibilidad varían dependiendo del modelo del integrado, existen tres modelos los cuales detallamos a continuación:

Modelo	Rango	Sensibilidad
ACS712ELCTR-05B-T	-5 a 5 A	185 mV/A
ACS712ELCTR-20A-T	-20 a 20 A	100 mV/A
ACS712ELCTR-30A-T	-30 a 30 A	66 mV/A

El sensor nos entrega un valor de 2.5 voltios para una corriente de 0A y a partir de allí incrementa proporcionalmente de acuerdo a la sensibilidad, teniendo una relación lineal entre la salida de voltaje del sensor y la corriente. Dicha relación es una línea recta en una gráfica Voltaje vs Corriente donde la pendiente es la sensibilidad y la intersección en el eje Y es 2.5 voltios. La ecuación de la recta sería la siguiente

$$V = m I + 2.5$$

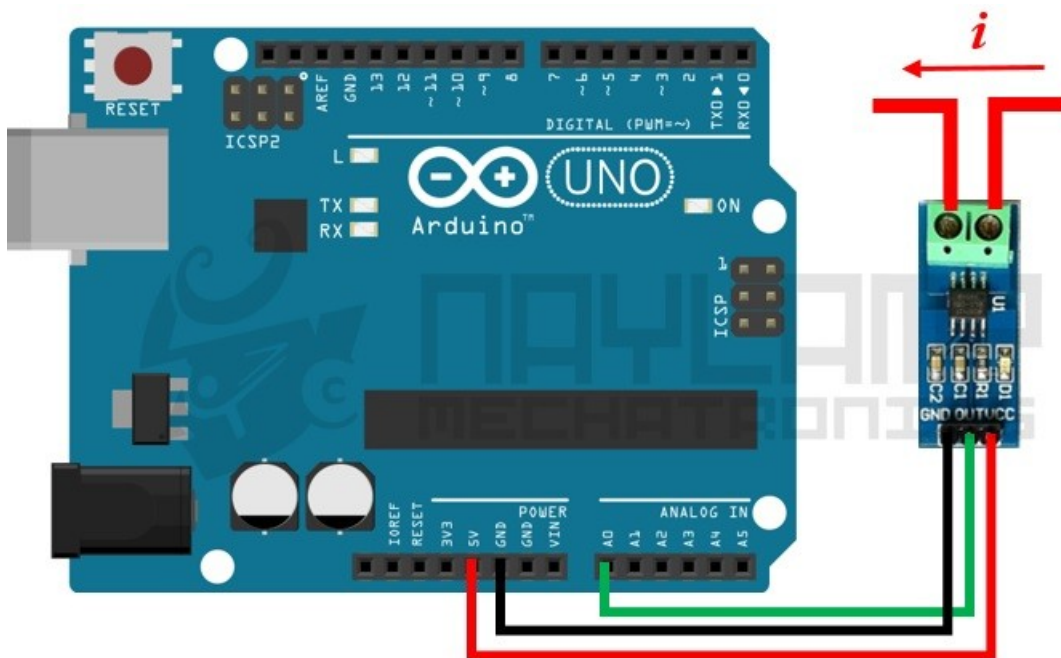
Donde la pendiente es m y equivale a la Sensibilidad

Despejando tendremos la ecuación para hallar la corriente a partir de la lectura del sensor:

$$I = \frac{V - 2.5}{Sensibilidad}$$

Con esta ecuación podemos pasar a realizar los ejemplos con Arduino:

## Conexiones entre Arduino y módulo ACS712



Para las conexiones en el módulo guiarse por los nombres de los pines, en algunos modelos vienen en diferente orden, en la bornera ingresa la línea de la cual se desea medir, para medir la corriente se debe conectar en serie con el dispositivo o carga, nunca conectar en paralelo a la fuente de voltaje.

## Ej.1 Realizando medidas de corriente con el ACS712

Para realizar la lectura de corriente simplemente se necesita leer la entrada analógica y con la formula antes expuesta obtener la corriente.

