ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

““DISEÑO DE UN DISPOSITIVO ELECTRÓNICO DE MEDICIÓN DE VOLTAJE Y CORRIENTE EN ALTERNO DE MANERA AUTONOMA PARA UN SISTEMA MODULAR FOTOVOLTAICO”

INFORME DE PROYECTO INTEGRADOR

Previo a la obtención del Título de:

INGENIERO EN AUTOMATIZACION INDUSTRIAL

DANIEL JOSÉ GALVEZ NAN

ABRAHAM ELIAS ZARI ANGEL

GUAYAQUIL – ECUADOR

AÑO: 2018

AGRADECIMIENTOS

Mi más sincero agradecimiento a Dios que me dio la fuerza necesaria en días difíciles donde divisaba un camino escabroso y me otorgo perseverancia para cumplir poco a poco mis objetivos, a mis padres y familia que tuvieron la paciencia para verme culminar mis estudios.

DEDICATORIA

El presente proyecto lo dedico a mi padre, a mi madre, pero en especial a mi abuela María Dolores Zari Guamán que, aunque ya no me acompaña en este mundo terrenal confió en mí y me dio sabios consejos que me sirvieron para no dejarme amilanar ante las adversidades de la vida.

_____
Abraham Elias Zari Angel

AGRADECIMIENTOS

Agradezco a Dios por bendecirme con excelentes padres, familia y amigos que me han guiado para culminar una de mis metas. A Lissette por su amistad incondicional. Son muchas personas que han formado parte de mi vida profesional a las que me encantaría agradecerles su amistad, consejos, ánimos, sin importan dónde estén quiero darles las gracias por formar parte de mí.

DEDICATORIA

A mis padres y hermana José Gálvez Procel, Alexandra Nan Palacios, Eunice Gálvez Nan, familia, maestros y amigos que han sido el pilar fundamental de todo lo que soy, en toda mi educación académica, por su incondicional apoyo ante toda circunstancias que se me han presentado.

_____
Daniel José Gálvez Nan

TRIBUNAL DE EVALUACIÓN

M.Sc. Franklin Kuonquí Gaínza	**PhD. Wilton Agila**
PROFESOR DE MATERIA INTEGRADORA	TUTOR ACADÉMICO

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

Daniel José Gálvez Nan

Abraham Elias Zari Angel

RESUMEN

Dado que el Centro de Investigación, Desarrollo e Innovación en Sistemas Computacionales “CIDIS” requiere un equipo de medición para conocer el consumo de la carga de sus instalaciones, se vio en la necesidad de adquirir uno. Sin embargo, la falta de disponibilidad en las tiendas tecnológicas locales conllevó a que se diseñe un dispositivo electrónico de bajo costo capaz de cumplir con las necesidades determinadas.

Para este fin, se utilizaron los conocimientos de electrónica y microcontroladores para diseñar un circuito que acondicione la señal, en conjunto con una placa de desarrollo de software libre que hizo que el proyecto integrador sea mejorado continuamente.

Del diseño se obtuvo un prototipo con el cual se realizó las pruebas de medición a partir de las variables de voltaje y corriente alterna. Observando las formas de onda que son registradas cada 0.5 segundo y guardándolas en forma de registro histórico en una memoria micro SD para su posterior análisis. Adicionalmente se le incorporó un display para visualizar los valores censados. Como logro posterior se hizo un programa en la plataforma Matlab para usar los datos registrados y mostrarlos gráficamente obteniendo los kilovatios consumidos en un periodo determinado por el usuario.

ÍNDICE GENERAL

Contenido

ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
CIDIS	Centro de Investigación Desarrollo e Innovación en Sistemas Computacionales

SIMBOLOGÍA

Vac Voltaje en alterna

Iac Corriente en Alterna

Kw Kilovatio

SCT Sensor de Corriente tipo gancho

ÍNDICE DE FIGURAS

CAPÍTULO 1

1. DELIMITACIÓN DEL PROBLEMA.

1.1 Planteamiento del Problema

El Centro de Investigación, Desarrollo e Innovación Sistemas Computacionales (CIDIS), en la búsqueda de nuevas tecnologías para mejorar sus instalaciones físicas, desarrolló un sistema modular fotovoltaico que genera energía limpia y aliviará el consumo de la carga del edificio cuya alimentación proviene de la empresa eléctrica del Ecuador.

El problema es encontrar una forma de cuantificar las variables eléctricas en alterna consumidas en un periodo determinado esto se lo podría hacer midiendo con un multímetro o un amperímetro de gancho, pero para poder registrar los consumos se los debería anotar continuamente de forma manual sin olvidar la persona que tome dichas lecturas.

El tiempo perdido mientras se lee y se transcribe la información de forma manual hace que se pierda el registro de transientes eléctricos, los cuales por aparecer en instantes de tiempos muy cortos pasan desapercibidos, y solo serían detectables cuando son picos muy altos de energía y dañan los equipos con componentes electrónicos conectados a la red.

Además, al buscar en el mercado tecnológico un equipo electrónico que registre la demanda de la carga y almacene dichos datos solo existen equipos fabricados en el exterior y su disponibilidad en el país es casi nula debido a los costos de importación y el tiempo de desaduanización.

1.2 Objetivos

1.2.1 Objetivo General

Diseñar un dispositivo electrónico que mida y registre las variables eléctricas de corriente alterna que se integre a un sistema modular fotovoltaico, visualizando sus formas de onda y almacenando registros para tener una base de datos que permita ver la factibilidad de la Gestión Activa de la demanda para su posterior análisis.

1.2.2 Objetivos específicos

En el siguiente proyecto de materia integradora se perseguirán los siguientes objetivos específicos:

- Implementar el sistema electrónico de monitoreo para el acondicionamiento de las señales obtenidas en la medición de las variables de voltaje y corriente en alterno.
- Implementar una aplicación software para el sistema de monitoreo, para la visualización y registro de datos eléctricos.
- Integrar el dispositivo electrónico a la salida del módulo inversor del sistema fotovoltaico.

.

1.3 Justificación.

Debido a que es indispensable llevar un registro de consumo de las cargas en el edificio del CIDIS porque actualmente se carece de esa información, hay que tener un equipo para poder realizar dichas mediciones y nos entregue un registro histórico. Para esto el equipo que se necesita es un monitor de energía de Domicilios que son algo escasos en el país.

Basado a lo expuesto anteriormente, se procede a buscar el equipo en las tiendas eléctricas o electrónicas en la ciudad y el país, pero solo se venden bajo pedido, en consecuencia, se procedió a buscar en internet en el mercado de equipos electrónicos encontrando una gran variedad de equipos pero debido a los impuestos de importación y a la poca política de regulación e implementación de tecnología en el país los precios se vuelven demasiado elevados para el presupuesto planificado. Todo esto y el tiempo de espera para disponer de los equipos en el país, hacen que sea más viable diseñar un prototipo para implementarlo en el CIDIS.

En aquel momento se decidió diseñar un dispositivo electrónico que cumpla todos los requerimientos utilizando conocimientos de electrónica y microprocesadores que podrían permitir el desarrollo de un modelo para ser fabricado y testeado en las instalaciones del CIDIS.

Para estar al tanto del uso que se da a las cargas conectadas a través del transcurso del día o la jornada laboral, es necesario tener un dispositivo que recolecte la información de una forma eficaz y lo más certera posible.

1.4 Alcance del proyecto

En el presente proyecto se diseñará un dispositivo que permita registrar, graficar y presentar en una computadora de manera gráfica los datos adquiridos de las variaciones de la demanda de la carga en tiempo real, entregada por el sistema fotovoltaico, que se encuentra ubicado en las oficinas del CIDS, el cual operará de manera autónoma y a un costo más asequible.

Se implementará un circuito electrónico constituido de Opams, potenciómetros, un display LCD 2x16 y resistencias, para mostrar el correcto funcionamiento, la debida protección al circuito y el acondicionamiento de las señales provenientes de los sensores analógicos: invasivos (terminales para medir el voltaje) y no-invasivos (SCT0-30 para la corriente).

Las señales analógicas acondicionadas llegarán a una tarjeta Arduino Uno, utilizado como controlador para la interpretación de las señales analógicas, transformándolas en datos digitales cuyos valores serán visualizados para representar el voltaje y la corriente RMS, junto con la potencia instantánea en el display LCD 2x16.

Estos datos además serán guardados en una tarjeta micro SD por medio del Micro controlador Atmega 328 y comunicados a la computadora por cable USB A-B (cable USB para impresora) y grabados en formato .csv para su fácil transmisión. Estos datos van a poder ser visualizados por medio de la aplicación Arduino en tiempo real y en una hoja de cálculo al terminar las mediciones.

Para visualizar las gráficas de voltaje, corriente, potencia y energía se utiliza el programa Matlab importando la base de datos grabada previamente en la micro SD. Así, se obtendrá un circuito seguro, confiable y económico para el monitoreo de las variaciones de la demanda de la carga del sistema fotovoltaico en tiempo real y autónomo.

CAPÍTULO 2

2. METODOLOGIA DEL TRABAJO.

Para la realización del siguiente proyecto integrador, se utilizarán conceptos fundamentales de Electrónica los cuales unidos con programación de microprocesadores darán como resultado un dispositivo que permita llevar un registro de la energía consumida.

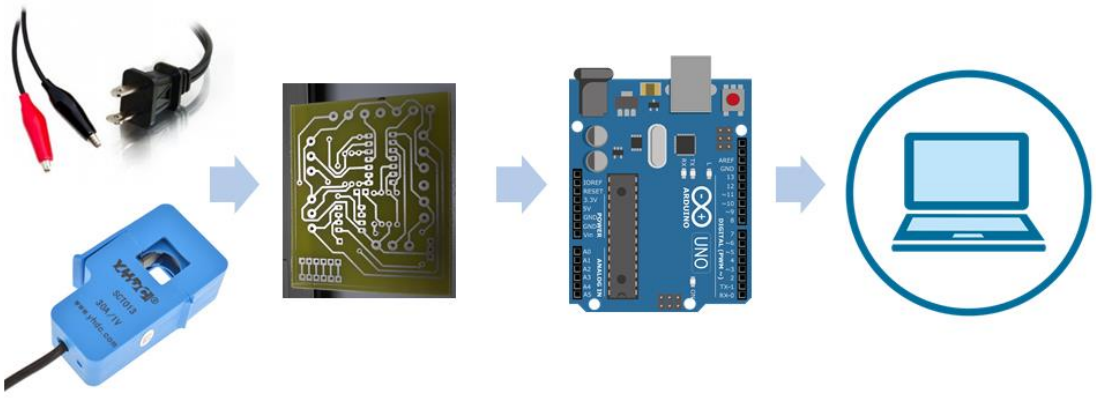

Figura 2. 1 : Esquema de funcionamiento

El proyecto consta de 4 etapas las cuales se muestran en el grafico 2.1, las cuales son la recepción de las señales en alterno, el acondicionamiento de estas señales en una placa electrónica, la transformación de las señales en la placa de desarrollo y el envío de los resultados para su visualización en un ordenador como se explica más detalladamente a continuación:

Se reciben las variables analógicas de voltaje a través de un cable de poder con terminales tipo pinza para acceder a las barras de alimentación, en este caso a la salida del inversor. En el caso de la entrada de la señal de corriente se usa un sensor tipo gancho amperimétrico SCT 013. Estas señales se introducen a la placa electrónica donde se acondicionan para que puedan ser leídas en la placa de desarrollo.

En el Arduino recibe los pulsos de la corriente y los convierte en señales digitales que son transmitidas mediante cable serial a un ordenador y a su vez es guardada en la micro SD con la ayuda de un módulo externo.

La información ingresada, es procesada por el software libre de Arduino instalada en el computador donde se observan las gráficas y datos en tiempo real.

2.1 Esquema Electrónico del circuito medidor de voltaje.

En el desarrollo del esquema se utiliza unos seguidores que provienen de los Opams LM324 y un circuito sumador que sirve para disminuir el voltaje del voltaje de entrada de la red que es 120v ac. Para su fácil comprensión se lo subdivide en etapas.

Figura 2. 2: Cable de sensor de Voltaje A.C.

La primera etapa es la reductora de voltaje que se muestra en la figura 2.3 que posee dos sub-etapas: el circuito reductor y el circuito seguidor. En el circuito reductor, la señal alterna es minimizada obteniéndose una señal de 0 a 7 Vpk. A continuación, en el circuito seguidor se busca reducir el nivel de corriente obtenido en la salida de la sub-etapa reductora.

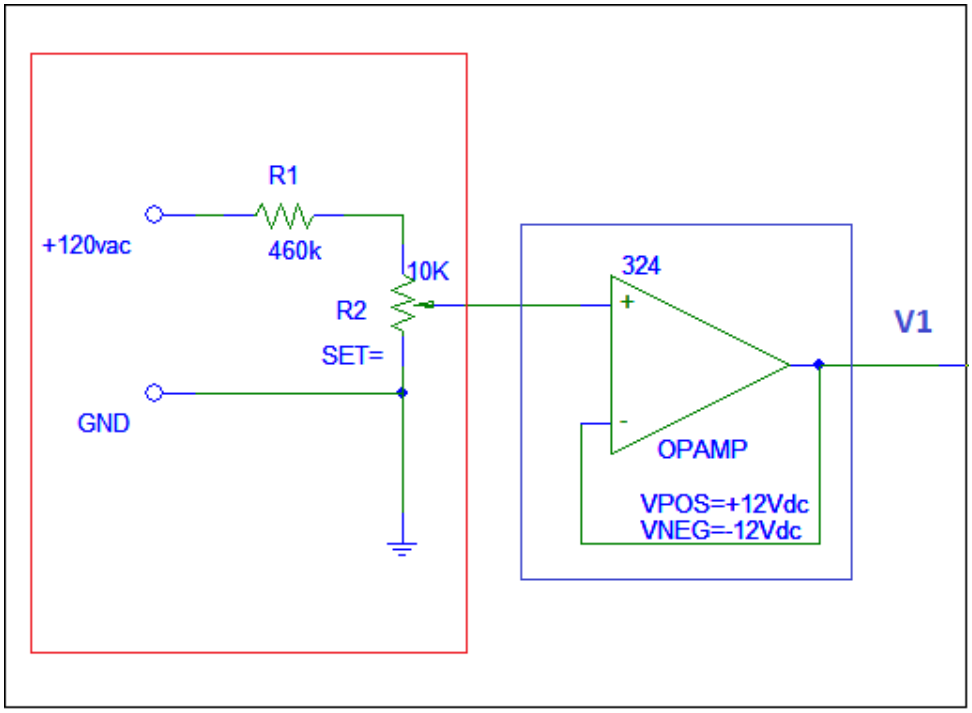

Figura 2. 3: Reductor de Voltaje

Luego, si el voltaje de entrada no satura el voltaje de polarización del seguidor, la salida será la misma de la entrante; caso contrario la salida se recortará al valor del voltaje de polarización, actuando como un recortador de señal. En el cuadro azul de la figura 2.3 se visualiza el proceso que realiza un seguidor y se la denomina V1.

