



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

DISEÑO E IMPLEMENTACIÓN DE UN OSCILOSCOPIO DIGITAL DE BAJO COSTO

Miguel Eduardo Argueta Garrido

Asesorado por el Ing. Carlos Eduardo Guzmán Salazar

Guatemala, septiembre de 2019

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO E IMPLEMENTACIÓN DE UN OSCILOSCOPIO DIGITAL DE BAJO
COSTO**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

MIGUEL EDUARDO ARGUETA GARRIDO

ASESORADO POR EL ING. CARLOS EDUARDO GUZMÁN SALAZAR

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN ELECTRÓNICA

GUATEMALA, SEPTIEMBRE DE 2019

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Luis Diego Aguilar Ralón
VOCAL V	Br. Christian Daniel Estrada Santizo
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. Marvin Marino Hernández Fernández
EXAMINADOR	Ing. Byron Odilio Arrivillaga Méndez
EXAMINADORA	Inga. María Magdalena Puente Romero
SECRETARIA	Inga. Lesbia Magalí Herrera López

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DISEÑO E IMPLEMENTACIÓN DE UN OSCILOSCOPIO DIGITAL DE BAJO COSTO

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 13 de julio de 2017.



Miguel Eduardo Argueta Garrido

Guatemala, 14 de mayo de 2019

Ingeniero

Julio Solares Peñate

Coordinador Área de Electrónica

Escuela de Ingeniería Mecánica Eléctrica

Facultad de Ingeniería

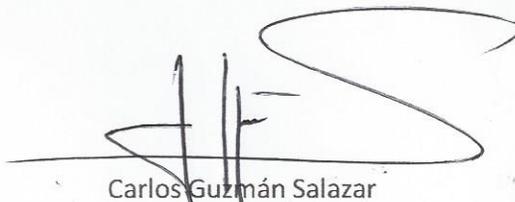
Universidad de San Carlos de Guatemala

Estimado ingeniero Solares:

Hago de su conocimiento que he finalizado la revisión del trabajo de graduación del estudiante **MIGUEL EDUARDO ARGUETA GARRIDO**, titulado **DISEÑO E IMPLEMENTACIÓN DE UN OSCILOSCOPIO DIGITAL DE BAJO COSTO**. El que ha cumplido con los objetivos que se plantearon para su desarrollo y ejecución. Asimismo, el prototipo construido muestra a cabalidad el funcionamiento del mismo. Por lo que, en mi calidad de ASESOR nombrado por la Escuela de Ingeniería Mecánica Eléctrica, doy mi APROBACIÓN al mismo.

Tanto el estudiante Argueta Garrido como el suscrito, somos responsables del contenido del trabajo de graduación en cuestión.

Reciba usted un cordial saludo.



Carlos Guzmán Salazar

ASESOR

CARLOS GUZMAN SALAZAR
Ingeniero Electricista
Col. No. 2762

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA

REF. EIME 34. 2019.

21 de mayo 2019.

Señor Director

Ing. Otto Fernando Andrino González
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

Señor Director:

Me permito dar aprobación al trabajo de Graduación titulado: **DISEÑO E IMPLEMENTACIÓN DE UN OSCILOSCOPIO DIGITAL DE BAJO COSTO**, del estudiante; Miguel Eduardo Argueta Garrido, que cumple con los requisitos establecidos para tal fin.

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,
ID Y ENSEÑAD A TODOS


Ing. Julio César Solares Peñate
Coordinador de Electrónica



UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA

REF. EIME 34. 2019.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto bueno del Coordinador de Área, al trabajo de Graduación de el estudiante: MIGUEL EDUARDO ARGUETA GARRIDO titulado: DISEÑO E IMPLEMENTACIÓN DE UN OSCILOSCOPIO DIGITAL DE BAJO COSTO, procede a la autorización del mismo.


Ing. Otto Fernando Andriño González

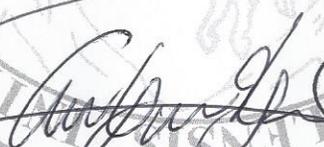
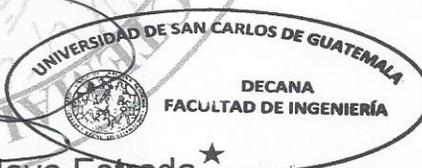


GUATEMALA, 28 DE JUNIO 2019.



La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al trabajo de graduación titulado: **DISEÑO E IMPLEMENTACIÓN DE UN OSCILOSCOPIO DIGITAL DE BAJO COSTO**, presentado por el estudiante universitario: **Miguel Eduardo Argueta Garrido**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, se autoriza la impresión del mismo.

IMPRÍMASE.



Inga. Aurelia Anabela Cordova Estrada ★
Decana

Guatemala, Septiembre de 2019

/cc

ACTO QUE DEDICO A:

- Dios** Por haberme dado la vida, salud y bendición, por darme la fuerza para cumplir mis metas y sueños como persona.
- Mis padres** Eduardo Argueta y Alma Garrido, por su amor incondicional y su apoyo incansable.
- Mis hermanas** Andrea y Laura Argueta, por ser mis mejores amigas, con quienes puedo contar siempre.
- Mis abuelos** Rolando Garrido y Laura Reyes, por su cariño y aprecio.
- Mis tíos** Por ser una importante influencia en mi vida y apoyo en todo momento.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por darme la oportunidad de estudiar.
Facultad de Ingeniería	Por formarme profesionalmente.
Mis amigos	Por todo lo bueno y lo malo a lo largo de la carrera.
Ing. Eduardo Guzmán	Por su apoyo y asesoría en este trabajo de graduación.

1.6.	Matlab	6
2.	DISEÑOS DEL PROYECTO	9
2.1.	Diseño de las placas de circuitos impresos de entradas.....	9
2.1.1.	Divisor de voltaje como entrada de señal.....	10
2.1.2.	Sumador de voltaje con amplificadores operacionales para elevar voltaje.....	13
2.1.3.	Diseño del mezclador de voltaje.....	15
2.1.4.	Diseño del filtro pasa bajo	16
2.1.5.	Fuente de alimentación de los amplificadores de operaciones.....	17
2.2.	Microcontrolador Tiva C	20
2.2.1.	Programación de Tiva C con programa Energia	21
2.2.2.	Código fuente del proyecto en Energia	23
2.3.	Entorno Matlab.....	24
2.3.1.	Lenguaje M.....	25
2.3.2.	Código fuente en lenguaje M.....	25
3.	ESPECIFICACIONES TÉCNICAS.....	33
3.1.	Interconexión del divisor de voltaje.	33
3.2.	Conexión entre el divisor de voltaje y los amplificadores operacionales.....	35
3.3.	Microcontrolador y placas de tratamiento de señal analógica.....	36
3.4.	Ensamblaje final.....	38
4.	ESPECIFICACIONES DE USO	41
4.1.	Colocación de la punta de osciloscopio	41
4.2.	Selección de la escala.....	42

4.3.	Funcionamiento y uso de la aplicación gráfica	43
4.3.1.	Botones	44
4.3.2.	Área de graficación.....	45
CONCLUSIONES		47
RECOMENDACIONES.....		49
BIBLIOGRAFÍA.....		51
APÉNDICES		53
ANEXOS		55

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Diseño esquemático del divisor de voltaje	11
2.	Divisor de voltaje en diseño PCB	12
3.	Foto de la placa del divisor.....	13
4.	Diseño del sumador de voltaje	14
5.	Diseño del mezclador de voltaje	15
6.	Filtro pasa bajos	16
7.	Convertidor de voltaje	17
8.	Diseño unificado del elevador de voltaje, sumador, filtro y convertidor de voltaje	19
9.	Diagrama PCB de los diseños con amplificadores operacionales	19
10.	Placa de amplificadores operacionales	20
11.	Tiva C	21
12.	Interfaz gráfica de Energia	22
13.	Código fuente en Energia.....	24
14.	Función para establecer comunicación serial y declaración de variables globales	26
15.	Creación de la pantalla que se utilizará para graficar	26
16.	Botón para agrandar gráfica en eje Y.....	27
17.	Botón para mover la gráfica en eje Y	28
18.	Botón para agrandar la gráfica en eje X.....	28
19.	Botón para mover la gráfica en eje X	29
20.	Creación de una matriz de ceros, variable i de iteraciones en eje Y, variable t para iteraciones en eje X.....	29

21.	Ciclo <i>while</i>	30
22.	Imprimir en pantalla	31
23.	Funciones de los botones	32
24.	Punta de osciloscopio	33
25.	Conector BNC.....	34
26.	Conector BNC en la caja de conexión	34
27.	Conector BNC y placa del divisor de voltaje	35
28.	Conexión de placa de divisor de voltaje y placa de amplificadores operacionales	36
29.	Conexión entre placa de amplificadores operacionales y microcontrolador Tiva C.....	37
30.	Declaración de puerto A7 en Energia	37
31.	Todas las placas acomodadas	38
32.	Vista del proyecto ensamblado	39
33.	Colocación de punta de osciloscopio.....	41
34.	Conexión a tierra de la punta de osciloscopio	42
35.	Selección de escala	43
36.	Botones de la aplicación	45
37.	Gráfica de una señal en la aplicación	46

LISTA DE SÍMBOLOS

Símbolo	Significado
C	Capacitor
CAP	Capacitor
Cm	Centímetros
U	Circuito integrado
X	Eje x del plano cartesiano
Y	Eje y del plano cartesiano
GND	Ground
Hz	Hertz
INT	Input
KHz	Kilo Hertz
KΩ	Kilo Ohmio
MHz	Mega Hertz
μF	Micro Faradio
nF	Nano Faradio
Ω	Ohmio
OUT	Output
A7	Puerto analógico 7 del microcontrolador
R	Resistencia
SW	<i>Switch</i>
SW-ROT	<i>Switch</i> rotativo multiposición
F7	Tecla F7 de un ordenador
J	Terminal Jump
V	Voltio

GLOSARIO

ADC	<i>Analogic Digital Conversor.</i>
Amplificador operacional	Circuito integrado analógico amplificador de señales de alta ganancia.
ARM	<i>Advanced RISC Machine.</i>
Baudio	Unidad de medida de la velocidad de transmisión de señales que se expresa en símbolos por segundo.
BNC	Bayonet Neill-Concelman.
Binario	Sistema de numeración compuesto de dos elementos: cero y uno.
C#	Lenguaje de programación orientado a objetos.
CPU	<i>Central Processing Unit.</i>
Java	Lenguaje de programación orientado a objetos.
Lenguaje M	Lenguaje de programación propio de Matlab.
LM358	Circuito integrado de ocho pines, con dos amplificadores operacionales en su interior.

Matlab	Entorno computacional de desarrollo numérico integrado.
MAX660	Circuito integrado convertidor de voltaje positivo a negativo.
Microcontrolador	Circuito integrado programable.
Ohm	Unidad de medida de la resistencia eléctrica.
Osciloscopio	Herramienta graficadora de ondas eléctricas.
PCB	<i>Printed Circuit Board.</i>
RISC	<i>Reduced Instruction Set Computer.</i>
Señal	Onda eléctrica.
Serial	Técnica de comunicación que utiliza únicamente dos pares de cables para establecer la comunicación, uno para recibir y el otro para enviar.
Tiva C	Microcontrolador de placa integrada de Texas Instrument.
UART	<i>Universal Asynchronous Receiver-Transmitter.</i>
USB	<i>Universal Serial Bus.</i>

Voltio

Unidad de medida del voltaje eléctrico.

While

Ciclo de programación utilizado para repetir una tarea hasta que se cumpla una condición.