A continuación se muestra el código para un realizar la lectura de corriente:

```
float Sensibilidad=0.185; //sensibilidad en Voltios/Amperio para sensor de 5A

void setup() {

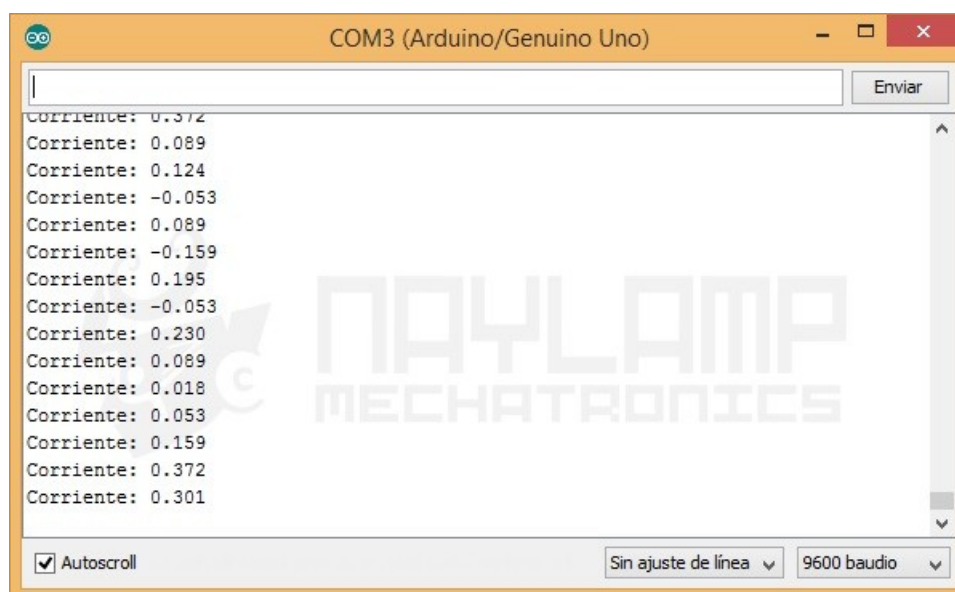
    Serial.begin(9600);
}

void loop() {

    float voltajeSensor= analogRead(A0)*(5.0 / 1023.0); //lectura del sensor
    float I=(voltajeSensor-2.5)/Sensibilidad; //Ecuación para obtener la corriente
    Serial.print("Corriente: ");
    Serial.println(I,3);
    delay(200);
}
```

En nuestro caso estamos trabajando con un sensor de 5A por eso usamos el valor de sensibilidad de 0.185V/A que es el equivalente 185mV/A que nos da el fabricante, si están trabajando con el sensor de 20A, reemplazar el valor de la sensibilidad por 0.100 V/A.

A continuación mostramos los resultados obtenidos del monitor serial:



La Imagen anterior es para una lectura de una corriente de 0A, notar que existe ruido en la lectura, si para su aplicación el ruido presente no tiene mucha importancia, podrían trabajar

con el programa de este ejercicio, sino se tendrá que aplicar un filtro, que se muestra a continuación.

## Ej.2 Aplicando Filtro a las lecturas del ACS712

Existen varios tipos de filtros, que dependiendo de la complejidad pueden consumir recursos en la programación de nuestro Arduino, en nuestro caso simplemente usaremos la media aritmética de varias lecturas consecutivas, implementar el promedio de las lecturas en Arduino es sencillo y fácil de entender, simplemente hay que sumar la lecturas y dividir las en el número de muestras.

La cantidad de muestras para calcular el promedio depende del nivel de ruido que tengan.