Esta señal pasa por una configuración seguidor de voltaje para trabajar con la menor corriente posible ya que el Arduino One soporta máximo hasta 50mA de corriente.

En la siguiente etapa, se manipula el offset de la señal para tener únicamente valores positivos. Ingresa la señal V1 y mediante la electrónica mostrada se posiciona la señal en un rango de 0 a 5 Vdc como se observa en la figura 2.4. dando como salida una nueva señal llamada V2.

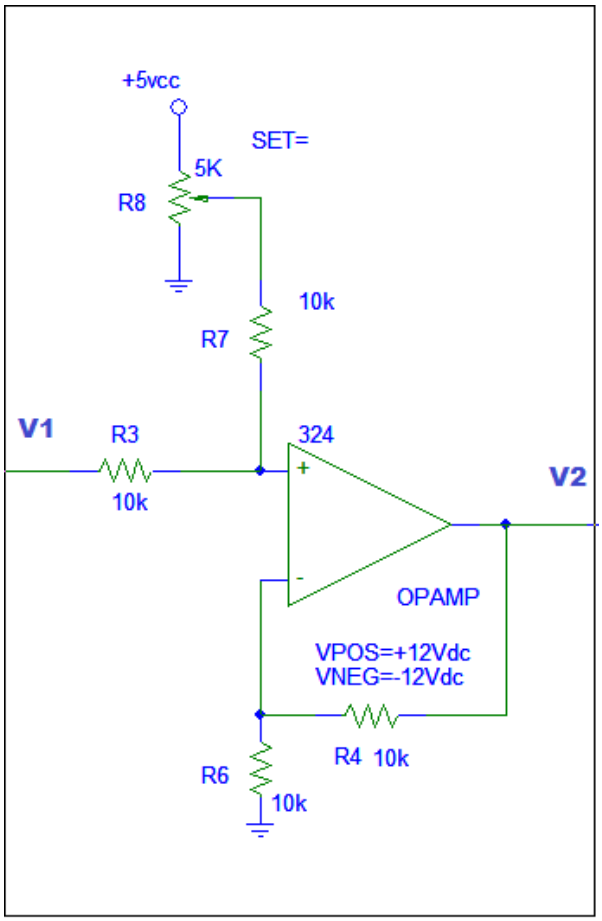

Figura 2. 4: Circuito offset

Para finalizar se conecta a un circuito aislador de la figura 2.5 para que el voltaje de la red No afecte al Arduino y también como medida de protección al Arduino. Este circuito también es conocido como un seguidor, pero en este caso se lo utiliza como aislador.

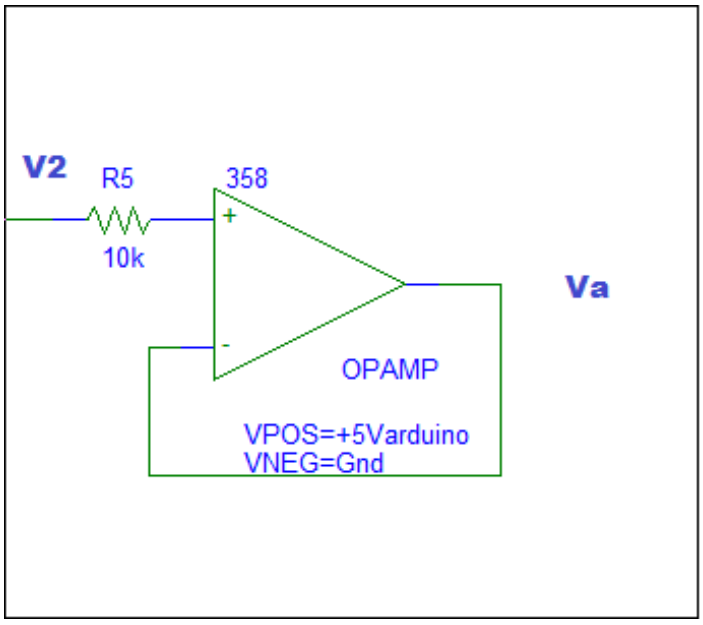

Figura 2. 5: Seguidor de voltaje

Uniendo estas 3 etapas da como resultado el circuito acondicionador de señal que se muestra en la figura 2.6.

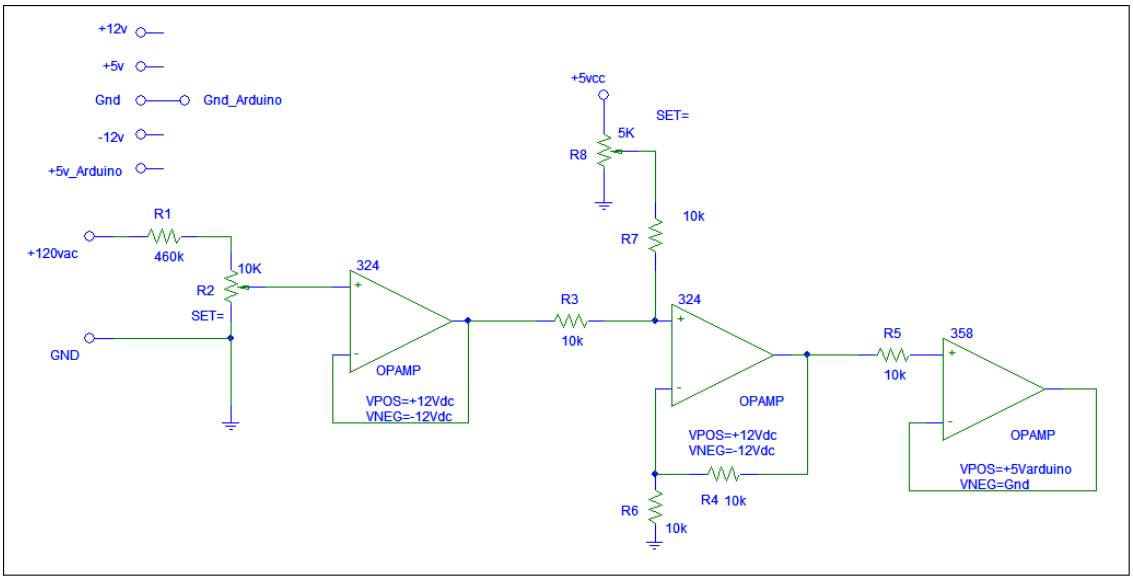

Figura 2. 6: Circuito de acondicionamiento de señal

2.2 Esquema Electrónico del circuito medidor de corriente.

Antes de comenzar se conecta el módulo medidor de corriente no intrusivo a una fase de la red, este transforma desde 0 a 30Amps a señales de voltaje de 0 a 1 volt.

Figura 2. 7: Sensor de corriente SCT013

Se empieza con la configuración de offset, para que suba la señal de 0 a 5 Vdc y así extrapolar los valores para que sean positivos en el rango de 0 a 5Vdc.

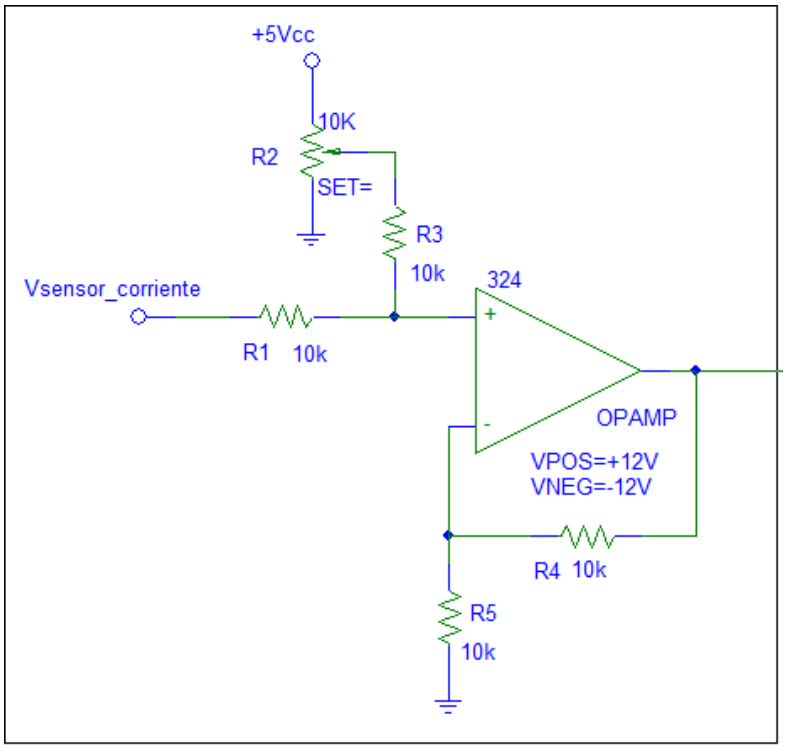

Figura 2. 8: Circuito offset

Luego pasa a un circuito amplificador de señal, se muestra en la figura 2.9 multiplica la señal de entra hasta 3 veces su valor.

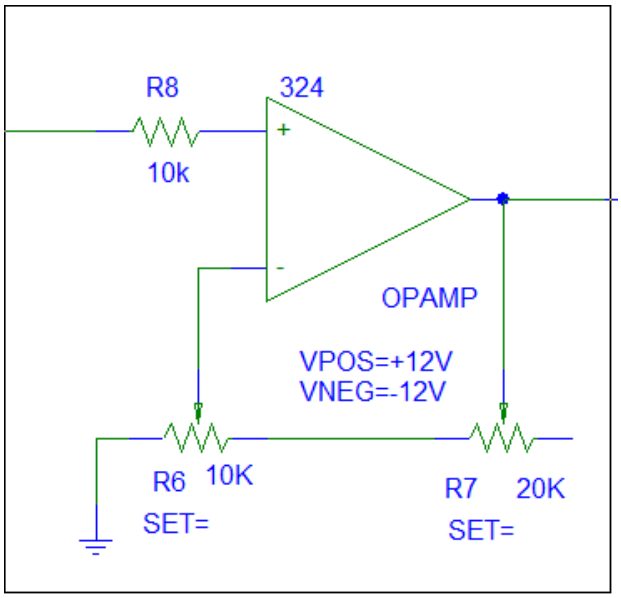

Figura 2. 9: Amplificador de Señal

Para finalizar se conecta a un circuito seguidor de voltaje mostrado en la figura 2.10 para aislar que el voltaje de la red No afecte al Arduino y también como medida de protección.

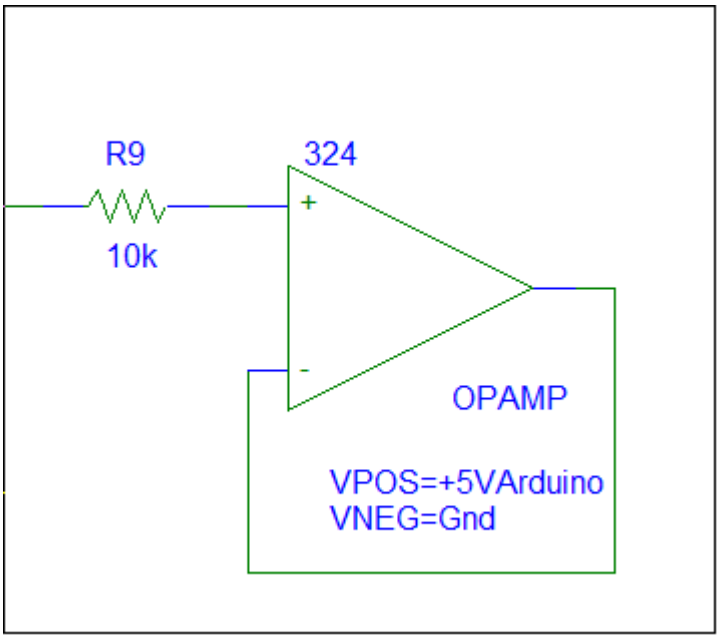

Figura 2. 10: Seguidor de voltaje

Para mejor apreciación visual en la figura 2.11 se observa cómo se unen las 3 etapas para conformar el circuito acondicionador de señal de corriente.

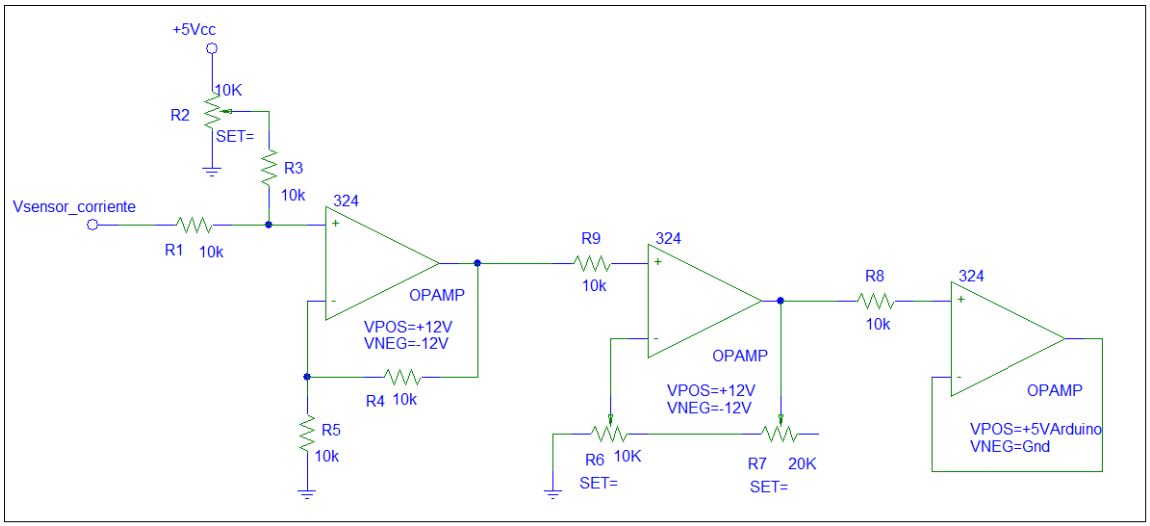

Figura 2. 11: Circuito medidor de corriente

2.3 Programa de Diseño de Placa PCB.

Con la finalidad de diseñar mejor el circuito y evitar posibles daños se procede a utilizar software de simulación de circuitos electrónicos en este caso se utiliza un simulador de circuitos electrónicos. Este programa también sirve para diseñar la placa en baquelita.