RESUMEN

El presente documento se realiza con el objetivo de diseñar e implementar un osciloscopio de bajo costo que sea capaz de graficar señales para ser utilizado en el diseño de circuitos electrónicos analógicos y digitales. El propósito es tener un osciloscopio construido a partir de materiales al alcance de un estudiante y que pueda construir su propia herramienta.

Para facilitar el entendimiento de los diseños del proyecto se inicia con la explicación de los fundamentos teóricos necesarios, seguido de la presentación de los diseños de los divisores de voltaje, los diseños con amplificadores operaciones, el microcontrolador elegido, así como su configuración y la programación utilizada en lenguaje M de Matlab.

Luego de la presentación de los diseños se explica cómo se conectan los circuitos entre sí, la interacción que hay entre ellos y cómo juntos conforman el sistema de recolección de datos analógicos, posteriormente digitalizados e interpretados para ser graficados.

Finalmente se presenta una pequeña guía de usuario que explica cómo se debe colocar la punta del osciloscopio, seleccionar una escala, cómo correr un programa en Matlab y el funcionamiento de cada uno de los botones agregados en la aplicación graficadora del ordenador.

OBJETIVOS

General

Diseñar y construir un osciloscopio digital que cumpla la función básica de graficar señales y su precio sea menor a un osciloscopio comercial.

Específicos

1. Diseñar e implementar, por medio de divisores de voltaje, amplificadores operacionales y filtros, un circuito de entrada de señales que sea capaz de evitar que el microcontrolador reciba voltajes muy altos o negativos.
2. Convertir por medio de un microcontrolador la señal analógica a digital, utilizando el ADC integrado en el microcontrolador a manera de convertir cualquier tipo de señal continua en el tiempo en una señal discreta y codificada en el tiempo, lista para ser transmitida por protocolo UART hacia un ordenador.
3. Utilizar un lenguaje de programación para desarrollar en una computadora una aplicación que reconstruya la señal a partir de los datos recibidos por UART y graficar la señal en una ventana.

INTRODUCCIÓN

En el estudio de ingeniería eléctrica o electrónica el diseño de circuitos electrónicos analógicos y digitales es una tarea que se repetirá a lo largo de los años y los osciloscopios son una herramienta muy utilizada y esencial en el diseño de circuitos. Debido a su alto costo, no todos tienen la posibilidad de tener un osciloscopio en casa. En la universidad existe el laboratorio de electrónica, donde se brinda un número limitado de osciloscopios para que los estudiantes puedan hacer uso de ellos, el laboratorio tiene limitaciones de uso como el horario.

Para dar una solución al problema de no tener un osciloscopio a cualquier hora y en cualquier lugar se diseña e implementa una herramienta con capas para graficar señales eléctricas, utilizando para su construcción materiales al alcance de todos y diseñado con teoremas y principios aprendidos durante la carrera. La adquisición de datos está a cargo de un microcontrolador junto con circuitos de electrónica analógica para el tratamiento de señal. La tarea de graficación es llevada a cabo por un ordenador.

En el capítulo uno se presentan los fundamentos teóricos necesarios para entender los diseños del proyecto. En el capítulo dos se abordan los diseños de circuitos, incluyendo la programación del microcontrolador y la programación del ordenador. El capítulo tres se trata de la interacción entre los diseños y la forma de conectarlos. Finalmente, en el capítulo cuatro se explica cómo funciona el proyecto y cómo utilizarlo.

1. MARCO TEÓRICO

Un osciloscopio digital requiere de varias etapas para poder analizar una señal. A continuación, se presentan los fundamentos teóricos necesarios de los componentes electrónicos y definiciones que servirán para poder explicar el diseño y posteriormente la implementación de un osciloscopio digital.

1.1. Electrónica

Es el estudio de sistemas basados en la conducción de flujo de electrones, así como de su control. Para lograr dicho control se emplea una gran variedad de materiales conductores y semiconductores, los cuales dan como resultado dos tipos importantes de dispositivos: los dispositivos electrónicos analógicos y los dispositivos electrónicos digitales. En la electrónica analógica los valores pueden ser continuos y prácticamente infinitos, mientras que la digital solo permite dos valores, ceros y unos o positivo y negativo. Esta electrónica está presente en los integrados, ya que están diseñados para poder operar números binarios para realizar cálculos lógicos y matemáticos¹.

1.2. Divisores de voltaje

Es una regla que indica que el voltaje aplicado a una serie de resistencias se divide entre la magnitud de las resistencias conectadas en el circuito. Se utiliza la Ley de Ohm para calcular el valor de una resistencia o el voltaje en un nodo del circuito².

¹ FLOYD, Thomas L. *Fundamentos de sistemas digitales*. p 4.

² BOYLESTAD, Robert L. *Introducción al análisis de circuitos*. p 138.

1.3. Osciloscopio

Un osciloscopio es una herramienta de visualización que sirve para graficar la forma de onda de señales variantes en el tiempo. Existen dos tipos de osciloscopios: los analógicos y los digitales. La diferencia entre ambos tipos consiste en las técnicas que se utilizan para poder generar la gráfica de la onda eléctrica.

1.3.1. Osciloscopios de tubo de rayos catódicos

Son aparatos que utilizan la mayor parte de dispositivos electrónicos analógicos. Se caracterizan por tener una pantalla de tubo de rayos catódicos para poder graficar las señales, este tubo es muy similar a los televisores antiguos y utiliza dos campos eléctricos para poder desviar las ondas eléctricas y dar el efecto de movimiento en la pantalla. Un campo eléctrico sirve para mover en dirección vertical y el otro en dirección horizontal.

1.3.2. Osciloscopios digitales

Son aparatos que utilizan mayormente dispositivos electrónicos digitales para poder graficar las ondas eléctricas. Su principal diferencia con los anteriores es que para graficar se utilizan técnicas digitales y una pantalla LCD o LED en vez de la pantalla de tubo de rayos catódicos³.

³ HELFRICK, Albert D.; COOPER, William D. *Instrumentación eléctrica moderna y técnicas de medición*. p 186.

1.4. Amplificador operacional

Es un dispositivo electrónico analógico integrado, su principal aplicación es la de amplificar señales o voltajes de corriente directa. Entre sus propiedades están una alta inmunidad al ruido, alta ganancia, alta impedancia de entrada y una baja impedancia de salida. En su representación más básica cuenta con dos entradas. La primera es la no inversora, la señal aplicada a esta no será invertida en la salida, la otra entrada es la inversora, esta invertirá la señal aplicada en ella en la salida, cuando se esté usando una o la otra entrada se debe conectar la restante a tierra. Otra de sus aplicaciones importantes sería la de poder hacer filtros activos con ellos.

1.4.1. Sumador de voltaje

Es una de las configuraciones del amplificador operacional, sirve para aumentar de 0v a un voltaje determinado, para que en su salida eleve el nivel de referencia a un valor de voltaje seleccionado. Este nivel de voltaje puede ser definido con divisores de voltaje a la entrada no inversora del amplificador operacional⁴.

1.4.2. Mezclador de voltaje

Es otra de las configuraciones del amplificador operacional y su función es la de unir dos señales o más. En su salida presenta la combinación de las dos señales o voltajes. Su configuración es simple y consiste en agregar cada señal con una resistencia en su entrada no inversora.

⁴ SEDRA, Adel S.; SMITH, Kenneth C. *Circuitos microelectrónicos*. p 60.

1.4.3. Filtros activos

Un filtro activo es el resultado de unir resistencias, capacitores y dispositivos amplificadores en una configuración apropiada para discriminar o la no discriminación de determinadas frecuencias de una señal. Este dispositivo amplificador puede ser un transistor o un amplificador operacional⁵.

1.5. Procesamiento digital de señal

Una señal para analizar por un computador o un microcontrolador primero debe convertirse en una señal digital, es decir convertirla en un sistema binario con solo dos valores, de esta manera la información contenida puede ser operada matemáticamente, procesada y hasta modificada para mejorar su contenido. Algunos de los procesos utilizados para lograr el procesamiento digital se describen a continuación.

1.5.1. Conversión analógica digital (ADC)

Consiste en transformar una señal continua en el tiempo en una señal discreta en el tiempo, esto se logra multiplicando la señal continua por un tren de impulsos, el resultado de tal multiplicación es la señal discreta en el tiempo. Los impulsos tomarán la amplitud de la señal continua y pueden ser codificados con un valor discreto, es decir un valor binario. El nuevo valor binario puede ser procesado o almacenado para su posterior utilización.

⁵ MALVINO, Albert Paul. *Principios de electrónica*. p 781.

1.5.2. Microcontrolador

Es un circuito integrado digital programable, a diferencia de un circuito integrado común, el cual tiene una función específica, el microcontrolador puede ser programado para realizar la tarea que se desee. Los microcontroladores también pueden ser de varios tipos o gamas, los de gama más baja pueden realizar menos tareas que los de gama alta pero su precio es mucho menor. Los de gama alta también pueden procesar señales incluyendo la conversión analógica digital.

1.5.3. Protocolos de comunicación

Un protocolo de comunicación es un proceso de transmisión de datos o señales a través de un medio, los más básicos se dan por medios guiados y los más complejos por medios inalámbricos. Entre los medios guiados están los protocolos seriales y los paralelos, la diferencia radica en que en los seriales se enviarán los datos por un solo hilo conductor, mientras que en los paralelos la transmisión será por varios hilos conductores al mismo tiempo. Cada una de estas formas de enviar los datos tiene sus ventajas y desventajas.

1.5.3.1. Transmisión paralela

Tiene la ventaja de transmitir varios bits de información al mismo tiempo, por consecuencia es más compleja su conexión al tener más alambres conductores actuando al mismo tiempo. Este tipo de transmisión es más común en el interior de los microcontroladores y sus periféricos cercanos y menos comunes en sus periféricos más lejanos.

1.5.3.2. Transmisión serial

En su forma más básica solo necesita de los dos alambres conductores de alimentación y dos alambres de transmisión de datos, uno para recibir los datos y el otro para transmitirlos. Su mayor desventaja es que al tener que pasar todos los bits de información por un solo hilo conductor su transmisión es más lenta, aunque en los últimos tiempos esto ya no es una desventaja, ya que las velocidades de transmisión se han multiplicado y mejorado. Su principal ventaja es la poca cantidad de alambres utilizados para lograr transmitir datos.

1.5.3.3. Receptor-transmisor asíncrono universal (UART)

Es un protocolo de comunicación serial, es uno de los más básicos ya que solo necesita de un cable receptor, un cable transmisor y los pines de alimentación. La mayoría de los microcontroladores pueden programarse para usar este protocolo, aunque hay circuitos integrados que pueden usarse para hacer este tipo de transmisión. Las computadoras también pueden programarse para utilizarse. Es uno de los protocolos de comunicación más extendido por su simplicidad y su efectividad⁶.

1.6. Matlab

Es un entorno de desarrollo matemático para computadoras que incluye, entre otras cosas, un lenguaje propio de programación. Entre las funciones más destacadas de Matlab está el poder analizar señales digitales. Consta de una línea de comandos donde se puede escribir los comandos y funciones que se

⁶ TOMASI, Weyne. *Sistemas de comunicaciones electrónicas*. p 546.

dese ejecutar. Su lenguaje de programación, llamado lenguaje M, permite crear aplicaciones para computadoras⁷.

⁷ COUCH, Leon. *Sistemas de comunicación digitales y analógicos*. p 7.

2. DISEÑOS DEL PROYECTO

Para poder graficar una señal un osciloscopio primero debe cumplir con una serie de etapas, la primera de ellas es tratar la señal, que consiste en tomar una muestra de la señal y transformarla en niveles lógicos que puedan ser procesados por un microcontrolador. Luego el microcontrolador aplicará procesos matemáticos o lógicos para generar una señal digital que tenga la información necesaria que será interpretada por un ordenador para graficar la señal.