El programa sería el siguiente:

```
float Sensibilidad=0.185; //sensibilidad en Voltios/Amperio para sensor de 5A

void setup() {

    Serial.begin(9600);

}

void loop() {

    float I=get_corriente(200); //obtenemos la corriente promedio de 500 muestras

    Serial.print("Corriente: ");

    Serial.println(I,3);

    delay(100);

}

float get_corriente(int n_muestras)
{
    float voltajeSensor;

    float corriente=0;

    for(int i=0;i<n_muestras;i++)
    {

        voltajeSensor = analogRead(A0) * (5.0 / 1023.0); //lectura del sensor

        corriente=corriente+(voltajeSensor-2.5)/Sensibilidad; //Ecuación para obtener la corriente

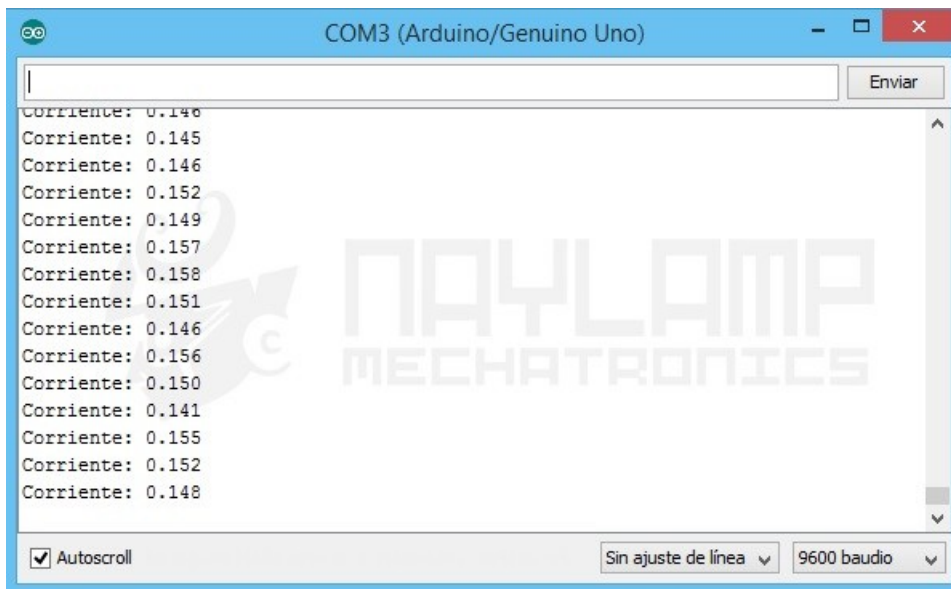
    }

    corriente=corriente/n_muestras;

    return(corriente);

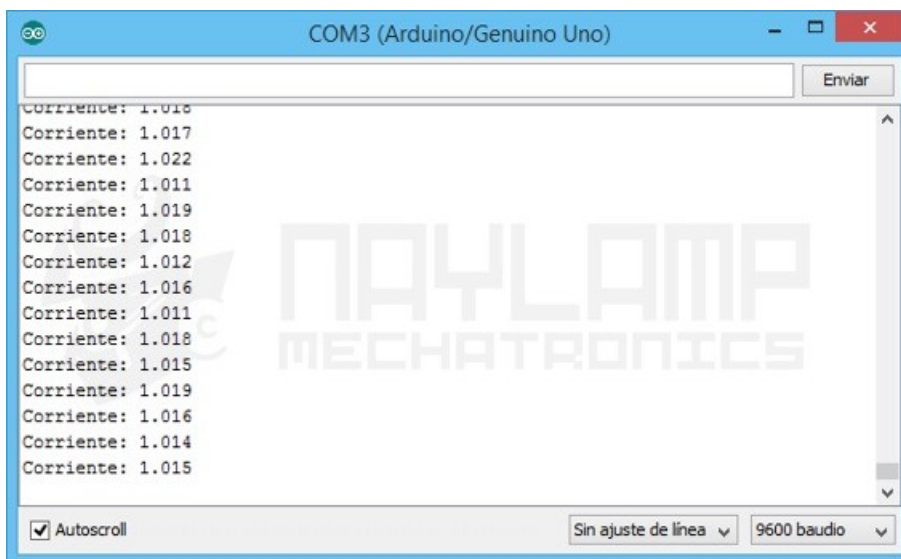
}
```

En nuestro caso debido a que hay bastante ruido trabajamos con 200 muestras, mientras mayor sea la cantidad de muestras obtendremos un mejor resultado pero también mayor será el tiempo que se demora el Arduino para realizar la medición, para 200 muestras Arduino se demora alrededor de 35 milisegundos que se lo puede considerar despreciable en la mayoría de aplicaciones.



Notar que ahora el ruido es de amplitud 10mA en comparación al ejemplo anterior que era superior a 100mA, No confundir el ruido con el error del offset que en este caso es de aproximadamente +150 mA , este último error si es representativo podemos restar al resultado en la ecuación de la corriente.

El siguiente resultado es para una carga de aproximadamente 1.2A



Del sensor obtenemos una medida de 1.02A aproximadamente, pero si medimos con un multitester o amperímetro el valor de la corriente es de 1.15, en nuestro caso esto nos da un error de -130 mA que es diferente al offset o el error para una corriente de 0A. Si sus resultados también son como el nuestro, significa que la ecuación que estamos usando para calcular la corriente no es la adecuada, para corregir esto es necesarios calibrar nuestro sensor y hallar los nuevos valores de sensibilidad y el voltaje equivalente a los 0A, valores que necesitamos para la ecuación.

## Ej.3 Calibrando nuestro ACS712

Si el error que obtenemos con los ejemplos anteriores es grande, debemos de recalibrar y hallar los valores reales puesto que los valores que nos da el fabricante no son exactos.