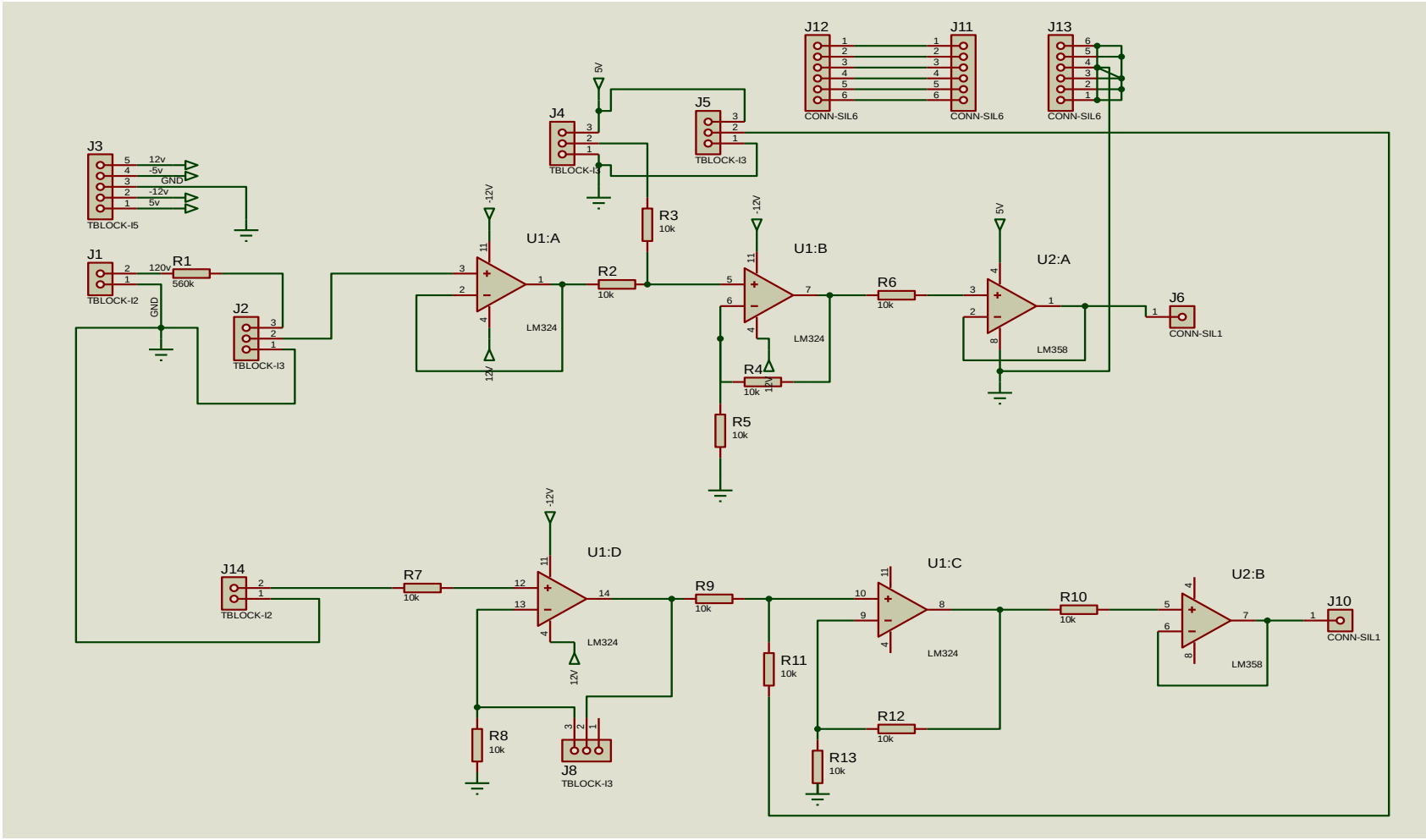

Figura 2. 12: Circuito General Simulado en Proteus

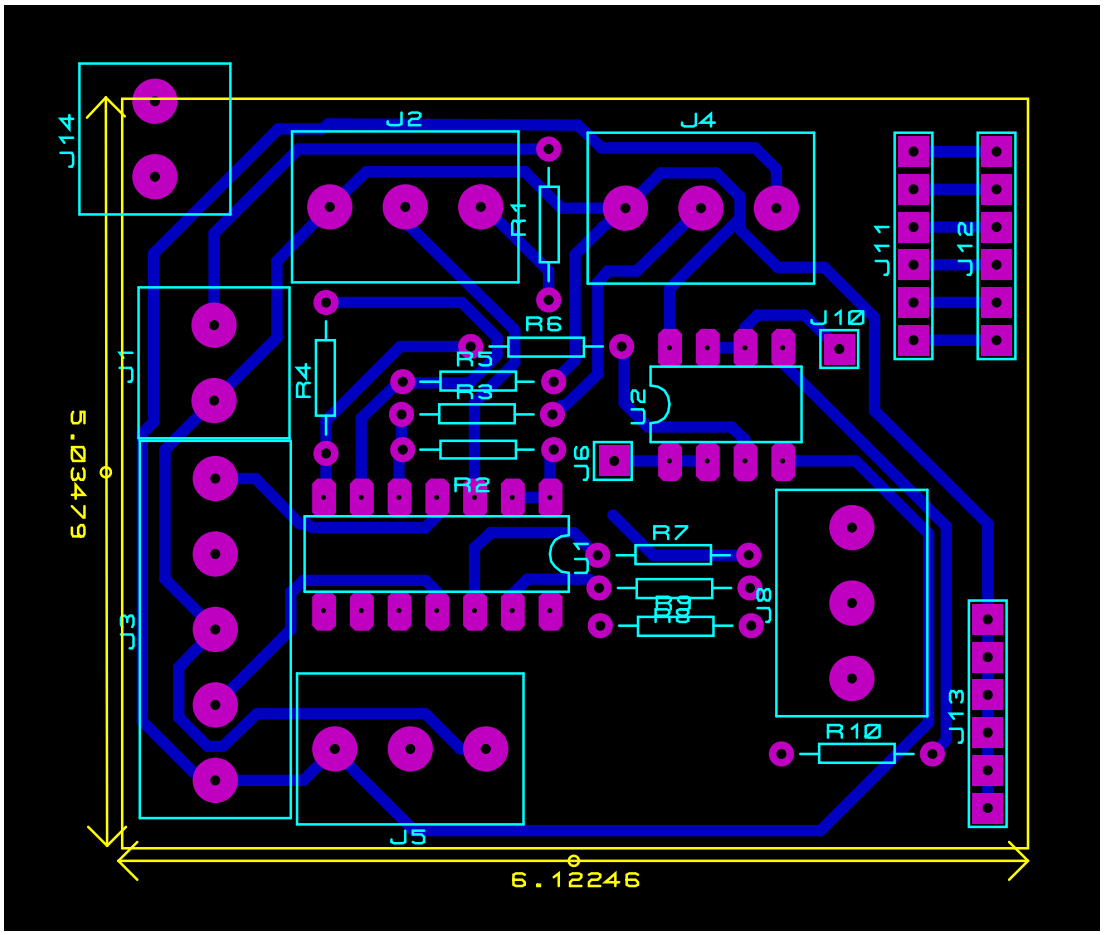

Figura 2. 13: Diseño PCB del prototipo electrónico

En la figura 2.13 muestra cómo se agrupan los elementos electrónicos para conformar una placa de reducido tamaño que ayuda en la fabricación de un dispositivo pequeño y robusto.

2.4 Programa para Transmisión de Datos

Para lograr transmitir los datos a un computador se usa el software libre de la compañía de tarjetas de desarrollo Arduino en su versión 1.8.1. En la siguiente figura 2.14 se visualiza parte del código de programación el cual permitirá simular las señales de los sensores de voltaje y corriente respectivamente.

Mediante la combinación de las señales de voltaje rms, corriente rms que ingresan a la tarjeta de desarrollo y con ayuda de la herramienta de programación se obtiene la potencia rms en wattios además estos Datos son transmitidos al computador

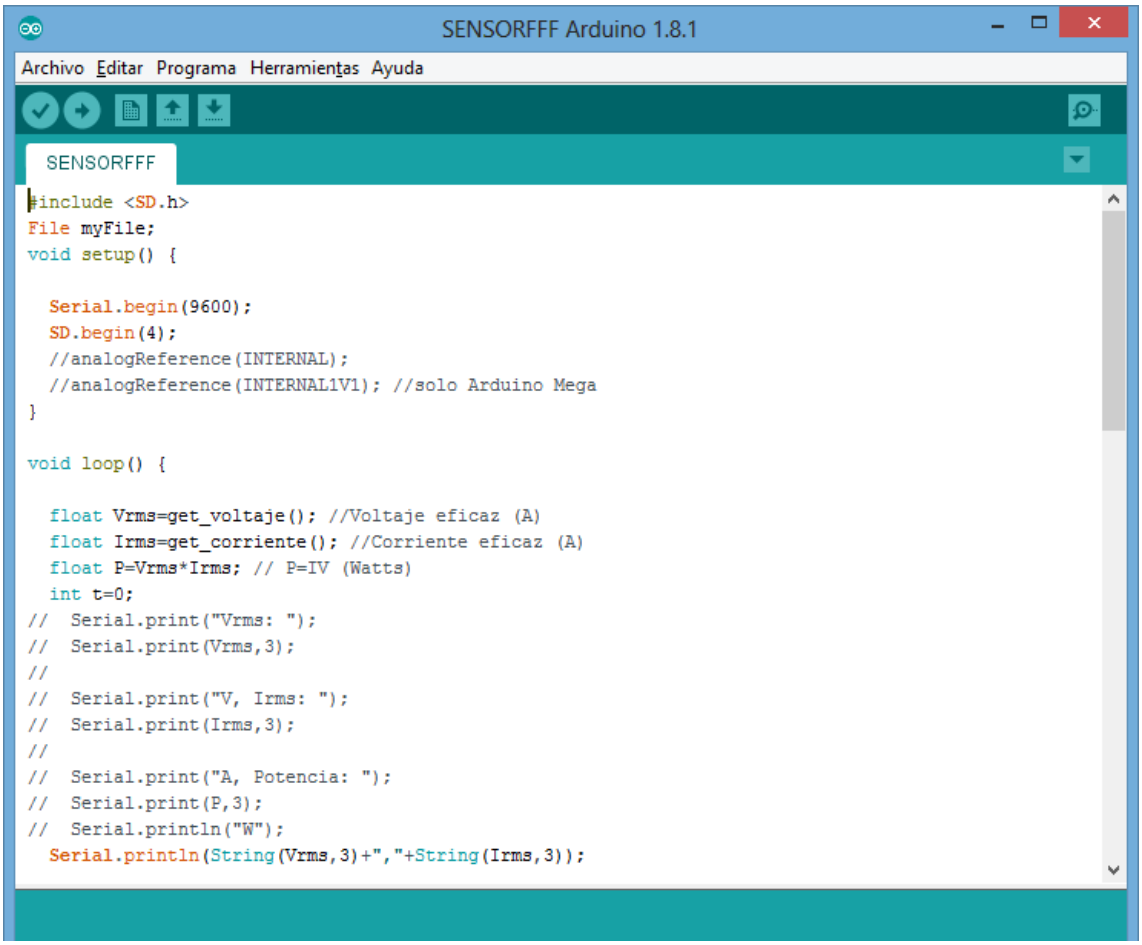

Figura 2. 14: Programa para los sensores de voltaje y corriente

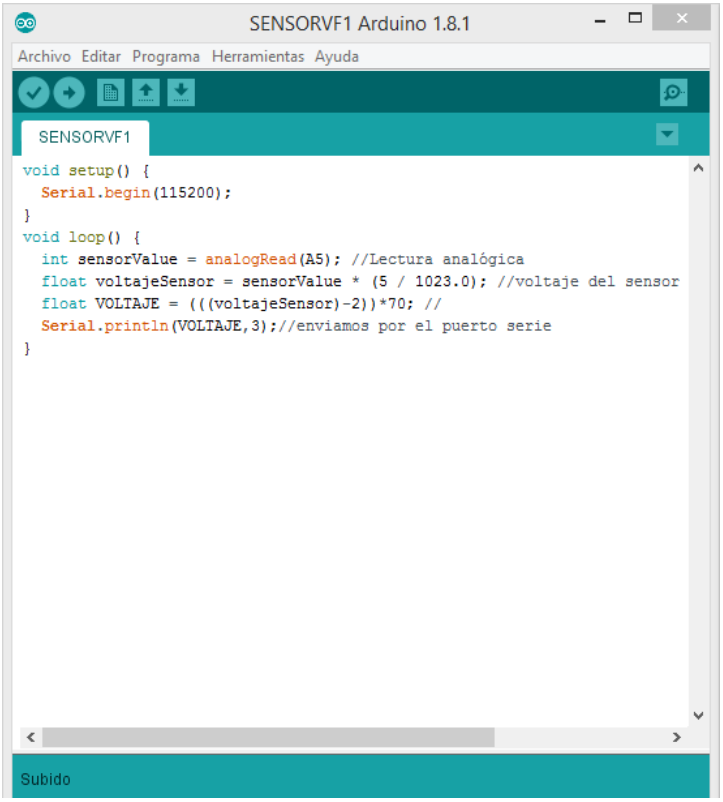

Figura 2. 15: Programa para sensor de Voltaje ac

Figura 2. 16 : Programa para sensor de Corriente ac

2.5 Programa para Almacenar datos en micro SD

Una vez recibido los datos, es fundamental guardarlos para su almacenamiento y posterior análisis en los programas EXCEL y Matlab los cuales se detallarán más adelante en el capítulo 4. En la figura 2.17 se muestra la pantalla del Software Arduino 1.81 donde contiene las líneas de programación que nos permita guardar los datos censados en una memoria micro SD.