2.1. Diseño de las placas de circuitos impresos de entradas

Las placas de entrada son las encargadas de mantener los niveles de voltaje adecuados para que el microcontrolador no sea dañado con voltajes mayores a los que soporta. A continuación se presentan los diseños utilizados en la etapa de recolección de muestras de señal. Estos diseños cumplen con tres objetivos principales:

- Proteger al microcontrolador de voltajes mayores a 3.3 voltios, que es el voltaje máximo que puede leer.
- Proteger al microcontrolador de voltajes negativos.
- Preparar la señal con los valores adecuados para que el microcontrolador pueda interpretarla sin alterar su forma.

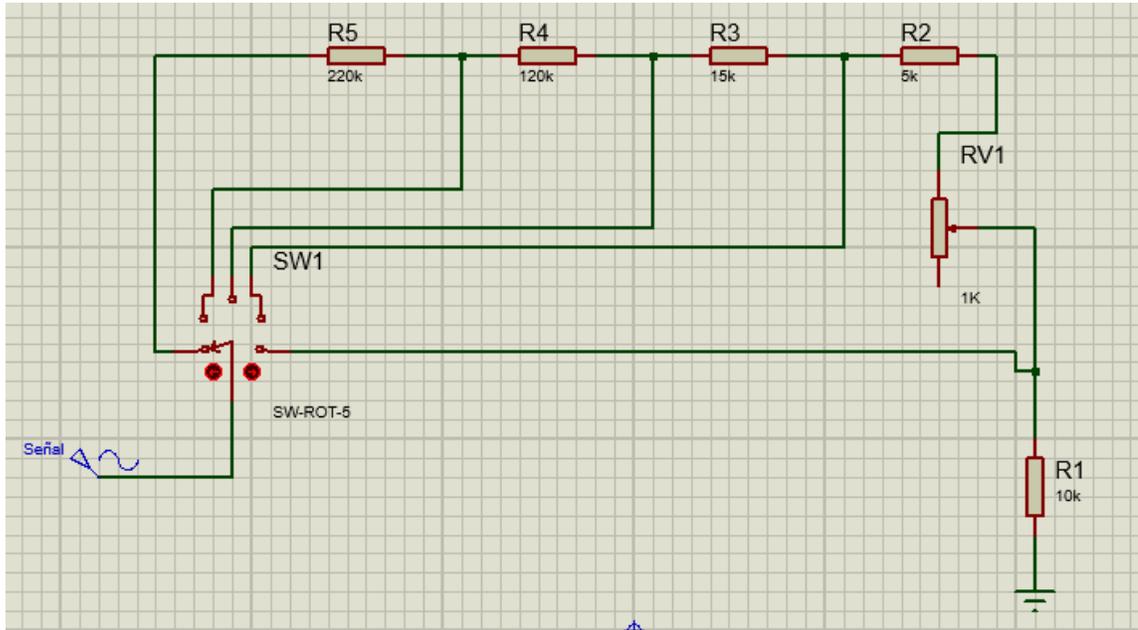
2.1.1. Divisor de voltaje como entrada de señal

Es un arreglo de resistencia, sirve para bajar los niveles de voltaje mayores a 1,65 voltios tanto positivos como negativos, se explicará más adelante porque el voltaje mayor es 1,65 voltios y no 3,3 voltios como el del microcontrolador. Como en un multímetro, se diseñaron cinco escalas de medición con rangos de voltaje que se pudo medir, estos voltajes son:

- Rango 1: de -1,65 V a 1,65 V
- Rango 2: de -2,5 V a 2,5 V
- Rango 3: de -5 V a 5 V
- Rango 4: de -25 V a 25 V
- Rango 5: de -62,5 V a 62,5 V

Por ejemplo, si se desea graficar una señal y se sabe que posee un voltaje que varía entre -1 V y 3 V se debe seleccionar la herramienta en la escala tres para evitar que el microcontrolador se quemara con voltajes mayores a 2,5 V, como sería el caso de seleccionar el rango dos.

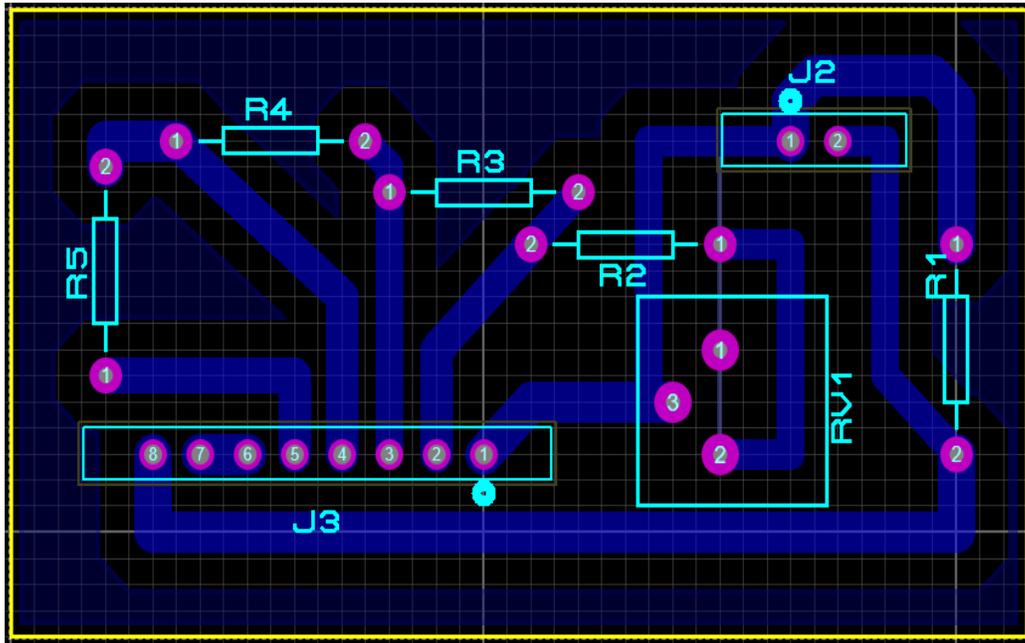
Figura 1. Diseño esquemático del divisor de voltaje



Fuente: elaboración propia, empleando Proteus 8.

En la figura 1, la resistencia R1 es utilizada como la carga que mantendrá el voltaje de salida en $\pm 1,65$ V, su valor es de 10 K Ω , el SW1 es un selector que permite seleccionar entre R2 a R5, las cuales son las resistencias que completan el divisor de voltaje y la carga necesaria que modifica la salida para mantener el voltaje máximo de referencia en la salida.

Figura 2. Divisor de voltaje en diseño PCB



Fuente: elaboración propia, empleando Proteus 8.

En la figura 2 se muestra el diseño en circuito impreso del divisor de voltaje propuesto. Adicional a las resistencias utilizadas, se agrega J2 y J3 que son los arreglos de terminales de donde saldrán los cables, en el caso de J2 hacia la siguiente placa y J3 hacia un puente que seleccionará el rango a medir que puede ser un selector. En la figura 3 se muestra la foto del circuito electrónico ya en placa y con componentes soldados.

Figura 3. Foto de la placa del divisor



Fuente: elaboración propia.

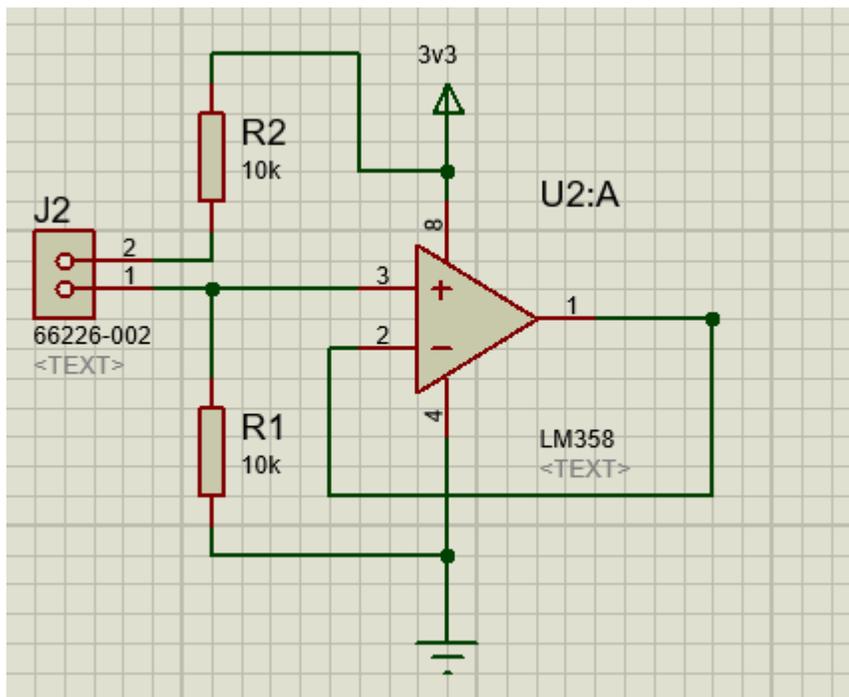
2.1.2. Sumador de voltaje con amplificadores operacionales para elevar voltaje

Como se mencionó anteriormente, el microcontrolador solo lee voltajes en el rango de 0 a 3,3 V. Para arreglar este problema se utiliza una técnica que se llama elevación de voltaje, la cual consiste en cambiar la referencia de cero voltios y elevarla hasta 1,65 V. El valor no es al azar y resulta que la mitad de 3,3 V es 1,65 V.

El funcionamiento es muy simple, si un voltaje negativo llega a esta etapa se le sumarán a ese voltaje 1,65 V, en otras palabras el voltaje negativo máximo que se puede ingresar en esta etapa siempre será -1,65 V, así, cuando

se le sumen los 1,65 V el valor obtenido será cero o un valor superior a cero, el cual coincide con los voltajes que puede leer el microcontrolador. Debe notarse que la referencia positiva también es 1,65 V porque al sumarle el 1,65 V que agrega el sumador de voltajes la salida es 3,3 V, valor máximo soportado por el microcontrolador. En otras palabras, se cambia el valor de referencia para que la herramienta pueda interpretar valores negativos.

Figura 4. **Diseño del sumador de voltaje**



Fuente: elaboración propia, empleando Proteus 8.