Recordemos la ecuación que usamos para hallar la corriente:

$$I = \frac{V - 2.5}{\text{Sensibilidad}}$$

En esta ecuación solo tenemos dos constantes: los 2.5 que es el voltaje del sensor cuando la corriente es 0V y la sensibilidad, que equivale a la pendiente de la recta Voltaje Vs Corriente; tenemos que calcular los valores reales de estas dos constantes.

Al ser una recta, basta con obtener dos puntos y con estos podemos calcular constantes. Para esto necesitamos un multitester o amperímetro, el cual debe ser de buena precisión, ya que este será la herramienta para calibrar. El amperímetro, el ACS712 y la carga deben de conectarse en serie.

Después de conectar el amperímetro y el sensor, debemos de tomar lecturas de voltaje del sensor, esto se hace con el siguiente programa.

```
void setup() {

    Serial.begin(9600);
}

void loop() {

    float voltajeSensor =get_voltaje(10000);//obtenemos voltaje del sensor(10000 muestras)

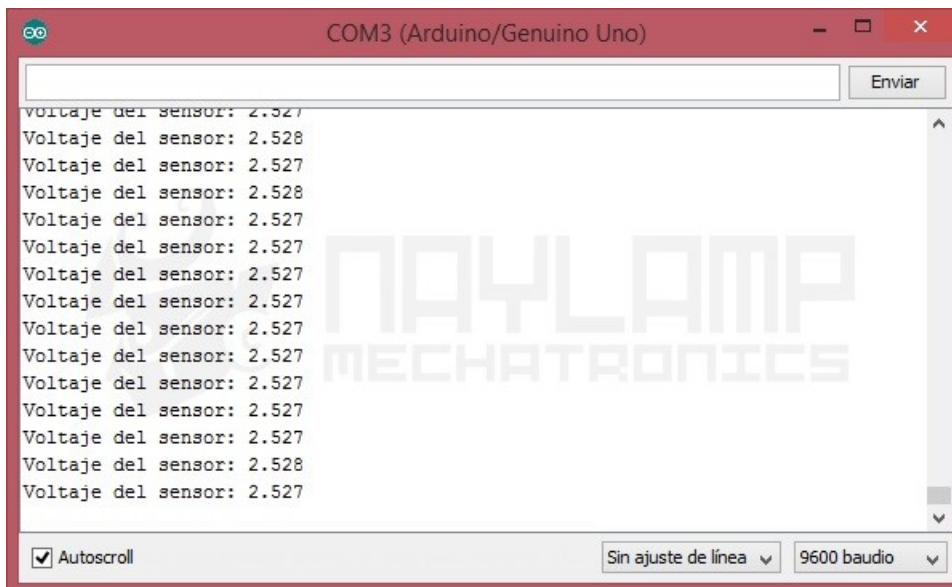
    Serial.print("Voltaje del sensor: ");
    Serial.println(voltajeSensor ,3);
}

float get_voltaje(int n_muestras)
{
    float voltage=0;

    for(int i=0;i<n_muestras;i++)
    {
        voltage =voltage+analogRead(A0) * (5.0 / 1023.0);
    }
    voltage=voltage/n_muestras;

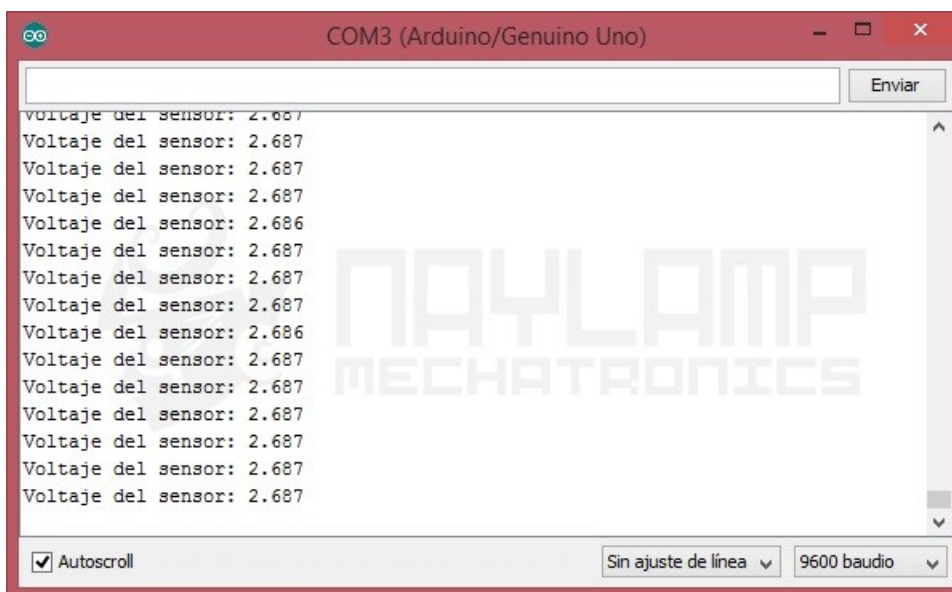
    return(voltage);
}
```