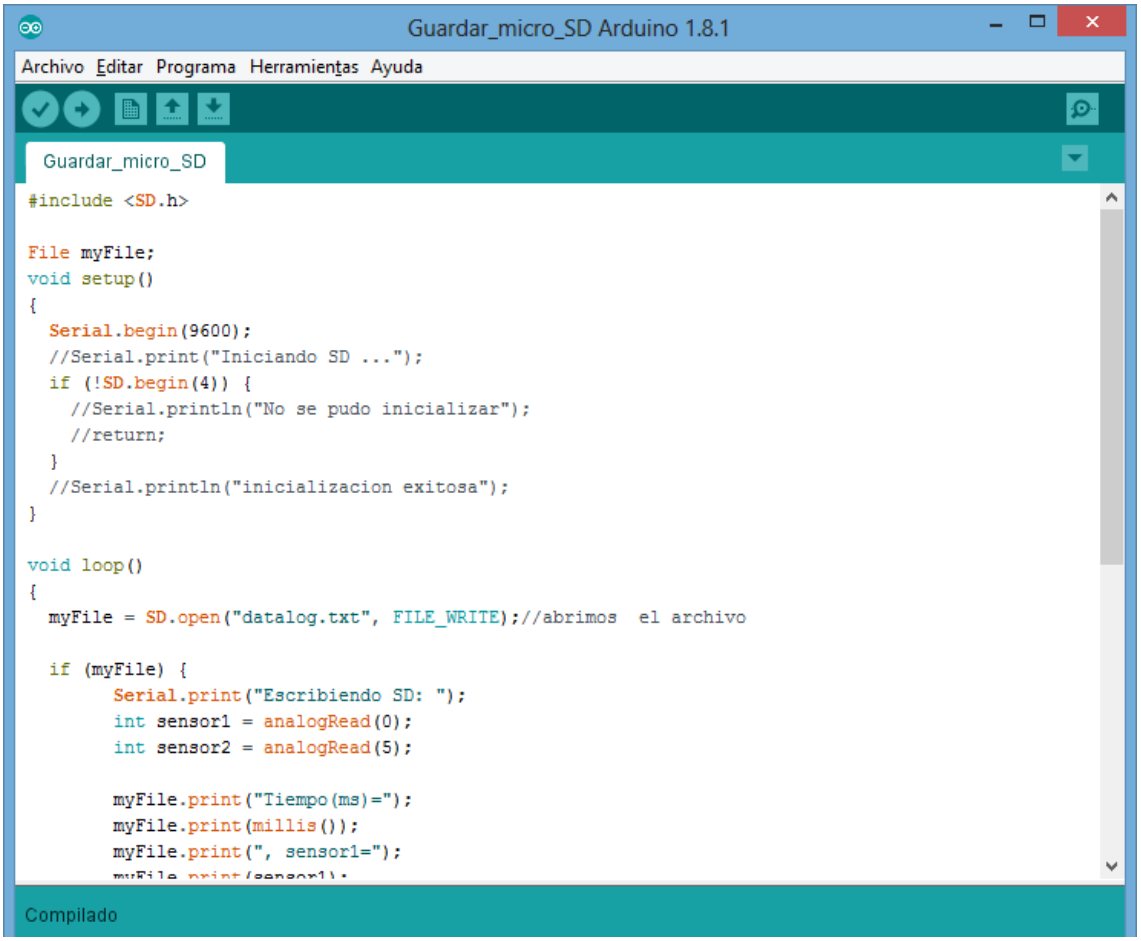

Figura 2. 17: Programa para el almacenamiento en micro SD

Para lograr almacenarlos es necesario adquirir el módulo de almacenamiento micro SD de la misma compañía de la tarjeta de desarrollo que son alimentados a 5V y cuya conexión se muestra en la figura 2.18

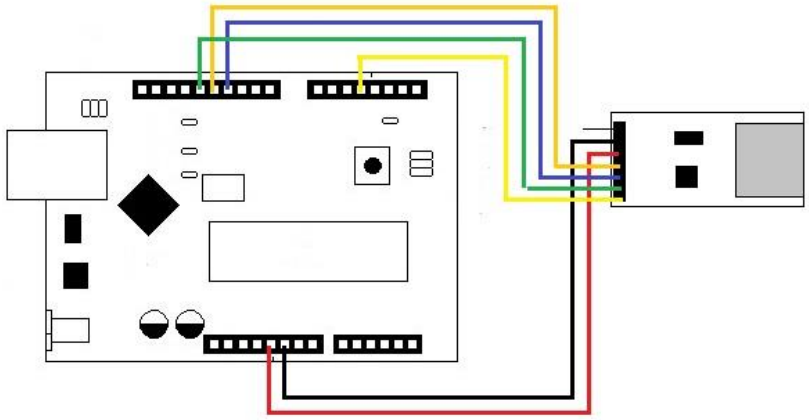

Figura 2. 18: Conexión a módulo micro SD.

CAPÍTULO 3

3. Resultados.

Se obtuvo como resultado que el dispositivo electrónico de medición pudo recopilar los datos al mismo tiempo que los monitoreaba. Creando una base de datos ordenada en carpetas y subcarpetas que nos indican el año, mes y día de lectura.

3.1 Curvas de medición de parámetros eléctricos en AC.

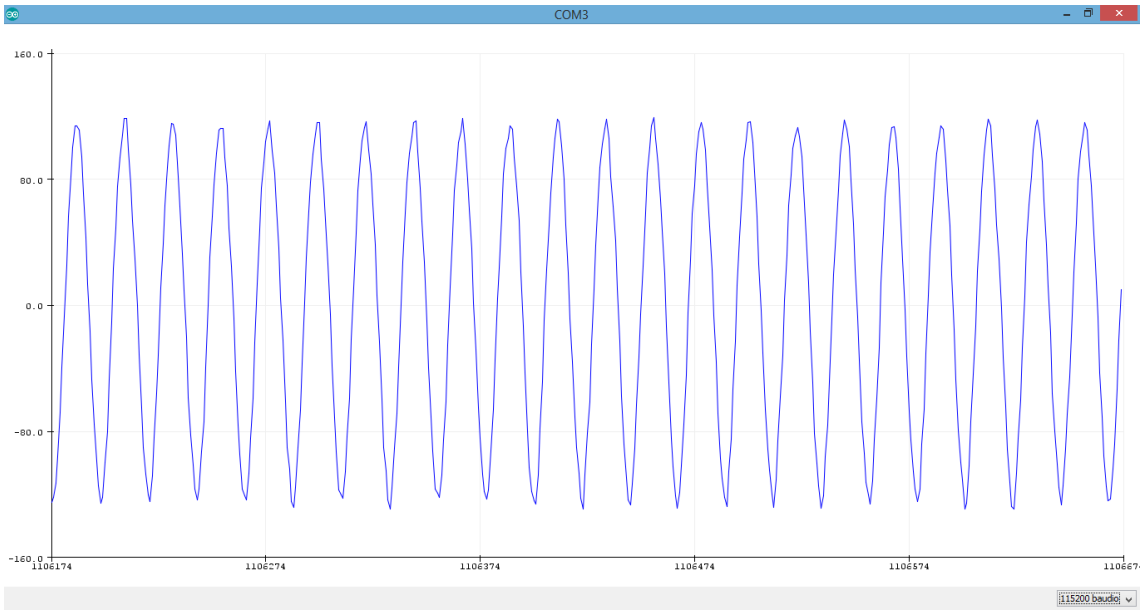

Figura 3. 1: Curva Voltaje en AC monitoreada

En la figura 3.1 se observar la forma y el patrón de oscilación del voltaje en AC el cual por limitaciones físicas de la placa de desarrollo que tiene una limitante, la cual es que no puede leer valores negativos. Por eso se agrega una resistencia variable para poder mover el offset de la curva y delimitarla en valores de 0 a 120 v.

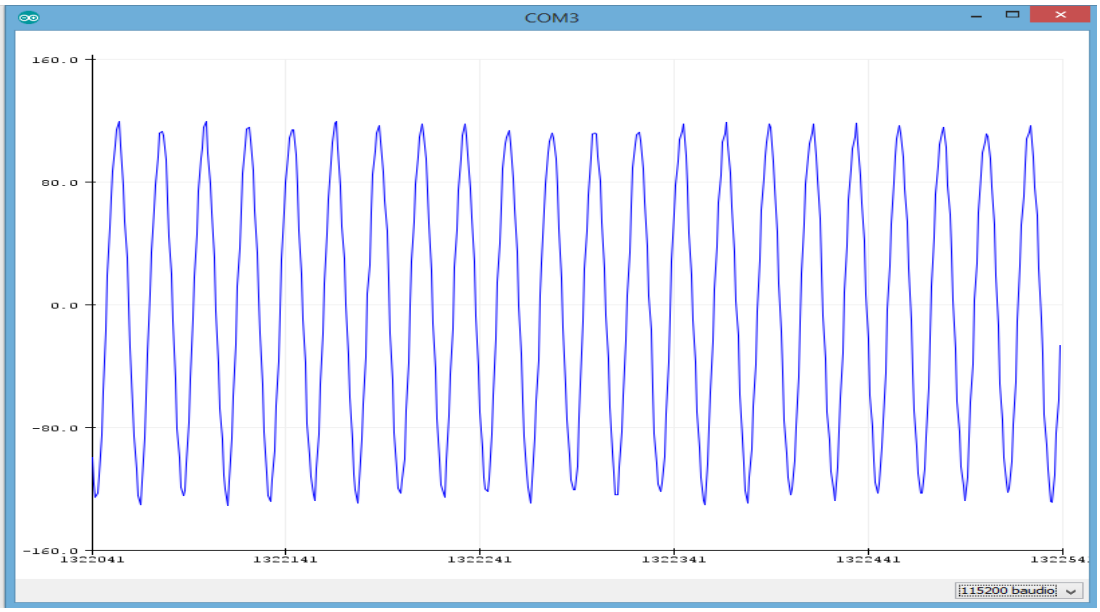

Figura 3. 2: Curva de Voltaje AC con programación de Arduino.

En la programación de Arduino, la función sensorValue es la encargada de receptar la lectura de voltaje alterna que proviene de la entrada analógica A5 y mediante una sencilla operación matemática segmenta los valores de 1023 en los valores que aproximen un valor máximo de 120 vac con un error +/- 10vac, debido a que la corriente alterna siempre es variable y los valores que entrega el suministro eléctrico llegan hasta los 130 vac.

Dado a que nuestro equipo será utilizado para realizar lecturas a la salida del inversor del módulo fotovoltaico, los valores que entreguen del inversor van a ser más estables y aproximados a los 120vac con valores mínimos de errores debido a su configuración interna.

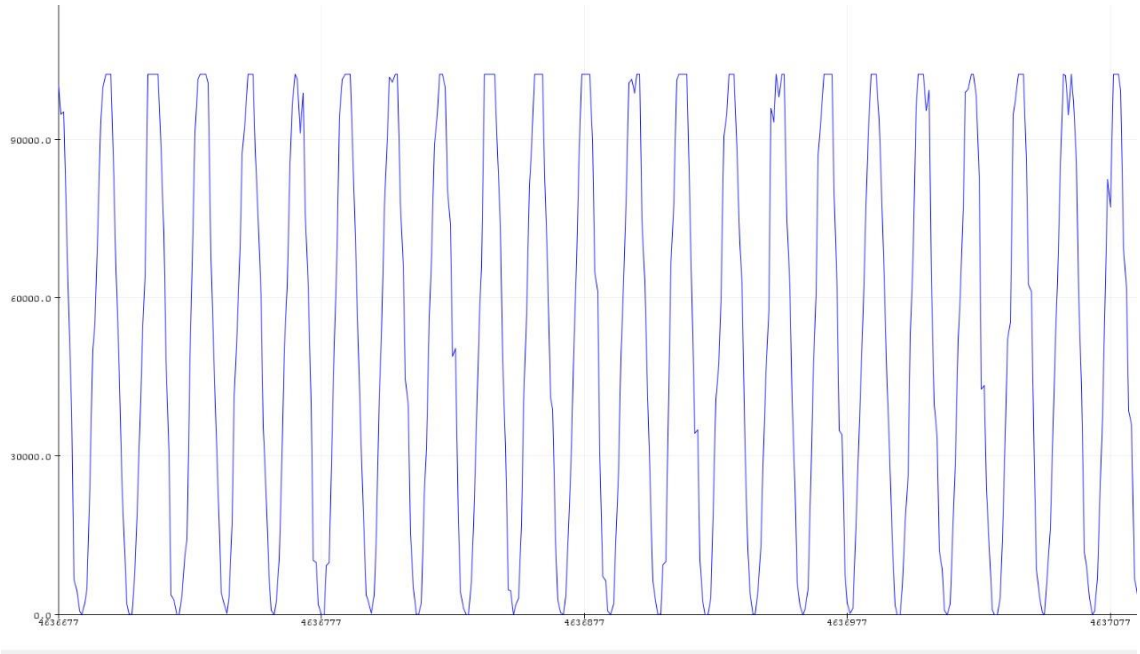

Figura 3. 3: Curva de Voltaje AC medida desde 0v

De la figura 3.3 se distingue la curva delimitada desde 0 a 120 vac como se observar dicha curva presenta variaciones en ciertos periodos de sus valores máximos los cuales normalmente solo se analizarían con un equipo de laboratorio como un osciloscopio