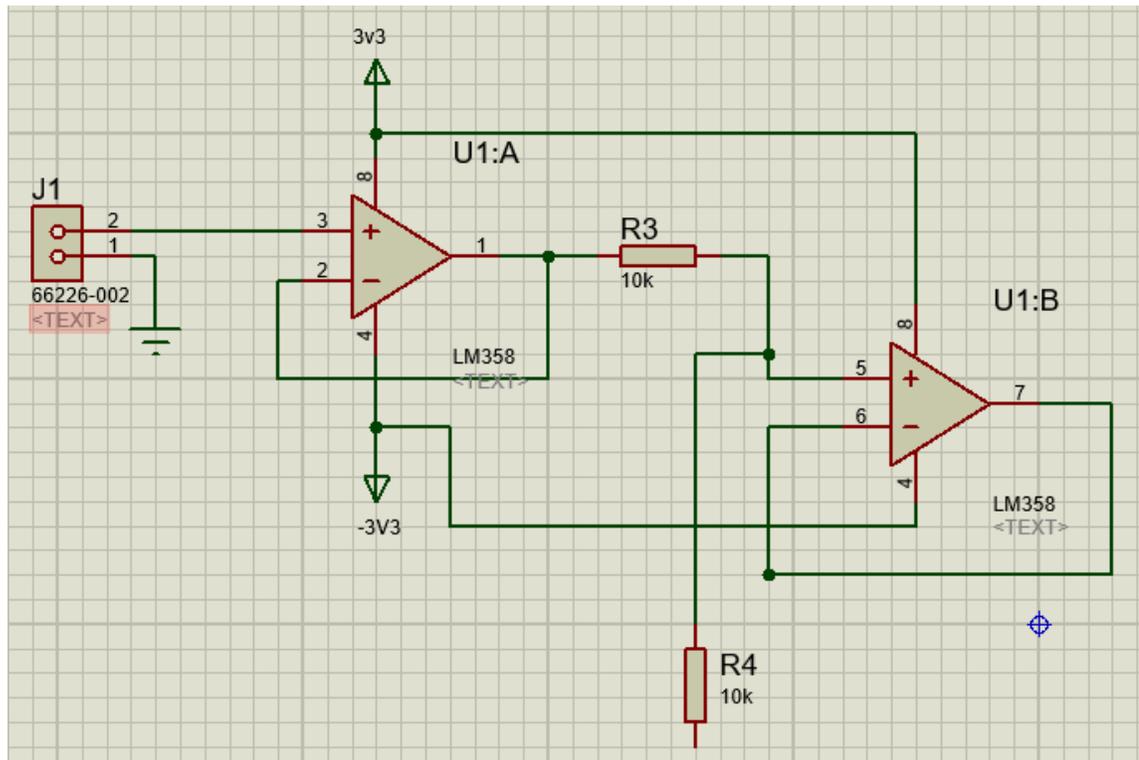
En la figura 4 se utiliza el amplificador operacional U2:A como un sumador de voltaje, J2 consiste en un puente que conecta R1 y R2, como las dos tienen el mismo valor el valor de voltaje que entregan a la entrada no inversora (pin 3) del amplificador, el voltaje será de la mitad de la fuente de alimentación, en este

caso 3,3 V. El amplificador operacional en la salida (pin 1) entregará el valor recibido en su entrada no inversora. Finalmente se conecta la salida a la entrada inversora (pin 2) para eliminar el ruido que pueda ingresar al sistema.

2.1.3. Diseño del mezclador de voltaje

Es necesario un mezclador de voltaje para unir señales, previo a procesarlas digitalmente. Para la realización del diseño esquemático se empleó el entorno integrada de desarrollo Proteus 8.

Figura 5. Diseño del mezclador de voltaje



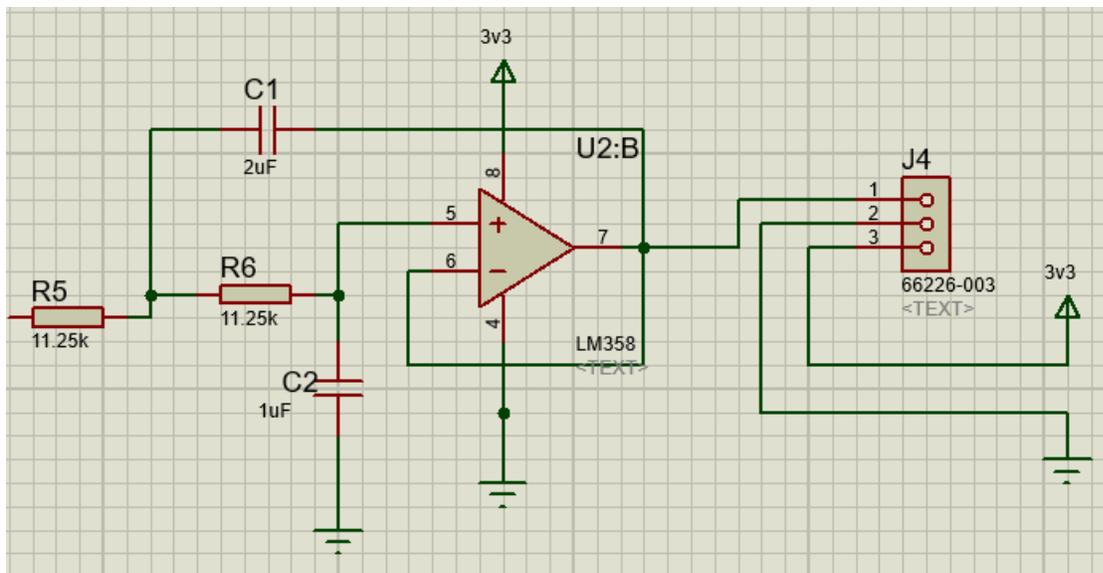
Fuente: elaboración propia, empleando Proteus 8.

En la figura 5 se muestra el diseño del mezclador de voltaje, U1:A es un seguidor de voltaje utilizado para aislar las entradas y proteger los circuitos posteriores a este, no modifica la señal de entrada proveniente del pin 2 de J1. J1 es un conector que trae la señal del divisor de voltaje. U1:B es un amplificador operacional, está configurado como mezclador de voltaje, sirve para unir la señal proveniente del seguidor de voltaje U1:A y del sumador de voltaje U1:A y del sumador de voltaje U1:B mostrado en la figura 4. Las resistencias R3 y R4 son necesarias para recibir las señales y mezclarlas en esta configuración del amplificador operacional.

2.1.4. Diseño del filtro pasa bajo

Creado en Proteus 8. Es un circuito necesario para limitar la frecuencia que ingresa en el microcontrolador, evita posibles daños por frecuencias elevadas y lecturas erróneas.

Figura 6. Filtro pasa bajos



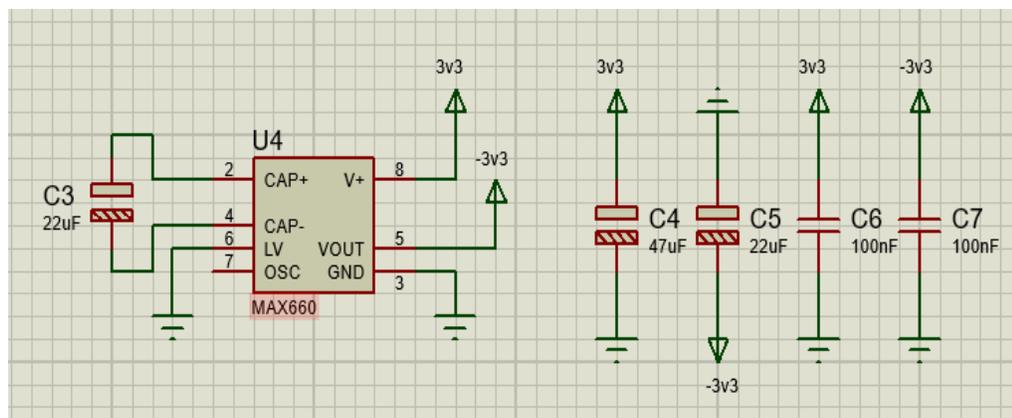
Fuente: elaboración propia, empleando Proteus 8.

Un microcontrolador solo puede captar muestras de señal que cumplen un determinado rango de velocidad, si sobrepasa la velocidad máxima con que puede obtener muestras comenzará a regenerar señales erróneas, para evitar la mala toma de muestras de la señal se utiliza un filtro de paso bajo, el cual discriminará las frecuencias mayores a 10 KHz.

2.1.5. Fuente de alimentación de los amplificadores operaciones

Todo circuito electrónico necesita ser alimentado por una fuente de energía. En el caso de algunos circuitos integrados esta tarea se complica debido a que necesitan voltajes positivos y negativos para operar. La placa integrada de desarrollo Tiva C no proporciona voltajes negativos, por lo que es necesario un circuito adicional que convierta voltajes positivos en negativos.

Figura 7. Convertidor de voltaje

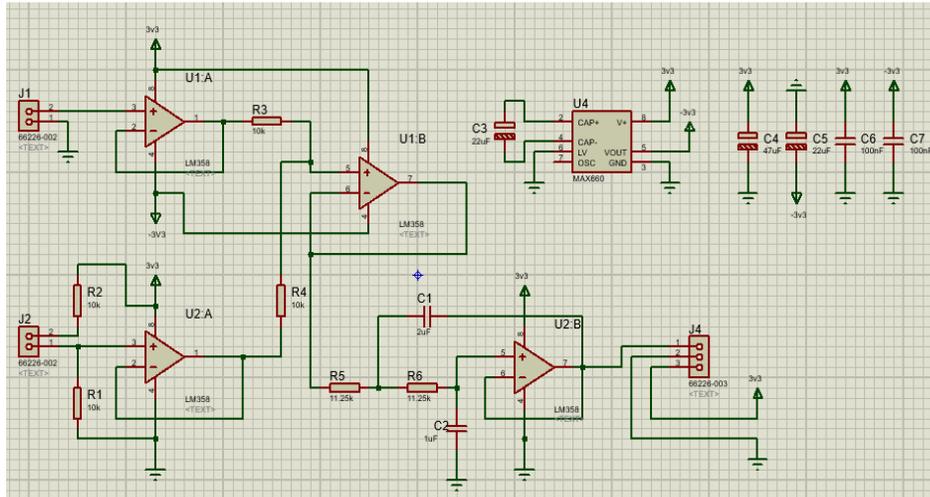


Fuente: elaboración propia, empleando Proteus 8.

La última etapa de tratamiento de señal es un convertidor de voltaje, el cual se muestra en la figura 7 y consiste en un circuito integrado especial (MAX660) que recibe como entrada un voltaje de 3,3 V, en su salida el integrado invierte el voltaje y entrega -3.3V. En este caso se utiliza para poder dar alimentación a los amplificadores operacionales, ya que ellos requieren tanto voltajes positivos como negativos y el microcontrolador que se utiliza como fuente de alimentación para todos los circuitos solo proporciona voltajes positivos. Para que este circuito integrado funcione necesita cinco capacitores, en la figura enumerados de C3 A C7 y conectados a las fuentes necesarias. Debe notarse que algunos capacitores se conectan a salida del MAX660 (pin 5).

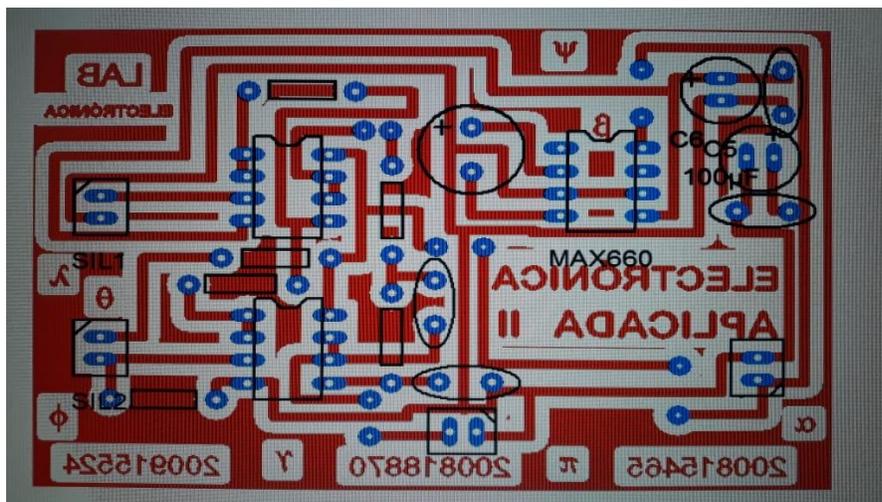
En esta parte no hay mucho que agregar, cada etapa se explicó por separado, se opta por colocar estos diseños en la misma placa para ahorrar espacio. En la figura 8 el diagrama esquemático unificado, en la figura 9 el diseño PCB y en la figura 10 la foto de la placa final ya impresa y con sus componentes soldados.