El primer punto que debemos medir es para una corriente de 0 amperios:



Esta es una de las constantes, en nuestro caso 2.527, que corresponde al punto P1(2.527,0)

Para hallar la segunda constante necesitamos una segunda medida, se recomienda que no sea cercano al valor de la primera medida, en nuestro caso usaremos una carga cuya corriente es superior a 1 amperio.



Como se observa el sensor tiene un voltaje de 2.687, y en el amperímetro medimos 1.155A, este sería nuestro segundo punto P2(2.687,1.155) , para calcular la sensibilidad simplemente calculamos la pendiente.

Teniendo estos dos valores la ecuación para calcular la corriente es:

$$\text{sensibilidad} = m = \frac{V_2 - V_1}{I_2 - I_1} = \frac{2.687 - 2.527}{1.155 - 0} = 0.139$$

Donde sensibilidad=0.139 V/A

Con estos nuevos valores debemos trabajar los ejemplos anteriores

Tener en cuenta que cada sensor tiene su propia característica



Otra forma puede ser en tomar varios puntos y al final aplicar regresión y hallara la ecuación de la recta.

## Ej.4 Vatímetro y Amperímetro de corriente alterna usando el módulo ACS712

En este ejemplo realizaremos mediciones de intensidad corriente alterna y con esto calcular la potencia.

Cuando realizamos una lectura del ACS712 obtenemos el valor de la corriente en ese instante, dicho valor estará oscilando a una frecuencia de 60Hz, los valores máximo en los que oscila son las corrientes picos, nuestro programa entonces deberá poder obtener los valores pico de la onda de corriente, Teniendo el valor de la corriente Pico podemos calcular la corriente eficaz o RMS y con esto la potencia.

A continuación mostramos el código de este ejemplo:

```
float Sensibilidad=0.139; //sensibilidad en V/A para nuestro sensor
float offset=0.100; // Equivale a la amplitud del ruido
void setup() {

    Serial.begin(9600);
}

void loop() {

    float Ip=get_corriente();//obtenemos la corriente pico
    float Irms=Ip*0.707; //Intensidad RMS = Ipico/(2^1/2)
    float P=Irms*220.0; // P=IV watts

    Serial.print("Ip: ");
    Serial.print(Ip,3);
    Serial.print("A , Irms: ");
    Serial.print(Irms,3);
    Serial.print("A, Potencia: ");
    Serial.print(P,3);
    Serial.println("W");
    delay(500);
}

float get_corriente()
{
    float voltajeSensor;
    float corriente=0;
    long tiempo=millis();
    float Imax=0;
    float Imin=0;

    while(millis()-tiempo<500)//realizamos mediciones durante 0.5 segundos
```



```

{
    voltajeSensor = analogRead(A0) * (5.0 / 1023.0); //lectura del sensor

    corriente=0.9*corriente+0.1*((voltajeSensor-2.527)/Sensibilidad); //Ecuación para obtener la corriente

    if(corriente>Imax)Imax=corriente;

    if(corriente<Imin)Imin=corriente;

}

return(((Imax-Imin)/2)-offset);
}

```

El algoritmo consiste en realizar mediciones durante medio segundo, lo cual equivale a realizar medidas durante 30 ciclos (señal de 60Hz), durante este tiempo obtenemos las lecturas máximas y mínimas que corresponden a +Ip y -Ip, la corriente Ip será el promedio de estas. A este resultado le restamos la amplitud del ruido que está presente cuando la corriente es 0.

Notar que para disminuir un poco el ruido aplicamos un filtro pasa bajos, que es similar a realizar un promedio de 10 muestras.