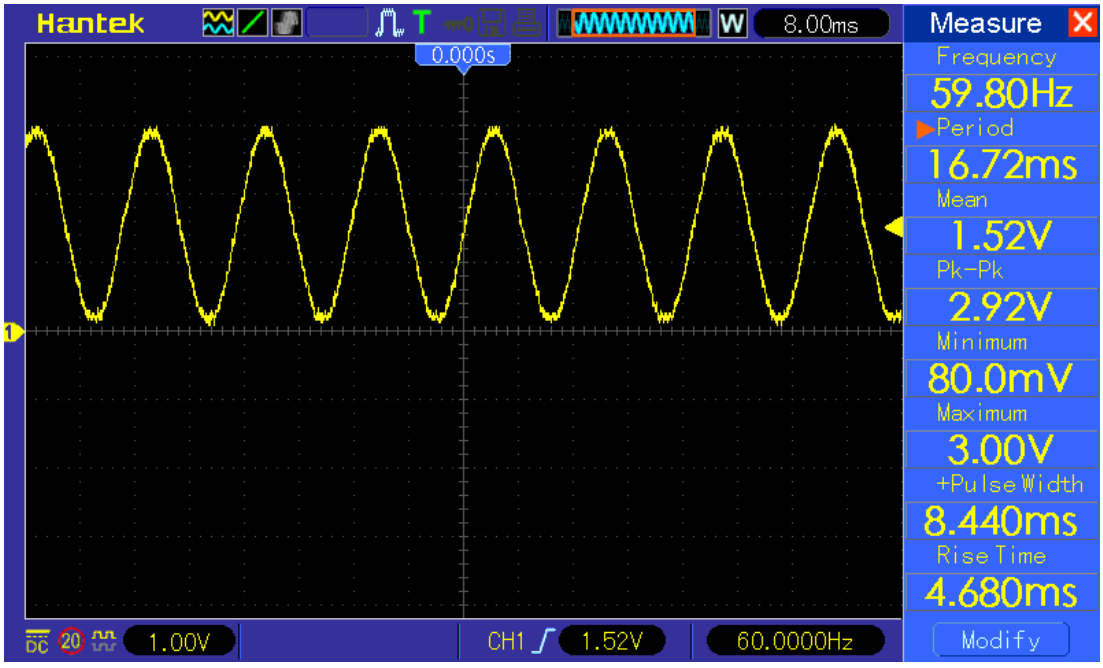

Figura 3. 4: Curva de Voltaje AC en Osciloscopio

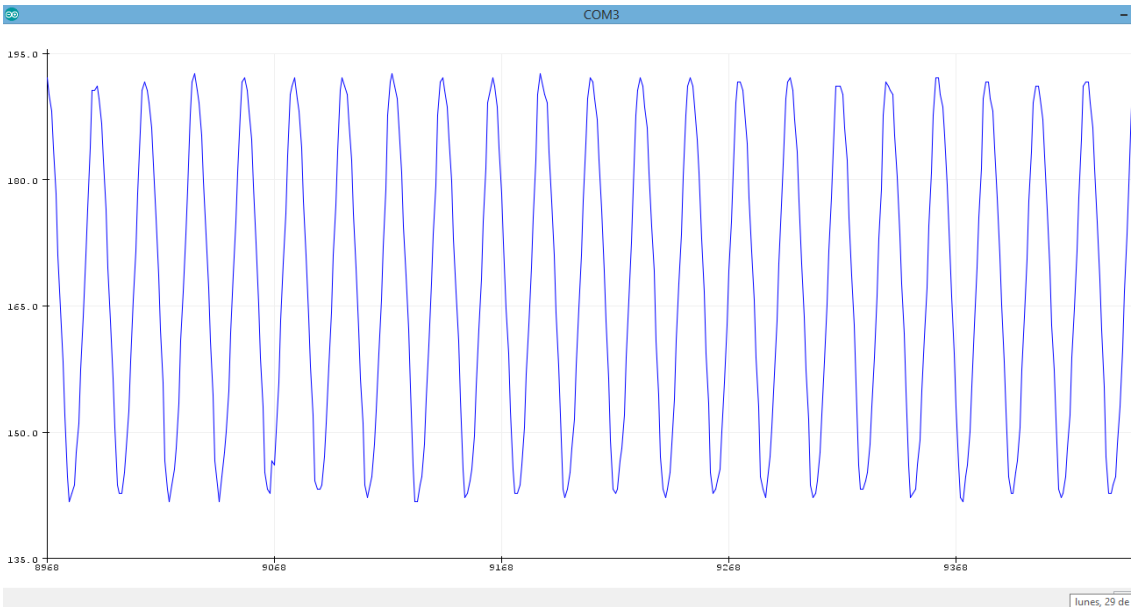

Figura 3. 5: Curva de Corriente AC monitoreada

La curva de corriente monitoreada a través del programa de software libre de Arduino nos refleja un consumo estable, este consumo puede variar en el transcurso del día dependiendo las necesidades de la edificación, a mayor carga la curva presentara valores picos altos. Más adelante se mostrará como el incremento de carga afecta los valores de la curva de corriente.

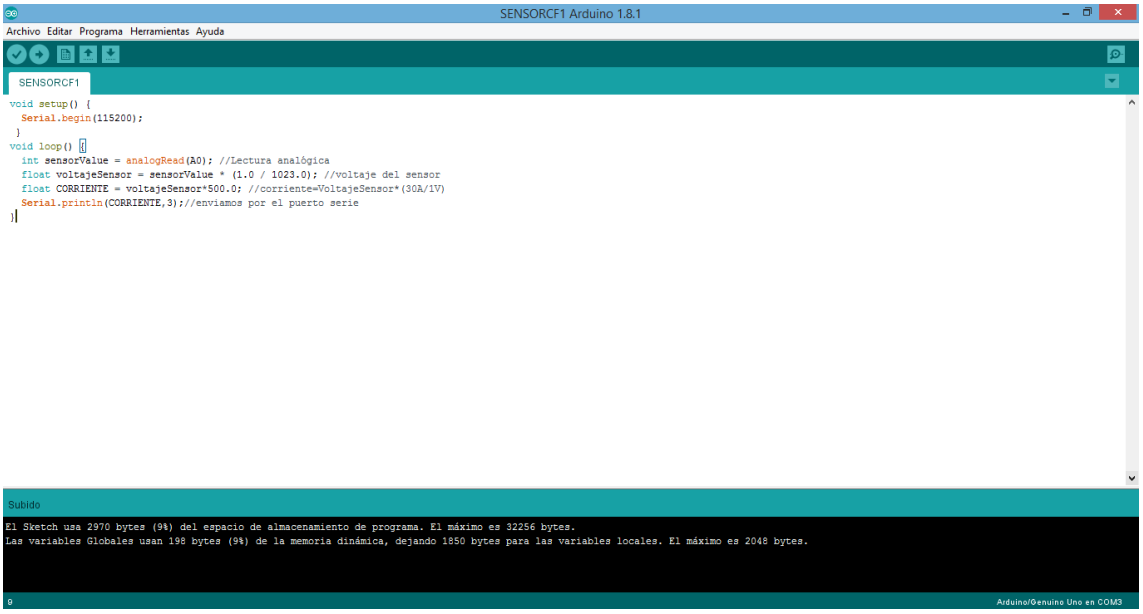

Figura 3. 6: Programación de Arduino para curva de corriente en AC

De la misma forma que se utiliza en la programación anterior se emplea la función sensorValue para receptar la lectura de corriente alterna que proviene de la entrada analógica A0, seguido de esto usamos la función float o punto flotante para que los valores receptados del sensor de corriente incrementen su precisión usando valores decimales de 32 bits.

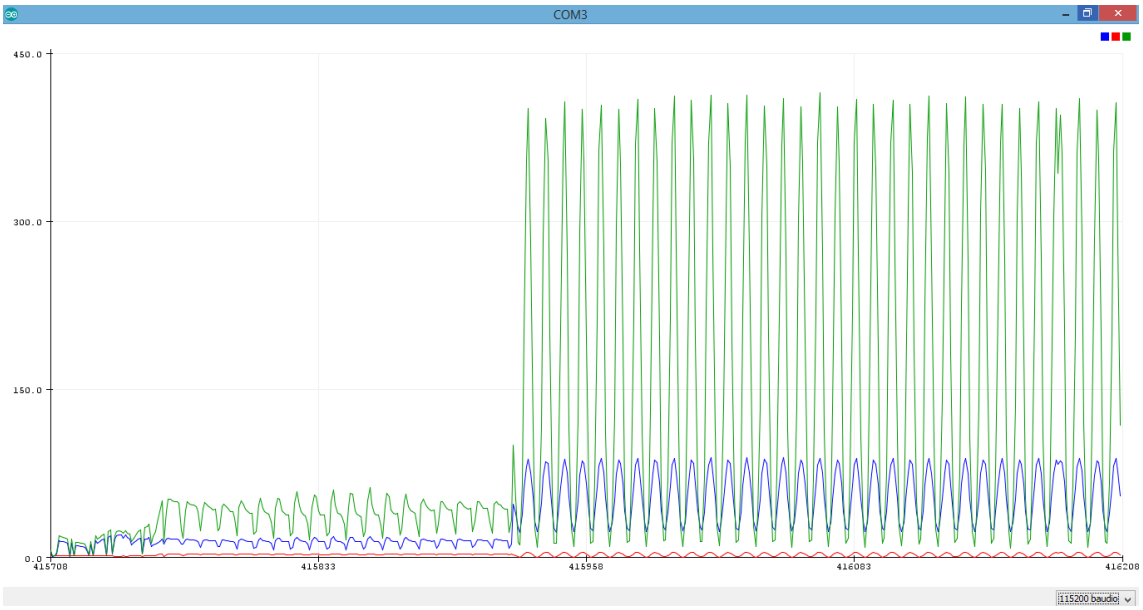

Figura 3. 7: Curvas de Voltaje, Corriente y Potencia en AC Energizando

Como se puede observar en la figura 3.7 se representa la curva de corriente ac en color rojo, la curva de voltaje ac en color azul y la curva de potencia en color verde. Los valores de la potencia se obtienen mediante programación multiplicando los valores de voltaje y corriente.

Después de un periodo largo de monitoreo sin carga, se procede a agregar la carga de un tablero de iluminación figura 3.8, con focos incandescentes de 120 watts de potencia lo que es reflejado en las curvas de monitoreo reflejando un transientes de encendido de 145 milisegundos.

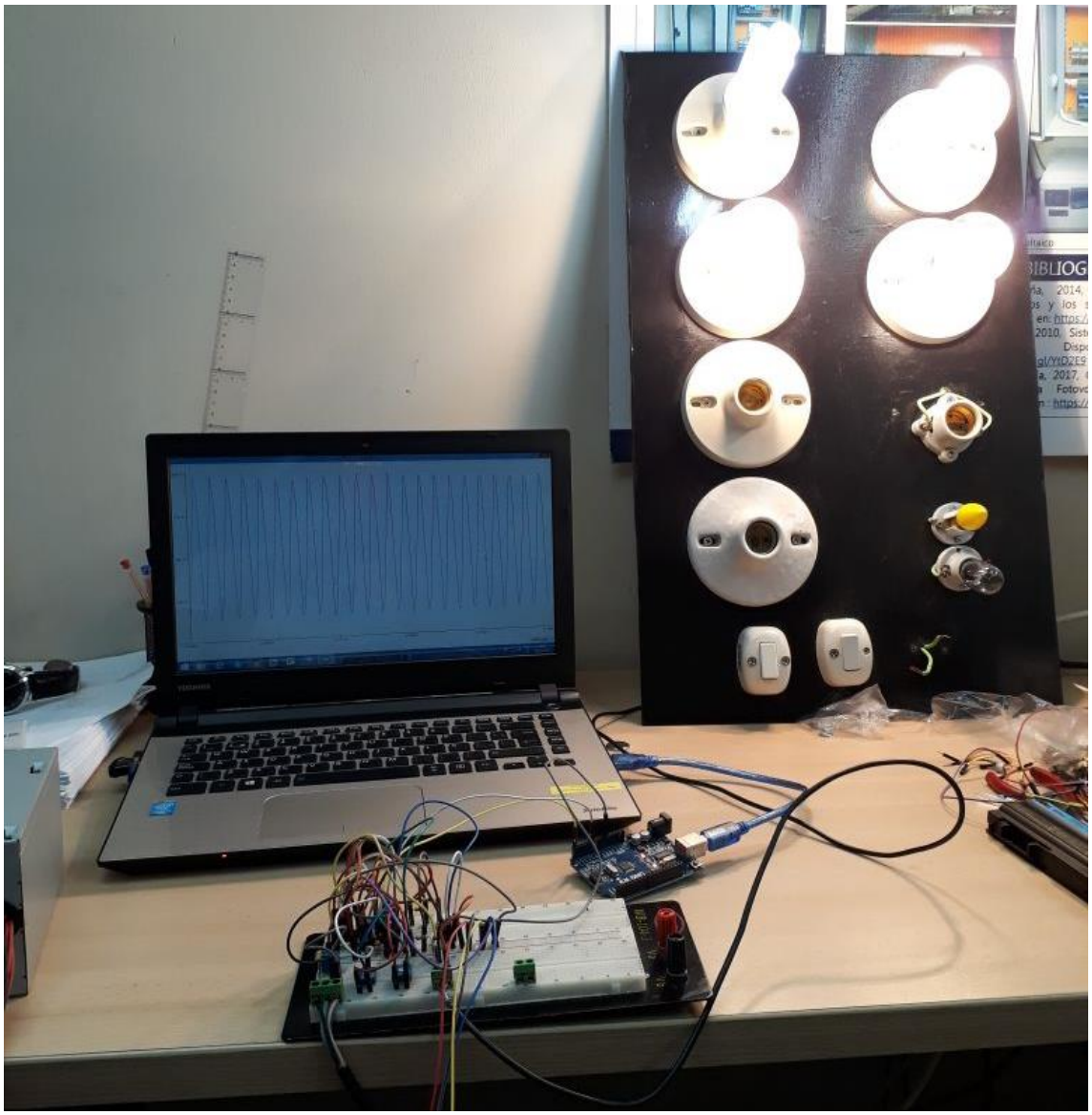

Figura 3. 8: Pruebas iniciales con tablero de Carga

Las cargas simuladas en el tablero representan el consumo de equipos eléctricos de bajo y mediano consumo. Para medir el consumo de equipos de mayor demanda será necesario conectar el prototipo a la salida del inversor o a un centro de distribución de carga en caso de utilizarlo para uso residencial.

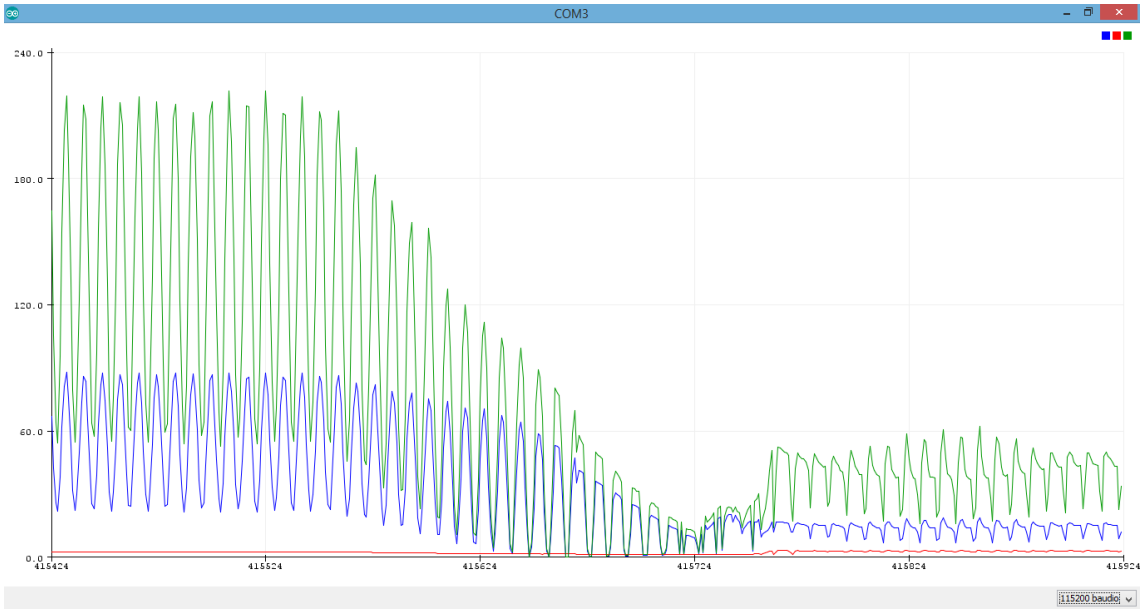

Figura 3. 9: Curvas de Voltaje, Corriente y Potencia en AC desenergizado

En la figura 3.9 se logra observar como la suspensión del consumo de la carga hace que la curvas decaigan sus valores hasta valores mínimos este efecto se realiza en unos 174 milisegundos aproximadamente. Mientras que en la figura 3.10 donde los valores son eficaces o RMS el tiempo de desnergizado fluctúa alrededor de unos 200 milisegundos gráficamente, para poder realizar cálculos más preciosos se recurre al archivo grabado en la micro SD.