Figura 8. **Diseño unificado del elevador de voltaje, sumador, filtro y convertidor de voltaje**



Fuente: elaboración propia, empleando Proteus 8.

Figura 9. **Diagrama PCB de los diseños con amplificadores operacionales**



Fuente: elaboración propia, empleando Livewire.

Figura 10. **Placa de amplificadores operacionales**

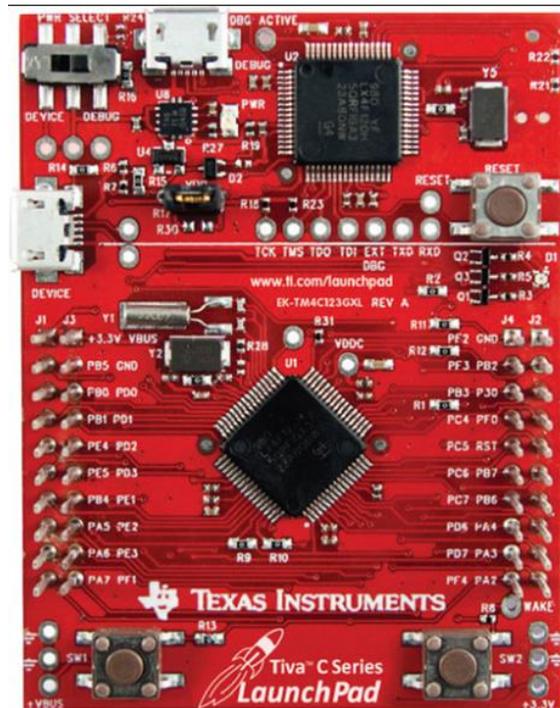


Fuente: elaboración propia.

2.2. **Microcontrolador Tiva C**

Son un tipo de microcontroladores autónomos de bajo costo de una sola placa. Son pequeños, cuentan con una CPU ARM Cortex-M4F de 32 bits que funciona a 80 a 120 MHz, fabricada por Texas Instruments. Una de las características más importantes es que todo está embebido en la tarjeta, debe recordarse que antiguamente a los microcontroladores era necesario crearles una placa adicional donde se conectarían y la placa debía contar con todo lo necesario para que funcionara. Las Tiva C ya están preparadas para funcionar, cuentan con dos puertos micro USB tipo B, el primero sirve como fuente de alimentación, mientras que el segundo es para conectar a una computadora y poder cargarle programación. Cuenta con 40 pines de entrada y salida, los cuales, dependiendo de la capacidad del dispositivo, pueden configurarse de varias maneras.

Figura 11. Tiva C



Fuente: *Tiva C series*. <https://www.ebay.co.uk/itm/Tiva-C-Series-LaunchPad-Evaluation-Kit-Texas-Instruments-EK-TM4C123GXL-/142880507935>. Consulta: 8 de marzo de 2019.

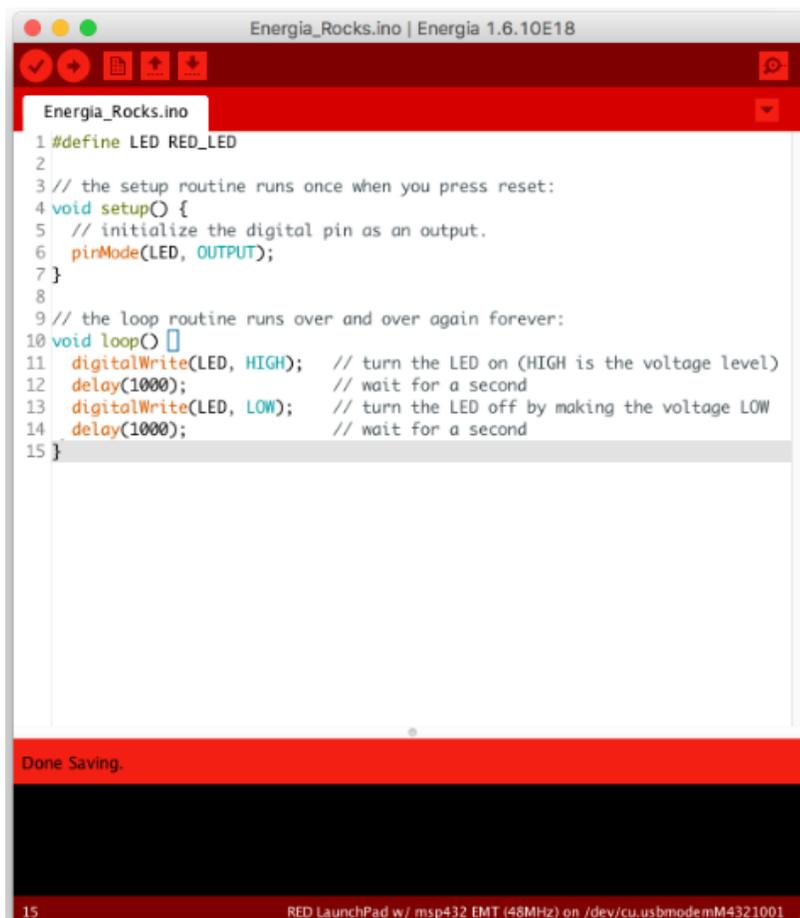
2.2.1. Programación de Tiva C con programa Energia

Energia es un entorno de programación de código abierto utilizado para cargar y desarrollar programas para los microcontroladores Texas Instruments, es en realidad una versión de Arduino para esta plataforma de microcontroladores, resulta ser fácil de utilizar y se asemeja a los lenguajes de programación de alto nivel como C# o Java.

Energia es de código abierto y cualquier programador puede contribuir en su desarrollo. Energia se puede cargar gratuitamente desde su página oficial, al descargar no será necesario instalar ya que es ejecutable y solo basta con hacer doble *click* en su icono y el programa iniciará automáticamente.

- Página oficial de Energia: <http://energia.nu/>

Figura 12. Interfaz gráfica de Energia



```
Energia_Rocks.ino
1 #define LED RED_LED
2
3 // the setup routine runs once when you press reset:
4 void setup() {
5   // initialize the digital pin as an output.
6   pinMode(LED, OUTPUT);
7 }
8
9 // the loop routine runs over and over again forever:
10 void loop() {
11   digitalWrite(LED, HIGH); // turn the LED on (HIGH is the voltage level)
12   delay(1000); // wait for a second
13   digitalWrite(LED, LOW); // turn the LED off by making the voltage LOW
14   delay(1000); // wait for a second
15 }
```

Done Saving.

15 RED LaunchPad w/ msp432 EMT (48MHz) on /dev/cu.usbmodemM4321001

Fuente: *Energia*. <http://energia.nu/>. Consulta: 1 de abril 2019.

2.2.2. Código fuente del proyecto en Energia

Después de que las placas de tratamiento de señal se encargaran de la parte de la electrónica analógica comienza la parte de digitalizar la señal, es en este punto donde se necesita del microcontrolador y la respectiva programación necesaria para que el microcontrolador sepa qué hacer con la señal.

El código fuente en Energia es sencillo y consta de tres partes. La primera es donde se declaran las variables, las cuales son dos, una para capturar la señal y la otra donde se guarda el valor de la señal en forma de bits, en otras palabras, es el valor de la señal en un instante de tiempo en forma digital.

En la segunda parte del programa solo se declara a qué valor de transmisión se quiere configurar el puerto serial, el estándar es 9600 baudios y en ese valor se deja. Finalmente, la tercera parte es un ciclo que se repetirá infinitamente, en esta parte se escribe todo lo que se quiere repetir varias veces, es en esta parte donde se captura la señal, se digitaliza y posteriormente se manda por el puerto UART del microcontrolador a un puerto USB del computador.

Figura 13. Código fuente en Energia

```
//Declaracion de variables
int lectura = A7; //Recibe la señal
int val = 0;      //Se escribe el valor de la señal digitalizado

void setup()
{
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop()
{
  // put your main code here, to run repeatedly:
  analogReadResolution(6); //resolucion 2 bits elevados a la 6
  val = analogRead(lectura); //se escribe la señal atrapada en lectura
                               //en forma de bits digitalizados
  Serial.print(val);        //la señal digitalizada se trasmite
                               //por UART a un puerto USB del Computador
  delay(10);               //Se esperan 0.2Segundos para repetir
}
```

Fuente: elaboración propia, empleando Energia.

2.3. Entorno Matlab

Como ya se mencionó anteriormente, Matlab es un entorno de desarrollo integrado, útil para sistemas numéricos y matemáticos. Matlab ofrece diversas herramientas como un lenguaje propio de programación, llamado lenguaje M, es en este lenguaje que se escribe el código fuente que llevará a cabo la tarea de recibir la señal digitalizada del microcontrolador y graficar la señal en la pantalla del ordenador.

2.3.1. Lenguaje M

Es un lenguaje de programación propio del entorno de desarrollo matemático Matlab. Matlab es muy funcional, es decir al querer resolver una operación matemática es necesario escribir un comando y el entorno responderá con un resultado, careciendo de interfaz gráfica, para eso se creó el lenguaje M, en que se puede escribir programas capaces de levantar interfaces gráficas y mostrar de una manera más agradable los resultados de operaciones matemáticas que realice Matlab.

Al procesar una señal digital y posteriormente graficarla se necesita de ciertos cálculos matemáticos para saber en qué momento se necesita graficar, en qué punto exacto de la pantalla del ordenador, entre otros cálculos, si bien estos cálculos no es necesario mostrarlos en pantalla, pero sí es necesario mostrar la gráfica, para ello se utiliza el lenguaje M, en este caso, para graficar los cálculos matemáticos realizados por Matlab.

2.3.2. Código fuente en lenguaje M

A continuación, se explica cada una de las partes del código fuente del programa en lenguaje M de Matlab, el cual es el encargado de graficar la señal digitalizada entregada por el microcontrolador por uno de los puertos USB del ordenador.

Figura 14. **Función para establecer comunicación serial y declaración de variables globales**

```
function proyecto_apli2
%para borrar instancias previas
delete(instrfind({'Port'},{'COM3'}));
%crea objeto serial
s = serial('COM3','BaudRate',9600,'Terminator','CR/LF');
warning('off','MATLAB:serial:fscanf:unsuccessfulRead');
fopen(s);

%variables
tmax = 200;
rate = 35;
yn=1; %valor inicial de escala en Y
ys=0; %valor obtenido de Y de lectura serial
xn=1; %valor inicial de escala en X
xs=0; %valor obtenido de X de lectura serial
x1 = 0; %valor inicial de X en coordenadas
x2 = 300; %ancho de X en coordenadas
y1 = -5; %valor en coordenada más bajo de la ventana
y2 = 5; %valor en coordenada más alto de la ventana
```

Fuente: elaboración propia, empleando lenguaje M de Matlab.

La primera de todo programa independientemente del lenguaje de programación utilizado es la de crear una función o programa principal, dentro de este se declaran las variables y funciones que serán necesarias, como se muestra en la figura 14.

Figura 15. **Creación de la pantalla que se utilizará para graficar**

```
%prepara el area que se utilizara para graficar
f = figure('Name','Capture','position',[360,200,650,485]); %dimensiones del lienzo en la pantalla
a = axes('Xlim',[x1+xs x2+xs], 'YLim',[(y1+ys)/yn (y2+ys)/yn],'position',[0.1,0.1,0.7,0.8]);
ll = line(nan,nan,'Color','r','LineWidth',2);

xlabel('Tiempo (s)');
ylabel('Voltaje (v)');
title('Osciloscopio');
grid on
hold on
```

Fuente: elaboración propia, empleando lenguaje M de Matlab.