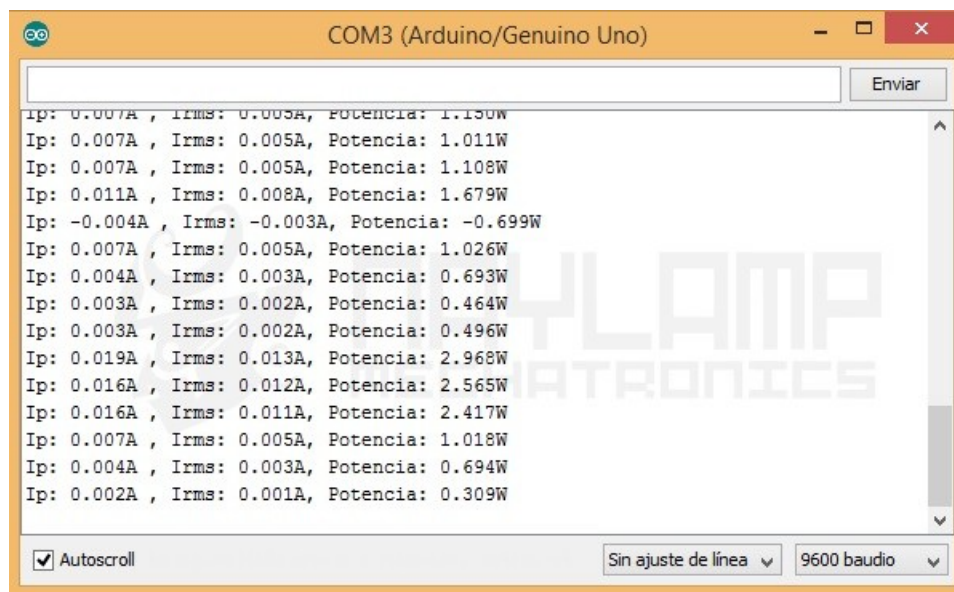
```

corriente=0.9*corriente+0.1*((voltajeSensor-2.527)/Sensibilidad); //Ecuación para obtener la corriente

```

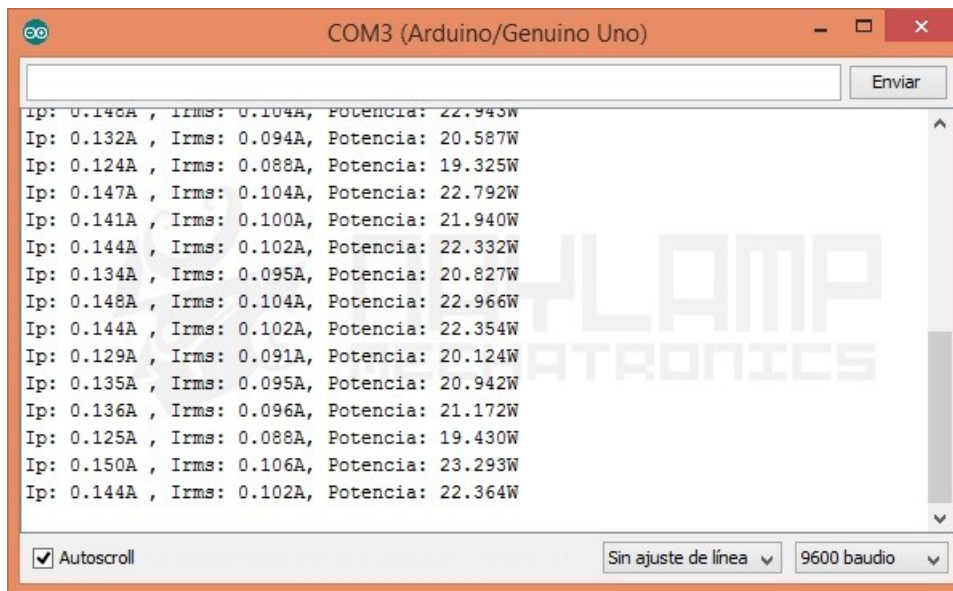
Con estos pesos amortiguamos un poco el ruido sin afectar la señal de 60Hz

A continuación se muestra el resultado cuando no tenemos ninguna carga conectada (0 amperios)



En nuestro caso la Corriente Ip es cercano a 0, puesto que nosotros ya hemos calculado el offset, en su caso deberán poner el offset en 0, y relizar la lectura sin carga, el valor del Ip será el valor del offset que deben rememplazar en su código.

Para ver si su amperímetro está bien calibrado deben contrastar con un amperímetro, vatímetro, o verificar con una carga conocida. Por ejemplo en continuación mostramos el resultado al trabajar con una carga de 20W (foco ahorrador)



Si su resultado no son los adecuados deberán de volver a calibrar su sensor, primero con corriente DC y encontrar la ecuación correcta para calura la corriente usando los ejemplos anteriores y posteriormente trabajarlo con AC.