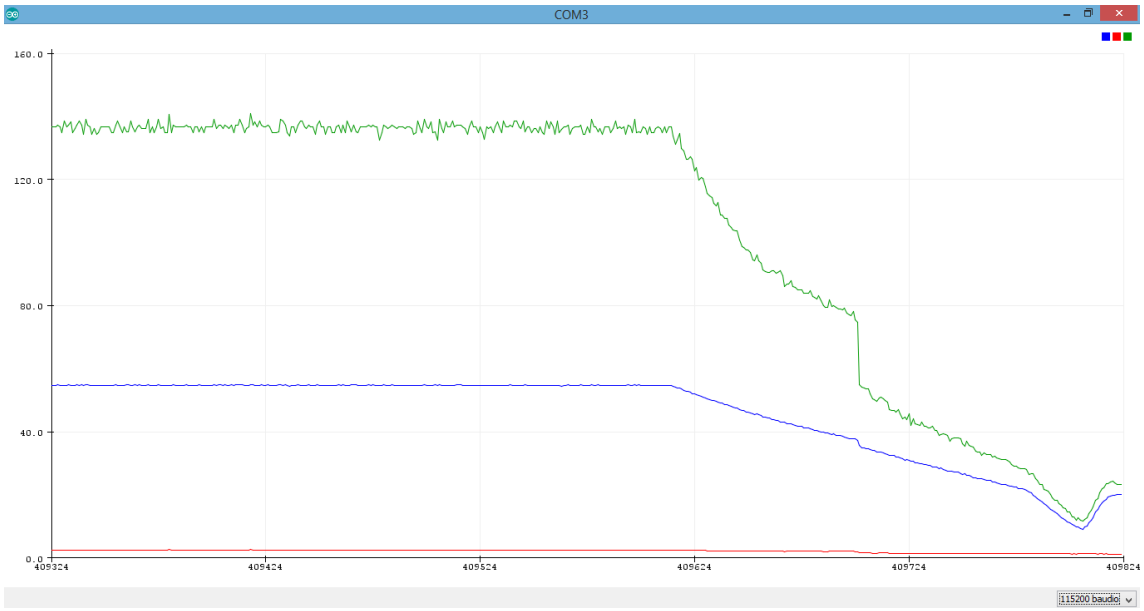

Figura 3. 10: Curvas RMS de Voltaje, Corriente y Potencia en AC desenergizado

La corriente Alterna, en su naturaleza, tiene instantes donde produce picos o valores instantáneos en pocos milisegundos los cuales podrían afectar a equipos electrónicos conectados a la red sino están protegidos en la figura 3.11 se captó el instante de este fenómeno q tan solo duro pocos milisegundos.

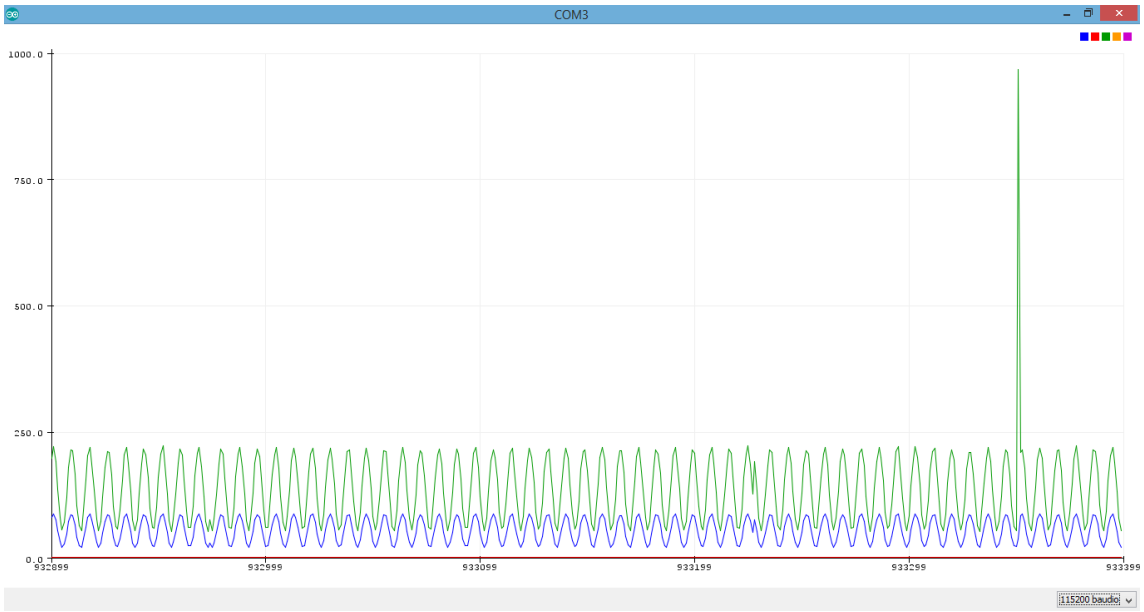

Figura 3. 11: Valores instantáneos durante la medición.

3.2 Respaldo de mediciones en micro SD.

La toma de mediciones no sería de mucha utilidad sino se registran dichos valores en un archivo histórico el cual permita acceder a la información de forma sectorizada y ordenada, para su posterior procesamiento manual o con ayuda de algún software realizar estudios a los datos obtenidos.

Para el grabado de los datos se programó el Arduino con ayuda de un módulo RTC o reloj para que los pueda almacenar los datos en carpetas y subcarpetas en orden descendente siendo la carpeta de mayor importancia la del año en curso 2018 pasando por 12 carpetas de meses y 30 a 31 carpetas de días según sea el caso.

Estas carpetas serán creadas en el instante que se realicen las mediciones guardando los datos en archivos con extensión .CSV dichos archivos tomaran el nombre de la hora en la cual empezó la medición de las variables eléctricas en alterna.

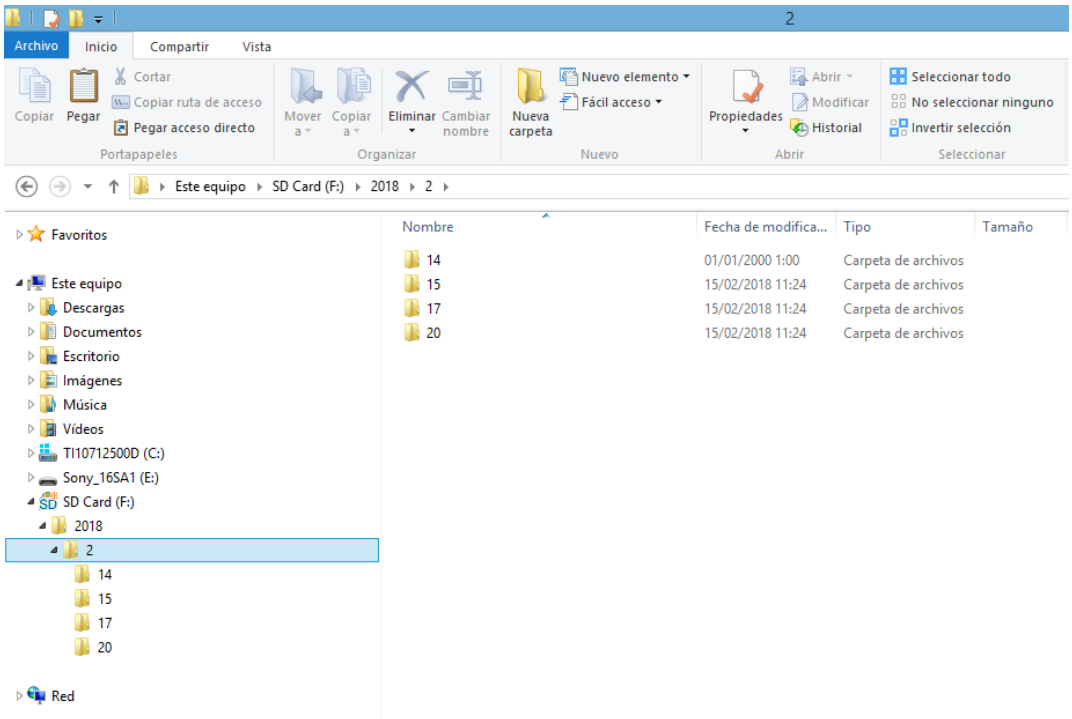

Figura 3. 12: Grabado de datos en micro SD

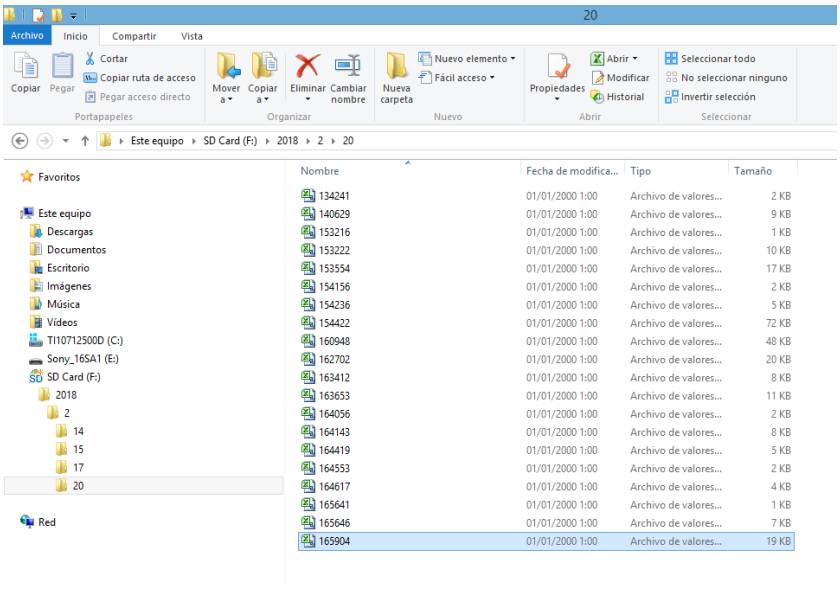

Figura 3. 13: Archivos generados en extensión .csv

3.3 Prototipo de placa electrónica.

Para poder realizar las pruebas se tuvo que crear un prototipo físico el cual permita realizar las mediciones, para esto se diseñó una placa lo más pequeña posible con borneras para la entrada de las señales de los sensores y salidas de señales hacia el Arduino.

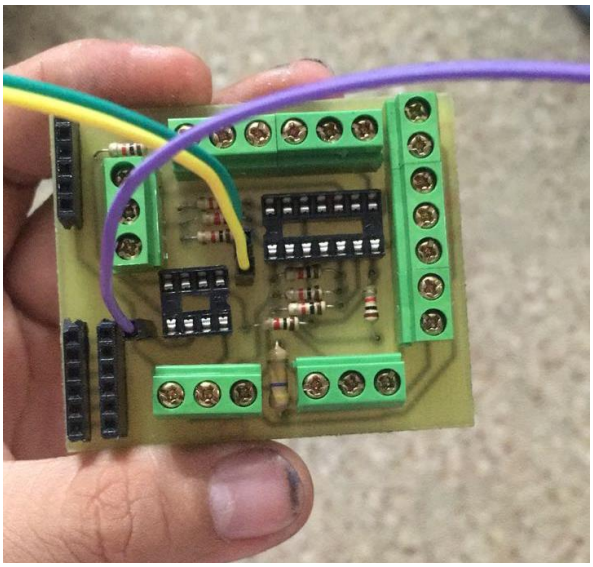

Figura 3. 14: Placa inicial

La fabricación del prototipo no presento mayor problema en la adquisición de elementos ya que todos los elementos electrónicos están disponibles en el mercado, a continuación, la figura 3.15 muestra los valores detallados de cada uno.

Archivo | Inicio | Insertar | Diseño de página | Fórmulas | Datos | Revisar

E22 | fx | =SUMA(E3:E21)

	A	B	C	D	E
1		Costo manufactura			
2					
3	Borneras de 3		4	0,15	0,60
4	Borneras de 2		6	0,10	0,60
5	Integrado 324		1	0,65	0,65
6	Integrado 358		1	0,50	0,50
7	Base socket 14 pines		1	0,15	0,15
8	Base socket 8 pines		1	0,10	0,10
9	Potenciometro 5k		2	0,20	0,40
10	Potenciometro 10k		1	0,20	0,20
11	Potenciometro 20k		1	0,20	0,20
12	Resistencia 560K		1	0,05	0,05
13	Resistencia 10K		12	0,05	0,60
14	Cable 120vac		1	1,00	1,00
15	Sensor corriente		1	10,00	10,00
16	Placa		1	5,10	5,10
17	Arduino uno		1	11,00	11,00
18	Modulo micro sd		1	3,50	3,50
19	Modulo reloj		1	3,50	3,50
20	Display 2x16		1	4,00	4,00
21	Socket espadines		1	0,10	0,10
22					42,25
23					

Figura 3. 15: Costos de fabricación

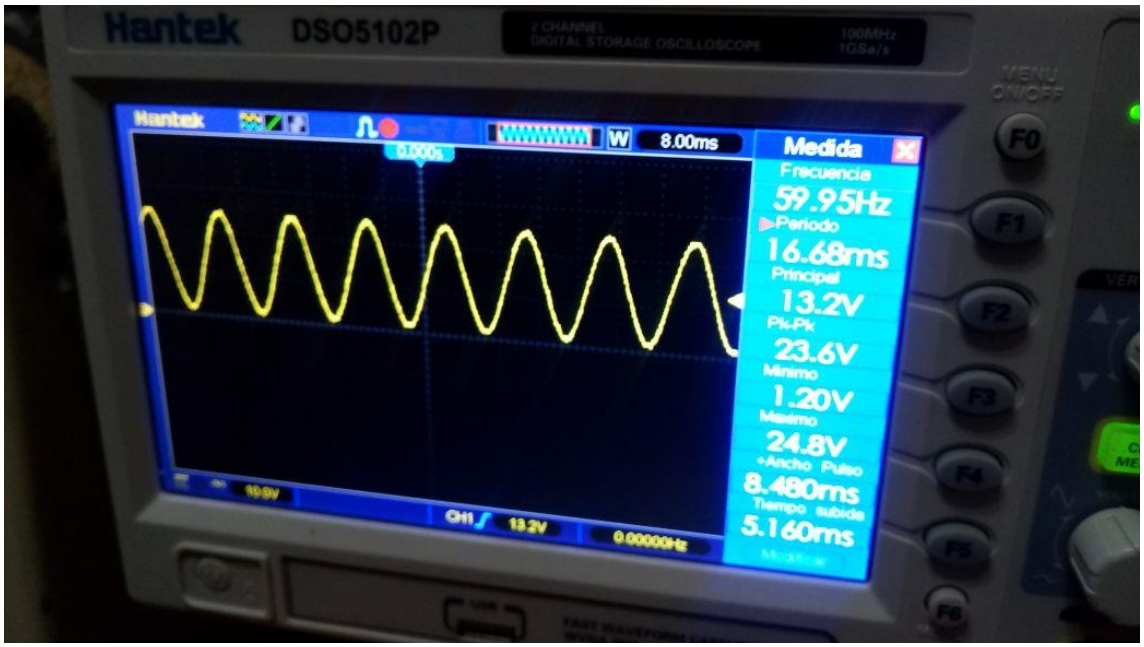

Figura 3. 16 : Señal del pin 1 del LM358

Señal del pin 1, que va a la entrada analógica A5 del Arduino, sin ruido.