Entre las funciones especiales que se utilizan está la función `figure`, que es la que crea la ventana, y la función `axes`, con la cual se crean los ejes de graficación dentro de la ventana. Estas funciones se muestran en la figura 15. De la figura 16 a la figura 19 se muestra el código utilizado para crear botones dentro de la ventana.

Figura 16. **Botón para agrandar gráfica en eje Y**

```
%agrandar la grafica en Y
texto = uicontrol('Style','text',...
                 'String','Vertical','Position',[530,430,70,25]);

escala_vertical_up = uicontrol('Style','pushbutton',...
                              'String','V up','Position',[530,410,70,25],...
                              'Callback',@v_up_Callback);

escala_vertical_down = uicontrol('Style','pushbutton',...
                                 'String','V down','Position',[530,380,70,25],...
                                 'Callback',@v_down_Callback);
```

Fuente: elaboración propia, empleando lenguaje M de Matlab.

Figura 17. **Botón para mover la gráfica en eje Y**

```
%mover grafica en y
texto2 = uicontrol('Style','text',...
    'String','Subir/Bajar','Position',[530,340,70,25]);

sube = uicontrol('Style','pushbutton',...
    'String','V up','Position',[530,320,70,25],...
    'Callback',@sube_Callback);

baja = uicontrol('Style','pushbutton',...
    'String','V down','Position',[530,290,70,25],...
    'Callback',@baja_Callback);
```

Fuente: elaboración propia, empleando lenguaje M de Matlab.

Figura 18. **Botón para agrandar la gráfica en eje X**

```
%agrandar la grafica en x
texto3 = uicontrol('Style','text',...
    'String','Horizontal','Position',[530,250,70,25]);

escala_horizontal_up = uicontrol('Style','pushbutton',...
    'String','H up','Position',[530,230,70,25],...
    'Callback',@h_up_Callback);

escala_horizontal_down = uicontrol('Style','pushbutton',...
    'String','H down','Position',[530,200,70,25],...
    'Callback',@h_down_Callback);
```

Fuente: elaboración propia, empleando lenguaje M de Matlab.

Figura 19. **Botón para mover la gráfica en eje X**

```
%mover en x
texto4 = uicontrol('Style','text',...
    'String','Mover en X','Position',[530,160,70,25]);

izquierda = uicontrol('Style','pushbutton',...
    'String','<','Position',[530,140,35,25],...
    'Callback',@iz_Callback);

derecha = uicontrol('Style','pushbutton',...
    'String','>','Position',[570,140,35,25],...
    'Callback',@dr_Callback);
```

Fuente: elaboración propia, empleando lenguaje M de Matlab.

Figura 20. **Creación de una matriz de ceros, variable i de iteraciones en eje Y, variable t para iteraciones en eje X**

```
%inicializar
v1 = zeros(1,tmax*rate);
i = 1;
t = 0;
```

Fuente: elaboración propia, empleando lenguaje M de Matlab.

En la figura 20 se muestra la declaración de variables locales, la primera es la matriz $v1$, que será llenada de ceros con dimensiones de 1 por $(tmax \text{ por } rate)$, donde $tmax$ que es una variable global que indica que se tomarán 200 muestras por $rate$, otra variable global, su valor es de 35, es decir 200 muestras por 35 que da igual a 7 mil muestras que serán utilizadas para reconstruir la señal.

Figura 21. Ciclo *while*

```
]while t<tmax
    t = toc;
    %leer del puerto serial
    b = fscanf(s,'%d',2);
    v1(i)=b*3.3*yn/63;
    fprintf('valor %f de variable', v1(i));
    %dibuja en la figura
    if i >= 299
        x2=x2+1;
        x1=x1+1;
    end
    %a = axes('Xlim',[x1 x2], 'YLim',[-1 5.1]);
    axis([x1+xs,x2+xs+2, (y1+ys)/yn, (y2+ys)/yn]);
    x = linspace(0,i*xn,i);
    set(ll,'YData',v1(1:i),'XData',x);
    drawnow
    %seguir
    i = i+1;
```

Fuente: elaboración propia, empleando lenguaje M de Matlab.

En la figura 21 se muestra un ciclo que se ejecuta desde que la variable *t* sea igual a cero hasta que alcance un valor menor a *tmax*, cuyo valor es de 200. El ciclo sirve para determinar cuánto tiempo se va a ejecutar el programa, el cual es de 200 muestras en el eje del tiempo, es decir en eje X. El ciclo también sirve para llenar la matriz *v1* con la muestra obtenida del puerto USB de la computadora, en este mismo ciclo se grafica cada valor de *v1* en la pantalla con la ayuda del comando *axis*.

Figura 22. **Imprimir en pantalla**

```
%resultado del cronometro
clc;
fprintf('%g s de captura a %g cap/s \n',t,i/t);

%Limpiar el puerto
fclose(s);
delete(s);
clear s;
%end
```

Fuente: elaboración propia, empleando lenguaje M de Matlab.

La función `fprintf` mostrada en la figura 22 puede ser descartada, es más bien una función que sirve para graficar en la consola de Matlab, no en la pantalla de graficación, el valor capturado por el puerto serial o puerto USB de la computadora más bien sirve para asegurar que en el puerto se estén depositando los datos. La función `fclose` sirve para cerrar la función `fprintf` una vez se deja de usar, el comando `delete` borra el último valor `fprintf` y finalmente `clear` limpia la consola de Matlab.

De la figura 16 a la figura 19 se muestra la programación de los botones de la pantalla, pero estos botones por si solos no son funcionales, es decir no hacen nada, solamente adornan la pantalla, es por lo que necesitan de una función de llamada en cada botón y en la figura 23 se muestra su programación.

Figura 23. **Funciones de los botones**

```
]function v_up_Callback(source,eventdata)
    yn=yn+1;
-end
]function v_down_Callback(source,eventdata)
    yn=yn-1;
-end
]function sube_Callback(source,eventdata)
    ys=ys+1;
-end
]function baja_Callback(source,eventdata)
    ys=ys-1;
-end
]function h_up_Callback(source,eventdata)
    xn=xn+1;
-end
]function h_down_Callback(source,eventdata)
    xn=xn-1;
-end
]function iz_Callback(source,eventdata)
    xs=xs+10;
-end
]function dr_Callback(source,eventdata) |
    xs=xs-10;
-end
```

Fuente: elaboración propia, empleando lenguaje M de Matlab.

La primera función `v_up_Callback` mueve la gráfica en el eje X hacia arriba, `v_down_Callback` mueve la gráfica hacia abajo en el mismo eje, `sube_Callback` y `baja_Callback` mueve la ventana de graficación hacia arriba y hacia abajo, respectivamente. Las funciones `h_up_Callback` y `h_down_Callback` mueven la gráfica hacia la izquierda y hacia la derecha, respectivamente. Finalmente, las funciones `iz_callback` y `dr_callback` mueven la pantalla hacia la izquierda y a la derecha respectivamente⁸.

⁸ Geek Electrónica. *Recepción de datos por puerto serial y graficación en Matlab*. <https://geekelectronica.com/recepcion-de-datos-por-puerto-serial-y-graficacion-en-matlab/>. Consulta: abril 2019.

3. ESPECIFICACIONES TÉCNICAS

En el capítulo anterior se explicaron los diseños del osciloscopio propuesto en este trabajo, pero no se explicó la interacción entre los diseños de los circuitos, el microcontrolador y el ordenador. Este capítulo debe tomarse como una guía para saber cómo debe conectarse los diseños ya implementados.

3.1. Interconexión del divisor de voltaje

Esta quizá sea la conexión más sencilla para poder medir una señal, los osciloscopios utilizan puntas de osciloscopio conectados a un conector BNC, este conector se va a interconectar a su vez con la placa impresa del divisor de voltaje.

Figura 24. Punta de osciloscopio



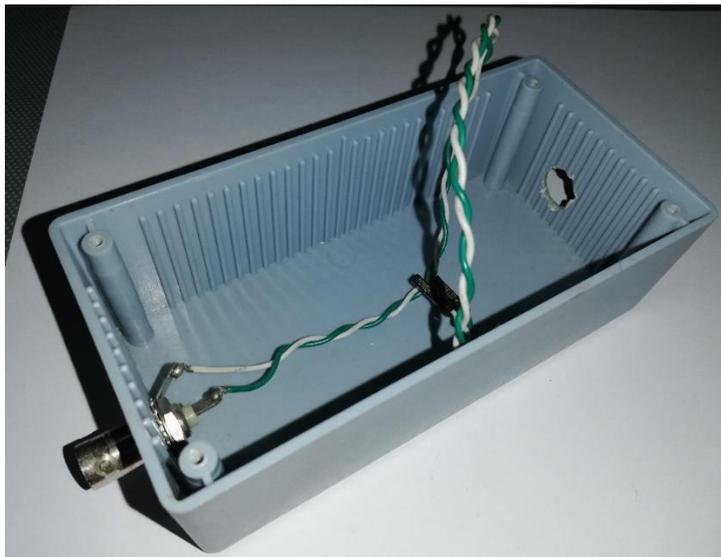
Fuente: elaboración propia.

Figura 25. **Conector BNC**



Fuente: elaboración propia.

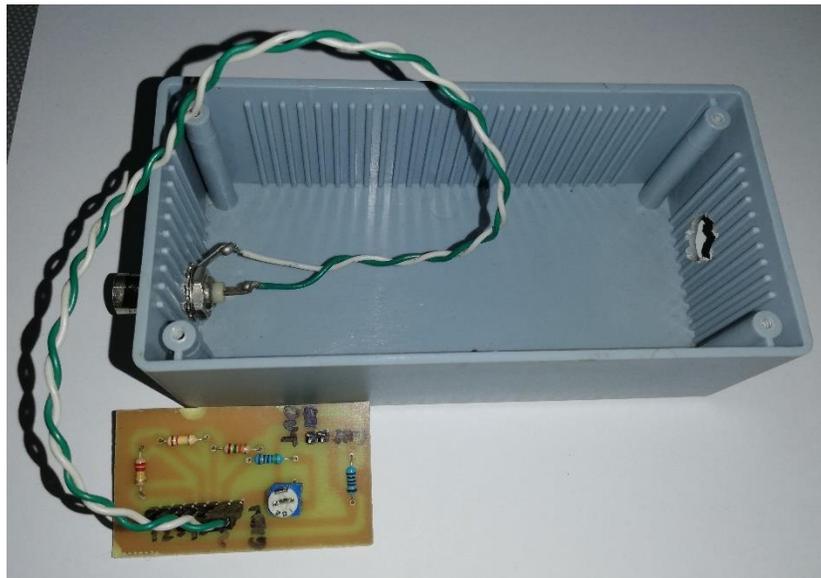
Figura 26. **Conecto BNC en la caja de conexión**



Fuente: elaboración propia.

En la figura 26 se observa el conector BNC en una caja de conexión eléctrica de plástico donde se instaló el conector y se le soldaron dos cables que irán conectados a la placa del divisor, como se muestra en la figura 27.

Figura 27. **Conecto BNC y placa del divisor de voltaje**

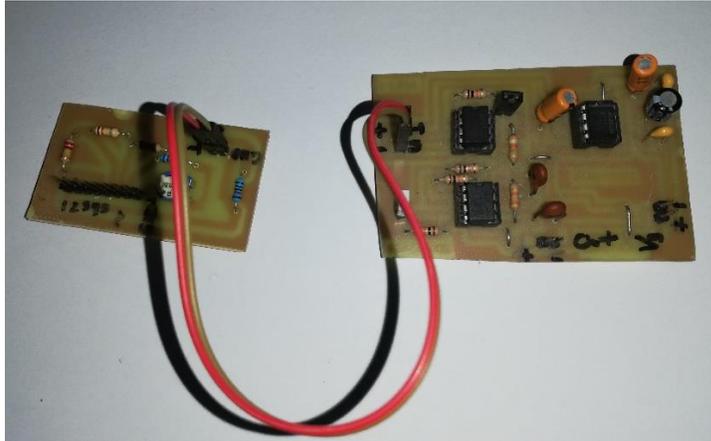


fuelle: elaboración propia.