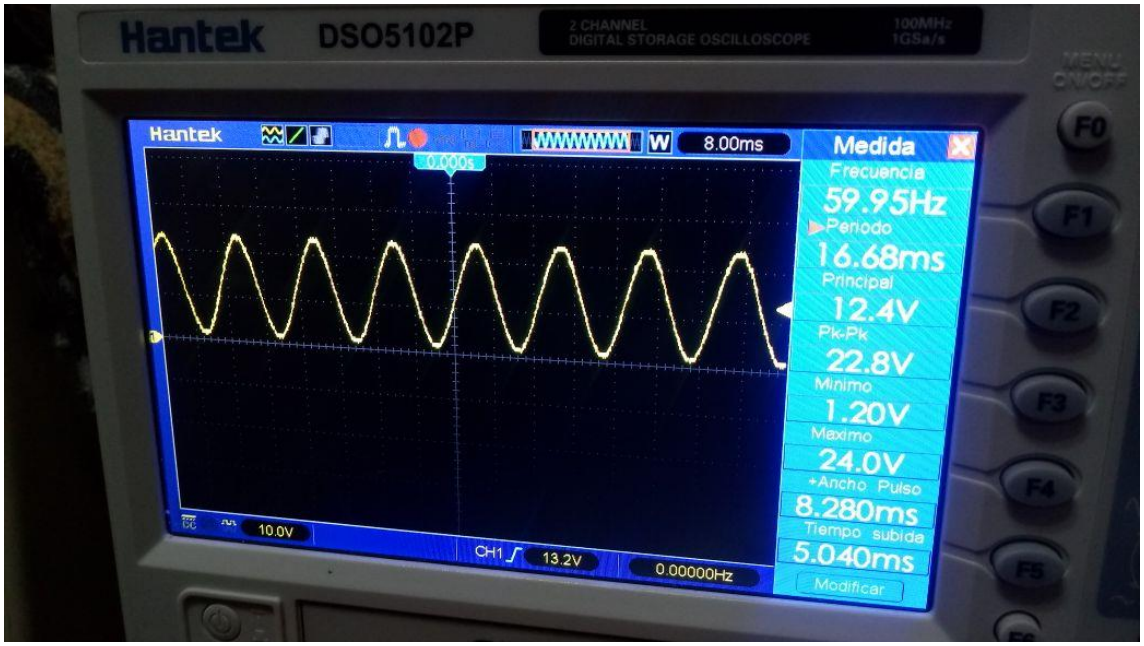

Figura 3. 17: Señal del pin 7 del LM324 y pin 3 del LM358

Es un seguidor voltaje, la corriente es aproximadamente cero entre estos nodos, y sin ruido.

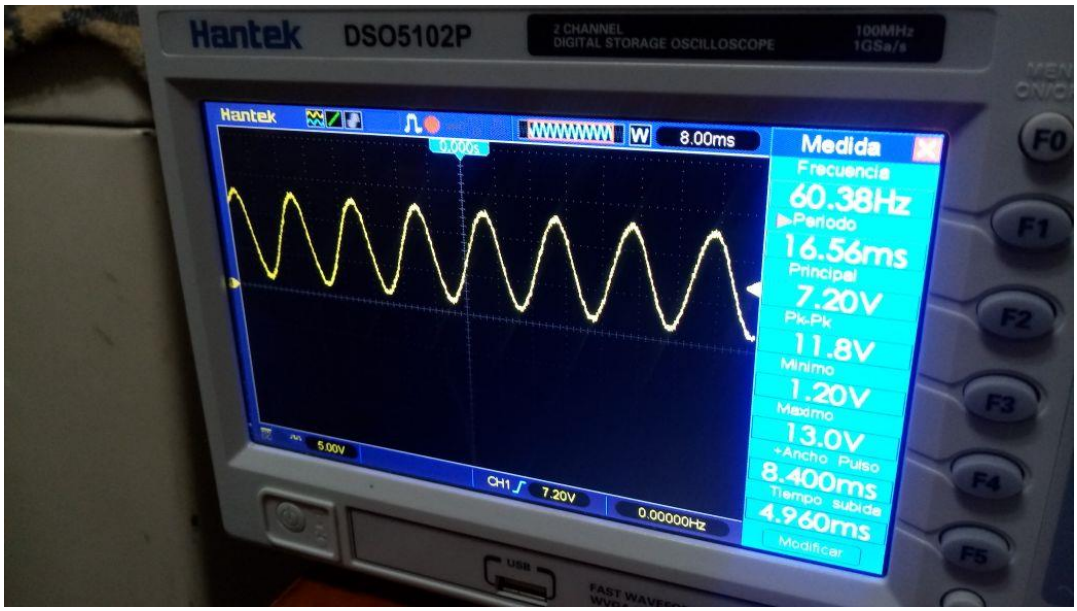

Figura 3. 18 : Señal del pin 5 del LM324

Señal de los 120Vac reducido de voltaje, tratada tanto con offset y con menos ruido.

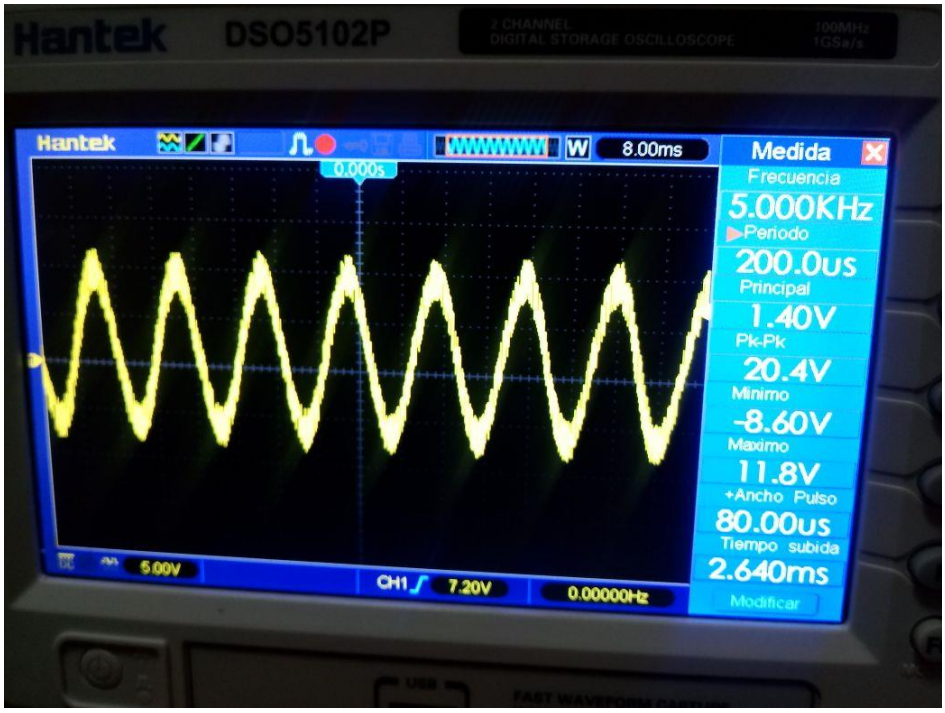

Figura 3. 19 Señal del pin 3 del LM324

Señal 120Vac reducida y con ruido.

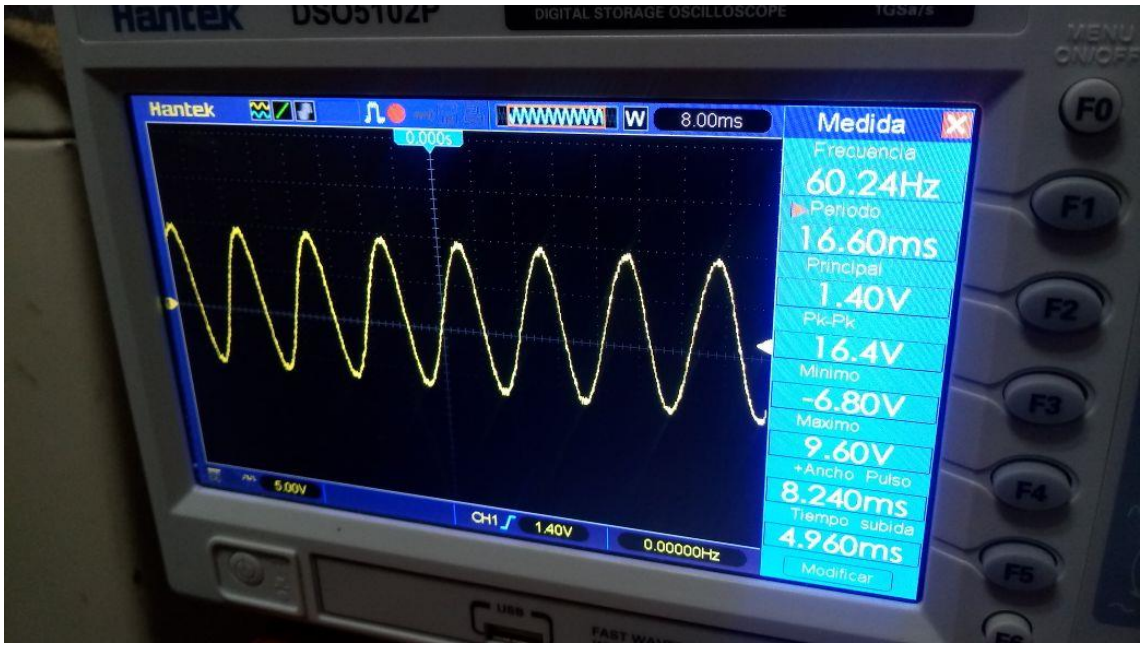

Figura 3. 20 : Señal del pin 1 del LM324

Señal de los 120Vac, reducida y con poco ruido.

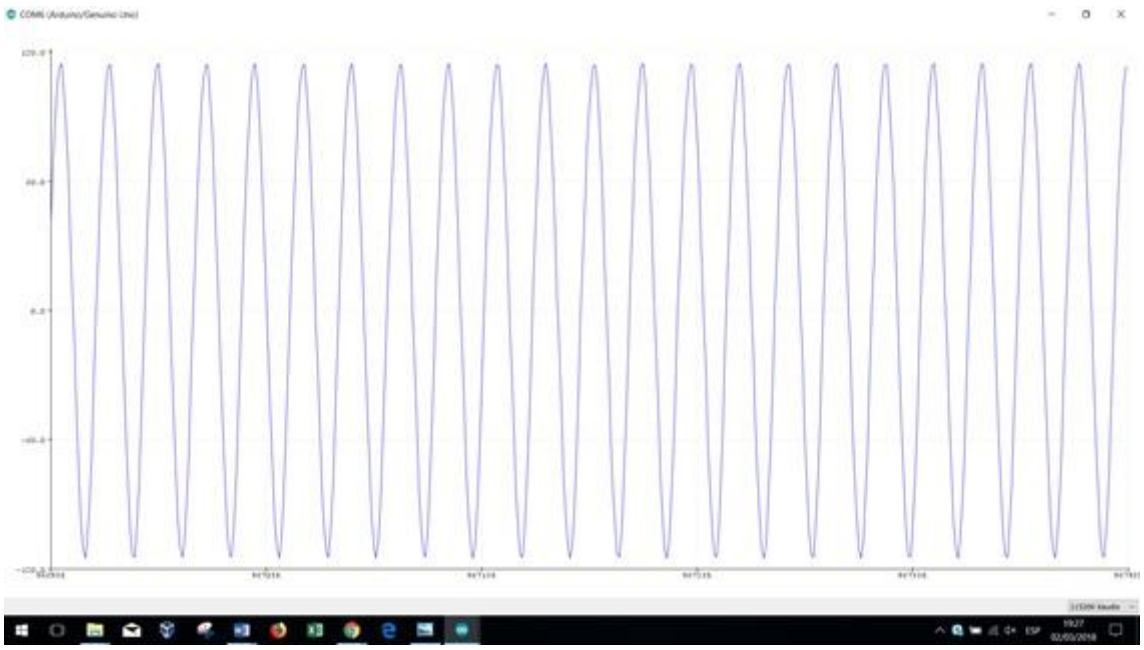

Figura 3. 21 : Grafica de Salida de señal de Arduino.

3.3 Graficas en Matlab.

Mediante el uso de la información grabada previamente en nuestro dispositivo de almacenamiento micro SD y con la asistencia de la programación en Matlab se generan las gráficas de los consumos de las variables alternas con relación al tiempo, tal como se muestra en la figura 3.22 donde se muestran los valores en un periodo de tiempo determinado por el usuario.