3.2. Conexión entre el divisor de voltaje y los amplificadores operacionales

En la figura 28 se muestran las conexiones entre la placa del divisor de voltaje y la placa que contiene el sumador de voltaje, el filtro y el mezclador de voltajes que se encarga de tratar de señal analógica, para evitar que al microcontrolador lleguen voltajes negativos.

Figura 28. **Conexión de placa de divisor de voltaje y placa de amplificadores operacionales**



Fuente: elaboración propia.

3.3. Microcontrolador y placas de tratamiento de señal analógica

La conexión entre la placa de amplificadores operacionales necesita dos pares de cables, el primero rojo y negro, son la alimentación de la primera placa, ya que es de la placa del microcontrolador que se alimenta a los demás circuitos, y los cables naranja y azul son los que llevan la señal a ser analizada, cable naranja positivo y cable azul negativo (figura 29).

Figura 29. **Conexión entre placa de amplificadores operacionales y microcontrolador Tiva C**



Fuente: elaboración propia.

Es importante notar que el cable de señal no va conectado a cualquier pin del microcontrolador Tiva C, este se debe conectar al puerto analógico A7 porque fue de esta manera que se declaró el programa en el microcontrolador, en la figura 30 se muestra esta programación.

Figura 30. **Declaración de puerto A7 en Energia**

```
int lectura = A7; //Recibe la señal
```

fuentes elaboración propia, empleando Energia.

Figura 31. **Todas las placas acomodadas**



Fuente: elaboración propia.

3.4. Ensamblaje final

En la figura 31 se muestra cómo quedan todas las placas acomodadas dentro de la caja de plástico de conexión. Finalmente, en la figura 32 la caja de conexiones cerrada forma el final de esta etapa, como último paso solo quedaría conectar el cable USB que sale del microcontrolador a un puerto USB de un ordenador que tenga instalado Matlab y la aplicación desarrollada para graficar.

Figura 32. **Vista del proyecto ensamblado**



Fuente: elaboración propia.

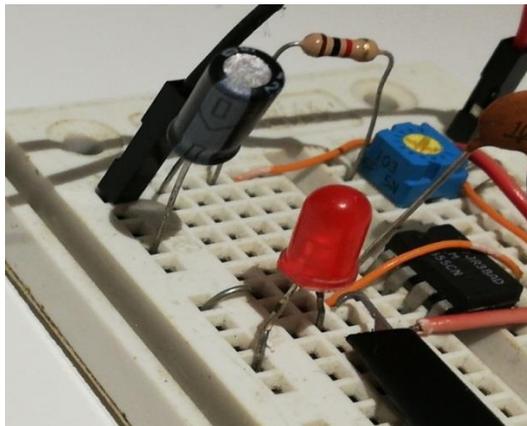
4. ESPECIFICACIONES DE USO

El capítulo anterior se trató de cómo se conectan todos los diseños implementados y cómo interactuarán las distintas partes entre sí, aún falta explicar cómo funciona la herramienta. Se podría decir que saber utilizarla se divide en colocación de la punta de osciloscopio, selección de la escala o rango de voltaje de entrada y utilización de la aplicación gráfica.

4.1. Colocación de la punta de osciloscopio

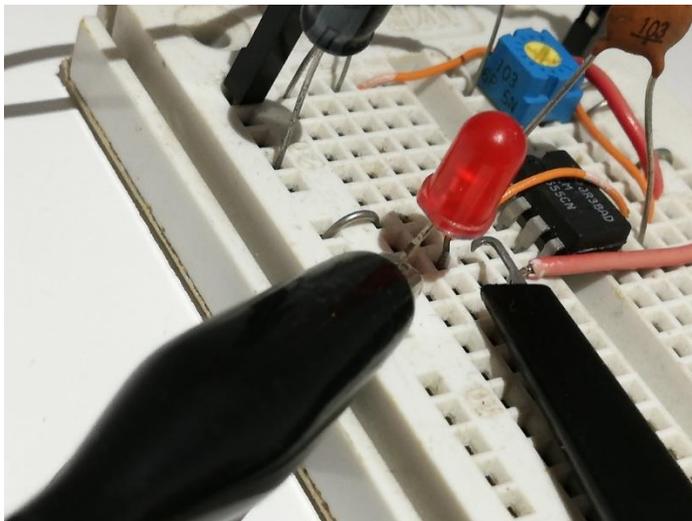
Es la parte más sencilla, al tener una señal que se desea graficar se debe colocar el gancho de la punta de osciloscopio en el cable que porta la señal, puede ser también en el lugar donde se encuentra la señal, y el lagarto de la punta de osciloscopio se debe referenciar a la tierra del sistema.

Figura 33. Colocación de punta de osciloscopio



Fuente: elaboración propia.

Figura 34. **Conexión a tierra de la punta de osciloscopio**



Fuente: elaboración propia.

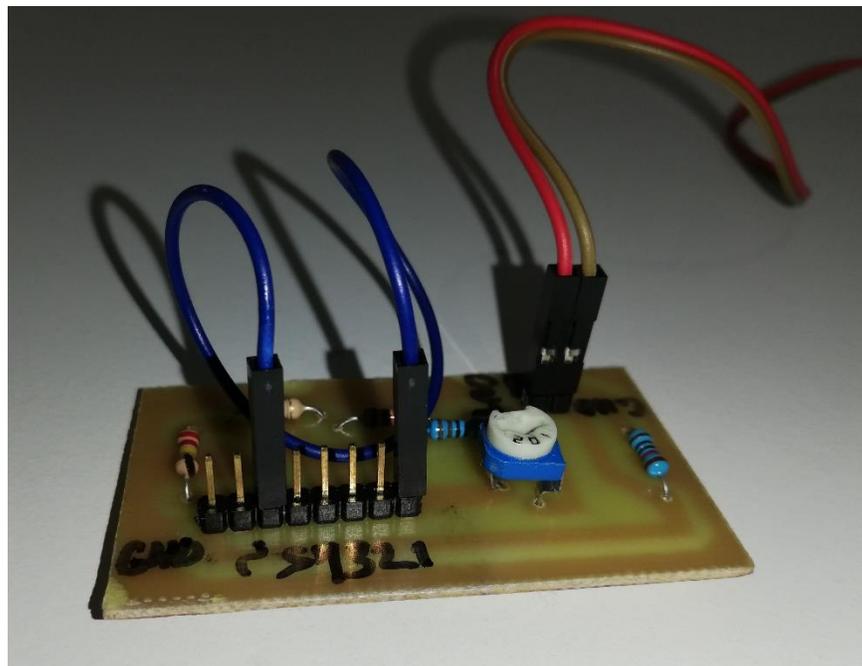
4.2. **Selección de la escala**

Como se mencionó en el capítulo 2, la herramienta como en un multímetro se diseñó con 5 escalas o rangos para poder medir una señal y ser más preciso, la escala 1 es utilizada para que medie los valores de voltaje más bajos, prácticamente es una conexión directa. La escala 5 es la escala para medir voltajes más elevados. A continuación se presentan las escalas implementadas:

- Escala 1: +/- 1,65 V
- Escala 2: +/- 2,5 V
- Escala 3: +/- 5 V
- Escala 4: +/- 25 V
- Escala 5: +/- 62,5V

Como siguiente paso después de colocar la punta de osciloscopio se debe seleccionar la escala, si se utiliza un generador de onda cuadrada como ejemplo, que en su salida genera un pulso que va de 0 a 1,25 V, entonces se debe seleccionar la escala 1 como se muestra en la figura 34.

Figura 35. **Selección de escala**



Fuente: elaboración propia.

4.3. Funcionamiento y uso de la aplicación gráfica

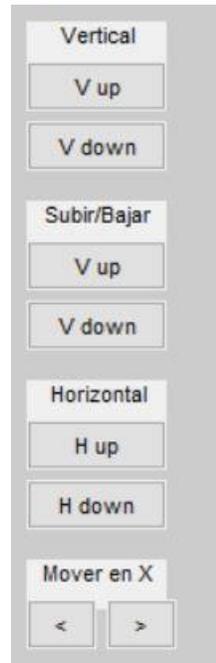
Con la colocación de la punta de osciloscopio y la selección del rango se concluye con la colocación física del osciloscopio, ahora es necesario saber cómo se utiliza la herramienta gráfica o aplicación del programa en Matlab. Cabe mencionar que esta herramienta gráfica cuenta con unos botones que son utilizados para mover la gráfica, agrandar y reducir la señal.

4.3.1. Botones

La aplicación cuenta con ocho botones, como se muestra en la figura 35 los botones están divididos en cuatro secciones, cada botón está en pareja con otro botón, las secciones y las parejas son las siguientes:

- Sección vertical, con botón V up y V down: esta sección sirve para agrandar o reducir la gráfica solo en el eje vertical o eje Y del plano cartesiano.
- Sección subir/bajar, con botones V up y V down: utilizados para mover la gráfica para arriba o para abajo en el eje Y.
- Sección horizontal, con botón H up y H down: sirven para hacer más grande la gráfica o más pequeña en el eje de las X del plano cartesiano.
- Sección mover en X, con botones a izquierda y derecha: como el nombre de sección indica, sirven para mover la gráfica de izquierda a derecha.

Figura 36. **Botones de la aplicación**

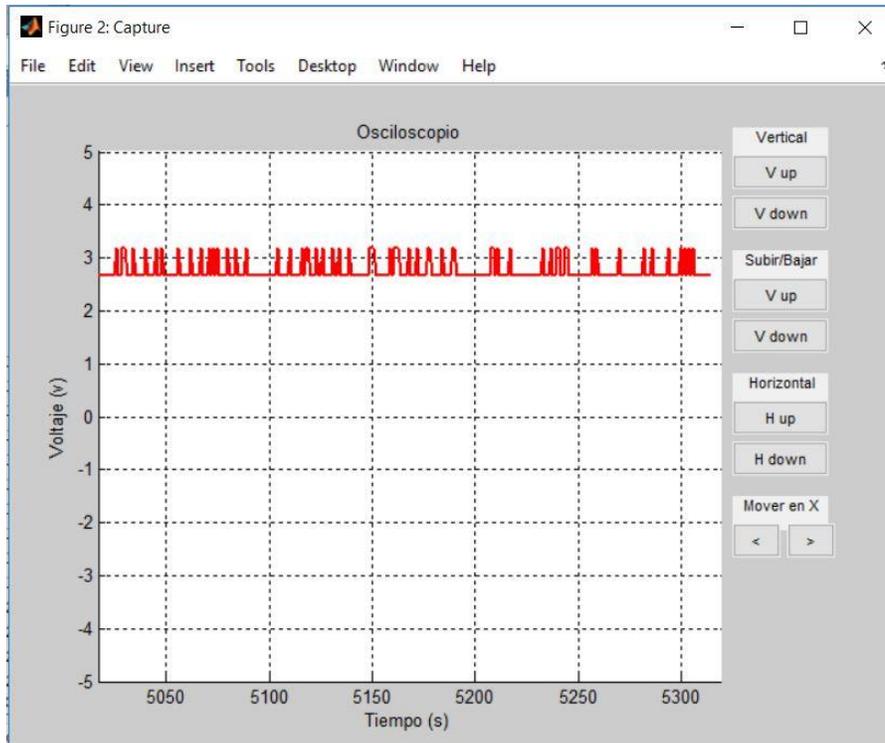


Fuente: elaboración propia, empleando lenguaje M de Matlab.