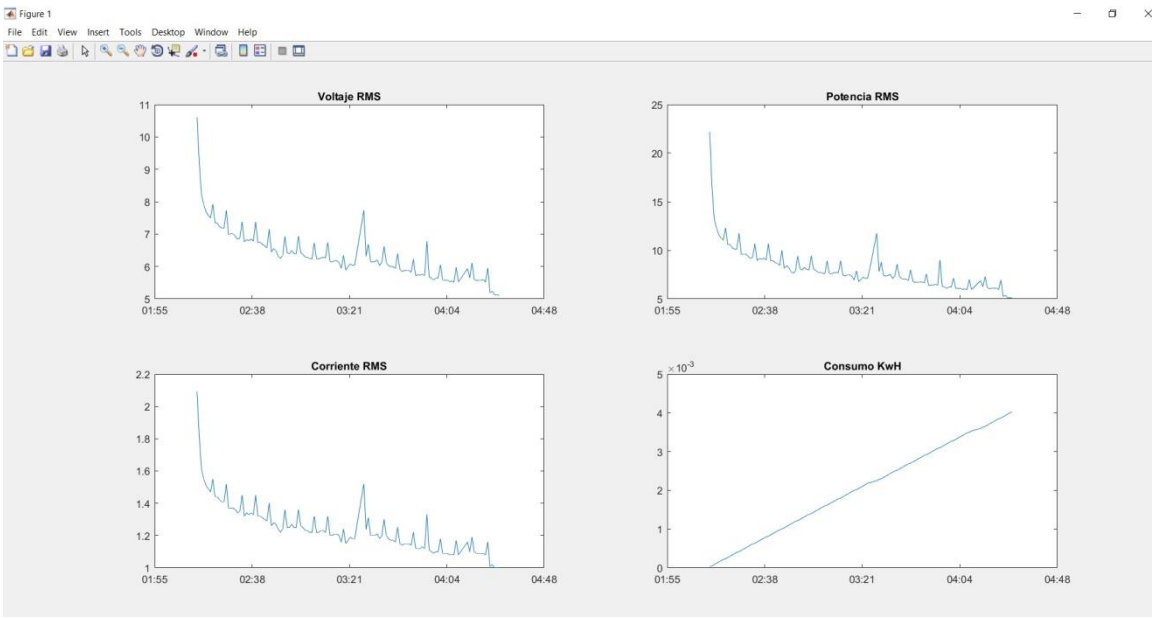

Figura 3.22: Graficas de Voltaje RMS, corriente RMS, Potencia RMS y Consumo

En la figura 3.23 se puede observar parte del código de programación de Matlab el cual permite la interacción del usuario que desee observar gráficamente la información grabada, usando la función “str_to” para ingresar la fecha inicial del periodo a graficar y “str_tf” para ingresar la fecha final del periodo. La función “Root” se usa para encontrar la ubicación en nuestro ordenador donde se encuentra la micro SD insertada y la Función “Plot” para mostrar las gráficas en una nueva pantalla

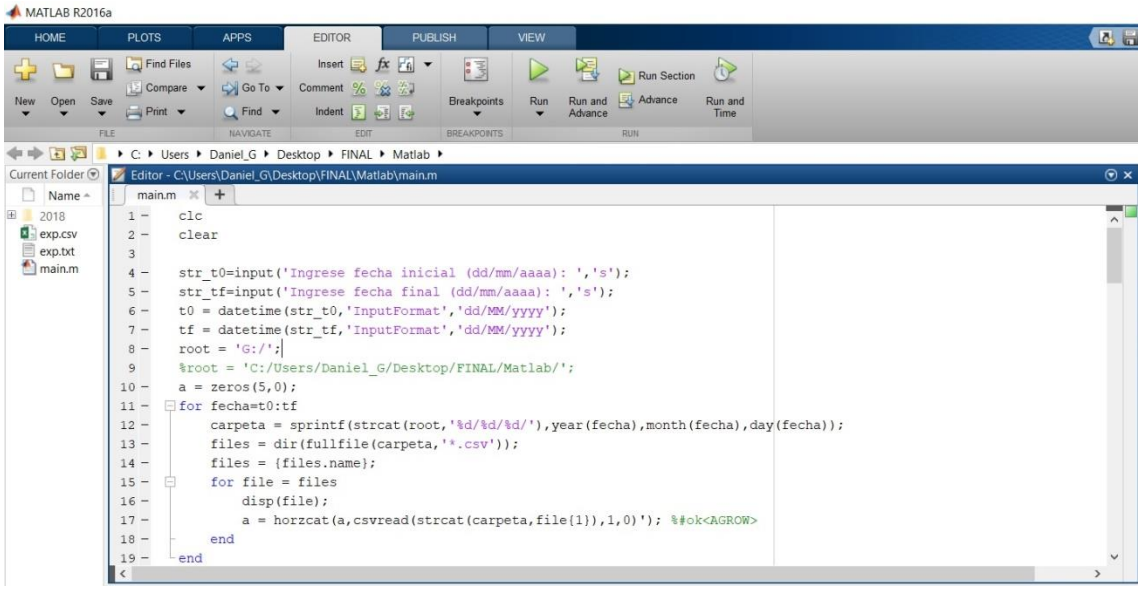

Figura 3.23: Programación en Matlab

CONCLUSIONES Y RECOMENDACIONES

- Se obtuvo un diseño de un prototipo de forma económica y funcional que permite registrar y almacenar las variables eléctricas en alterna y por ser desarrollado en una plataforma libre permitirá realizar mejoras.

- Con los datos almacenados en forma de registro histórico se puede realizar un análisis de eficiencia de las instalaciones que permitan tomar medidas correctivas.

- Para la producción masiva del prototipo se puede prescindir de la placa de desarrollo y usar solo el micro controlador integrándolo en la placa general, lo que reduciría dimensiones y costos.

- Se puede mejorar el diseño utilizando módulos de comunicación bluetooth o gsm para una conectividad online y que siga las tendencias de la tecnología IOT.

- Para la señal de entrada de voltaje en alterna del diseño es imprescindible verificar la polaridad de la red eléctrica porque una polaridad inversa provocaría daños en los integrados y esto se reflejará en la toma de datos.

- El diseño se desarrolló en función de un sensor de corriente SCT 013 el cual adquiere datos hasta de 30 amperios. Para valores superiores a ese valor es necesario rediseñar el circuito del sensor de corriente para evitar ruidos y señales erróneas.

- Para recibir la señal de entrada se puede utilizar un cable de alimentación con terminales tipo lagarto o tipo pinza para una fácil colocación del dispositivo.

BIBLIOGRAFÍA

- Arduino Software libre de desarrollo [online] disponible en https://www.arduino.cc/
- •Arduino Circuits and Projects Guide by Elektor 2014.
- •Electronic Circuits for All by Michael Shustov & Andrey Shustov of Elektor International Media 2016
- •Sensores de corriente [online] disponible en http://www.naylampmechatronics.com/blog/51_tutorial-sensor-de-corriente-ac-no-invasivo-s.html

Anexos.

Código de Programación de Arduino

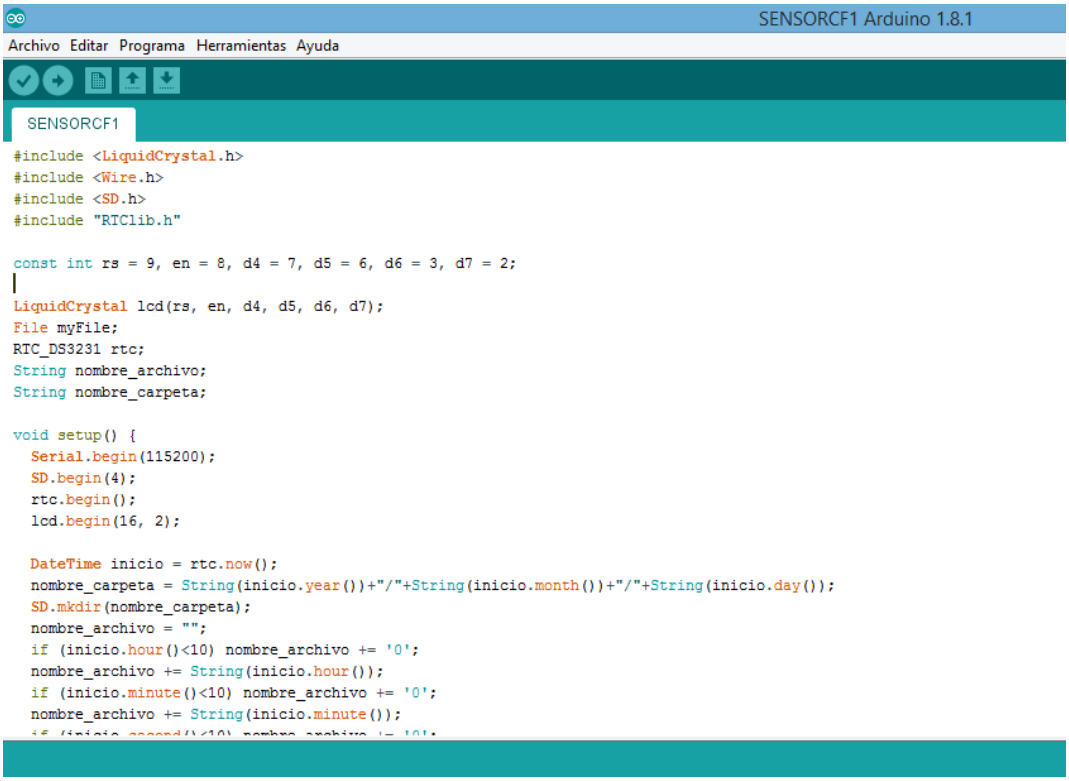

Figura 4.16: Código de programación.

```
#include <LiquidCrystal.h>
#include <Wire.h>
#include <SD.h>
#include "RTClib.h"
const int rs = 9, en = 8, d4 = 7, d5 = 6, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
File myFile;
RTC_DS3231 rtc;
String nombre_archivo;
String nombre_carpeta;
void setup() {
```

```
  Serial.begin(115200);
  SD.begin(4);
  rtc.begin();
  lcd.begin(16, 2);
    DateTime inicio = rtc.now();
  nombre_carpeta =
String(inicio.year())+"/"+String(inicio.month())+"/"+String(inicio.day());
  SD.mkdir(nombre_carpeta);
  nombre_archivo = "";
  if (inicio.hour()<10) nombre_archivo += '0';
  nombre_archivo += String(inicio.hour());
  if (inicio.minute()<10) nombre_archivo += '0';
  nombre_archivo += String(inicio.minute());
  if (inicio.second()<10) nombre_archivo += '0';
  nombre_archivo += String(inicio.second())+".csv";
}

void loop() {
  DateTime now = rtc.now();
  if(now.hour()==0 && now.minute()==0 && now.second()<10){
    nombre_archivo = "";
    if (now.hour()<10) nombre_archivo += '0';
    nombre_archivo += String(now.hour());
    if (now.minute()<10) nombre_archivo += '0';
    nombre_archivo += String(now.minute());
    if (now.second()<10) nombre_archivo += '0';
    nombre_archivo += String(now.second())+".csv";
    nombre_carpeta =
String(now.year())+"/"+String(now.month())+"/"+String(now.day());
    SD.mkdir(nombre_carpeta);
    while(now.second()<10)now = rtc.now();
  }
   get_valores();
}
```

```
void get_valores()
{
  float VSVoltaje;
  float Voltaje=0;
  float SVoltaje=0;
  float VSCorriente;
  float corriente=0;
  float SCorriente=0;
  long tiempo=millis();
  int N=0;
  myFile = SD.open(nombre_carpeta+"/"+nombre_archivo, FILE_WRITE);
  while(millis()-tiempo<500)//Duración 0.5 segundos(Aprox. 30 ciclos de 60Hz)
    {
      VSVoltaje = analogRead(A0) * (500 / 1023.0);////voltaje del sensor voltaje
      Voltaje=(VSVoltaje-100)*1.25; //Voltaje= voltaje del sensor voltaje* 150
      SVoltaje=SVoltaje+sq(Voltaje);//Sumatoria de Cuadrados
      N=N+1;


      VSCorriente = analogRead(A1) * (1.2 / 1023.0);////voltaje del sensor
      corriente=(VSCorriente*30.0)-10; //corriente=Voltaje Sensor
corriente*(30A/1V)
      SCorriente=SCorriente+sq(corriente);//Sumatoria de Cuadrados
      N=N+1;

Serial.println(String(Voltaje)+","+String(corriente)+","+String(Voltaje*corriente));
      //Serial.println(String(corriente));
      delay(1);
    }
    Voltaje=sqrt((SVoltaje)/N); //ecuación del RMS
    SCorriente=SCorriente*1;//Para compensar los cuadrados de los semiciclos
negativos.
    corriente=sqrt((SCorriente)/N); //ecuación del RMS

myFile.println(String(millis())+","+String(Voltaje)+","+String(corriente)+","+String(
Voltaje*corriente));
```

```
    myFile.close();

    lcd.clear();
    lcd.print("Vrms: ");
    lcd.print(Voltaje,3);
    lcd.print("V");

    lcd.setCursor(0,1);
    lcd.print("Irms: ");
    lcd.print(corriente,3);
    lcd.print("A");
}
float get_corriente()
{
  float VSCorriente;
  float corriente=0;
  float SCorriente=0;
  long tiempo=millis();
  int N=0;
  while(millis()-tiempo<500)//Duración 0.5 segundos(Aprox. 30 ciclos de 60Hz)
  {
    VSCorriente = analogRead(A5) * (1.2 / 1023.0);////voltaje del sensor
    corriente=VSCorriente*30.0; //corriente=Voltaje Sensor corriente*(30A/1V)
    SCorriente=SCorriente+sq(corriente);//Sumatoria de Cuadrados
    N=N+1;
    delay(1);
  }
  SCorriente=SCorriente*1;//Para compensar los cuadrados de los semiciclos
negativos.
  corriente=sqrt((SCorriente)/N); //ecuación del RMS
  return(corriente);
}
```

Código de programación de Matlab

```
clc
clear

str_t0=input('Ingrese fecha inicial (dd/mm/aaaa): ','s');
str_tf=input('Ingrese fecha final (dd/mm/aaaa): ','s');
t0 = datetime(str_t0,'InputFormat','dd/MM/yyyy');
tf = datetime(str_tf,'InputFormat','dd/MM/yyyy');
a = zeros(4,0);
for fecha=t0:tf
    carpeta =
sprintf('F:/%d/%d/%d/',year(fecha),month(fecha),day(fecha));
    files = dir(fullfile(carpeta,'*.csv'));
    files = {files.name};
    for file = files
        disp(file);
        a = horzcat(a,csvread(strcat(carpeta,file{1}),1,0)');
%#ok<AGROW>
    end
end

subplot(3,1,1);
plot(a(1,:),a(2,:));
title('Voltaje RMS')

subplot(3,1,2);
plot(a(1,:),a(3,:));
title('Corriente RMS')

subplot(3,1,3);
plot(a(1,:),a(4,:));
title('Potnecia RMS')

% x = 4*pi:.1:6*pi;
% y = [x; sin(x);cos(x);sin(x).*cos(x)];
% fid = fopen('exp.txt', 'wt');
% fprintf(fid, '%6.2f,%12.8f,%12.8f,%12.8f\n', y);
% fclose(fid);
```