4.3.2. Área de graficación

Al saber cómo funcionan los botones no hay mucho que agregar, más que mostrar el área donde se despliega la gráfica, en la figura 36 se muestra la aplicación graficando una señal, para que esta aplicación se ejecute hay que correr el programa en Matlab.

Figura 37. Gráfica de una señal en la aplicación



Fuente: elaboración propia, empleando lenguaje M de Matlab.

El orden de los pasos a seguir para utilizar el osciloscopio no necesariamente debe ser como el que se explicó en este capítulo, puede ejecutarse la aplicación y luego seleccionar la escala y por último colocar la punta de osciloscopio. Lo que sí es recomendable es seleccionar la escala previo a conectar la punta, ya que de lo contrario se corre el riesgo de quemar el microcontrolador.

CONCLUSIONES

1. Es necesario limitar la velocidad de muestreo del ADC del microcontrolador para evitar que se grafique la señal con errores, esto debido a que al reconstruir las señales por el lenguaje M de Matlab se grafican lento.
2. El divisor de voltaje es un teorema de diseño que se utiliza para obtener voltajes deseados en una resistencia que puede servir como referencia de salida. Si bien es una herramienta muy popular en el diseño de circuitos electrónicos, en la práctica agrega demasiado ruido a una señal.
3. La digitalización de una señal se lleva a cabo por el ADC del microcontrolador Tiva C, esto consiste en convertir la señal continua en el tiempo a discreta en el tiempo. Para ello se divide la señal en muestras que son codificadas y transmitidas por medio de protocolo UART hacia un ordenador.
4. Matlab es un sistema numérico integrado que cuenta con un lenguaje de programación propio, llamado lenguaje M, el cual cuenta con una librería especial capaz de recibir y enviar datos por los puertos USB de una computadora con protocolos UART.

RECOMENDACIONES

1. Al utilizar un ordenador para realizar la tarea de graficar la señal se presenta el problema de la lentitud, para evitarlo se puede emplear una pantalla LCD conectada directamente al microcontrolador. Aunado a esto se debe considerar que el microcontrolador, al tener un dispositivo extra conectado, podría funcionar más lento.
2. Para reducir el ruido que agregan los divisores de voltaje a una señal se pueden usar otras técnicas de reducción de voltaje como los transformadores, estos reducen el voltaje por medio de inducción magnética y no dividiéndolo como es el caso de los divisores.
3. El ADC integrado de la Tiva C tiene una velocidad de reloj de 80Mhz, muy superior a varios ADC externos. Si se desea mejorar la velocidad de muestro es posible implementar un ADC externo y manipularlo con un microcontrolador, ya sea Tiva C o algún otro.
4. El lenguaje M de Matlab cuenta con una librería capaz de recibir datos por los puertos USB de un ordenador, sin embargo, este lenguaje presenta el problema de ser muy lento y retardar las muestras, para evitar esto se puede programar una nueva aplicación en otro lenguaje de programación que también cuente con librerías USB como lo es Python.

BIBLIOGRAFÍA

1. BOYLESTAD, Robert L. *Introducción al análisis de circuitos*. 10a ed. México: Pearson Educación, 2004. 1248 p.
2. COUCH, Leon. *Sistemas de comunicación digitales y analógicos*. 7a ed. México: Pearson Educación, 2008. 784 p.
3. FLOYD, Thomas L. *Fundamentos de sistemas digitales*. 9a ed. Madrid, España: Pearson Educación, 2006. 1024 p.
4. Geek Electrónica. *Recepción de datos por puerto serial y graficación en Matlab*. ©2019. [en línea]. <<https://geekelectronica.com/recepcion-de-datos-por-puerto-serial-y-graficacion-en-matlab/>>. [Consulta: 21 de abril de 2019].
5. HELFRICK, Albert D.; COOPER, William D. *Instrumentación eléctrica moderna y técnicas de medición*. México: Prentice Hall Hispanoamericana, 1991. 450 p.
6. MALVINO, Albert Paul. *Principios de Electrónica*. 6a ed. Madrid, España: McGraw-Hill, 2000. 1125 p.
7. SEDRA, Adel S.; SMITH, Kenneth C. *Circuitos microelectrónicos*. 4a ed. México: Litográfica Eros, 2002. 1355 p.

8. TOMASI, Weyne. *Sistemas de comunicaciones electrónicas*. 4a ed.
México: Prentice-Hall INC, 2003. 976 p.

APÉNDICES

Apéndice 1. **Instalación de controladores no firmados en Windows 10**

Si se desea utilizar Tiva C en Windows 10 hay que instalar los *driver* o controladores de la placa. En Windows 10, a diferencia de otras versiones de Windows como Windows 7, se presenta una restricción de uso de controladores. Si los controladores no están firmados por Microsoft o alguno de sus socios no se pueden instalar, es por lo que hay que deshabilitar esta opción cada vez que se utilice Tiva C o alguna otra placa de desarrollo no creada por Microsoft.

- Paso 1: *click* en el menú inicio.
- Paso 2: *click* en configuración.
- Paso 3: actualización y seguridad.
- Paso 4: opción; recuperación.
- Paso 5: seleccionar inicio avanzado.
- Paso 6: *click* en reiniciar ahora y espera que reinicie el sistema.
- Paso 7: al iniciar no se carga la ventana de inicio de Windows, en su lugar se carga una opción menos que ofrece varias opciones, en este caso la opción a elegir será: solución de problemas.

Continuación apéndice 1.

- Paso 8: en el siguiente submenú seleccionar: opciones avanzadas.
- Paso 9: ahora seleccionar: configuraciones de inicio.
- Paso 10: *click* en reinicio.
- Paso 11: por último, teclear 7 o F7 para seleccionar la opción de instalación de controladores no firmados en Windows 10.

Aunque la instalación de los *drivers* solo se debe hacer la primera vez, sí es necesario realizar este procedimiento cada vez que se dese utilizar el microcontrolador Tiva C. Este procedimiento es aplicable a cualquier controlador no firmado, puede ser de otro microcontrolador distinto a la Tiva C o algún otro periférico.

Fuente: elaboración propia.

Apéndice 2. Código completo en lenguaje Energia para Tiva C

Texto en verde son los comentarios en el programa.

```
//Declaración de variables
int lectura = A7; //Recibe la señal
int val = 0;     //Se escribe el valor de la señal digitalizado

void setup()
{
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop()
{
  // put your main code here, to run repeatedly:
  analogReadResolution(6); //resolución 2 bits elevados a la 6
  val = analogRead(lectura); //se escribe la señal atrapada en lectura
                               //en forma de bits digitalizados
  Serial.print(val);         //la señal digitalizada se transmite
                               //por UART a un puerto USB del Computador
  delay(10);                //Se esperan 0.2Segundos para repetir
}
```

Fuente: elaboración propia, empleando Energia

Apéndice 3. Código completo en lenguaje M de Matlab

En azul palabras reservadas del lenguaje, en verde comentarios, en lila algunos parámetros de funciones.

```
function proyecto_apli2
%para borrar instancias previas
delete(instrfind({'Port'}, {'COM3'}));
%crea objeto serial
s = serial('COM3', 'BaudRate', 9600, 'Terminator', 'CR/LF');
warning('off', 'MATLAB:serial:fscanf:unsuccessfulRead');
fopen(s);

%variables
tmax = 200;
rate = 35;
yn=1;      %valor inicial de escala en Y
ys=0;      %valor obtenido de Y de lectura serial
xn=1;      %valor inicial de escala en X
xs=0;      %valor obtenido de X de lectura serial
x1 = 0;    %valor inicial de X en coordenadas
x2 = 300;  %ancho de X en coordenadas
y1 = -5;   %valor en coordenada más bajo de la ventana
y2 = 5;    %valor en coordenada más alto de la ventana

%prepara el área que se utilizara para graficar
f = figure('Name', 'Capture', 'position', [360,200,650,485]);
%dimensiones del lienzo en la pantalla
a = axes('Xlim', [x1+xs x2+xs], 'YLim', [(y1+ys)/yn
(y2+ys)/yn], 'position', [0.1,0.1,0.7,0.8]);
l1 = line(nan,nan, 'Color', 'r', 'LineWidth', 2);

xlabel('Tiempo (s)');
ylabel('Voltaje (v)');
title('Osciloscopio');
grid on
hold on

%botones
%agrandar la grafica en Y
texto = uicontrol('Style','text',...
    'String','Vertical', 'Position', [530,430,70,25]);

escala_vertical_up = uicontrol('Style','pushbutton',...
    'String','V up', 'Position', [530,410,70,25],...
    'Callback', @v_up_Callback);
```

Continuación apéndice 3.

```
escala_vertical_down = uicontrol('Style','pushbutton',...
    'String','V down','Position',[530,380,70,25],...
    'Callback',@v_down_Callback);

% mover grafica en y
texto2 = uicontrol('Style','text',...
    'String','Subir/Bajar','Position',[530,340,70,25]);

sube = uicontrol('Style','pushbutton',...
    'String','V up','Position',[530,320,70,25],...
    'Callback',@sube_Callback);

baja = uicontrol('Style','pushbutton',...
    'String','V down','Position',[530,290,70,25],...
    'Callback',@baja_Callback);

% agranda la grafica en x
texto3 = uicontrol('Style','text',...
    'String','Horizontal','Position',[530,250,70,25]);

escala_horizontal_up = uicontrol('Style','pushbutton',...
    'String','H up','Position',[530,230,70,25],...
    'Callback',@h_up_Callback);

escala_horizontal_down = uicontrol('Style','pushbutton',...
    'String','H down','Position',[530,200,70,25],...
    'Callback',@h_down_Callback);

% mover en x
texto4 = uicontrol('Style','text',...
    'String','Mover en X','Position',[530,160,70,25]);

izquierda = uicontrol('Style','pushbutton',...
    'String','<','Position',[530,140,35,25],...
    'Callback',@iz_Callback);

derecha = uicontrol('Style','pushbutton',...
    'String','>','Position',[570,140,35,25],...
    'Callback',@dr_Callback);
```

Continuación apéndice 3.

```
%inicilizar

v1 = zeros(1,tmax*rate);
i = 1;
t = 0;

%ejecutar bucles cronometrado

tic

while t<tmax
    t = toc;

    %leer del puerto serial

    b = fscanf(s, '%d', 2);
    v1(i)=b*3.3*yn/63;
    fprintf('valor %f de variable', v1(i));

    %dibuja en la figura

    if i >= 299
        x2=x2+1;
        x1=x1+1;
    end

    %a = axes('Xlim',[x1 x2], 'YLim',[-1 5.1]);

    axis([x1+xs,x2+xs+2, (y1+ys)/yn, (y2+ys)/yn]);
    x = linspace(0,i*xn,i);
    set(l1, 'YData',v1(1:i), 'XData',x);
    drawnow

    %seguir
    i = i+1;
end

%resultado del cronometro

clc;
fprintf('%g s de captura a %g cap/s \n',t,i/t);

%Limpiar el puerto

fclose(s);
delete(s);
clear s;
%end
```

Continuación apéndice 3.

```
function v_up_Callback(source,eventdata)
    yn=yn+1;
end

function v_down_Callback(source,eventdata)
    yn=yn-1;
end

function sube_Callback(source,eventdata)
    ys=ys+1;
end

function baja_Callback(source,eventdata)
    ys=ys-1;
end

function h_up_Callback(source,eventdata)
    xn=xn+1;
end

function h_down_Callback(source,eventdata)
    xn=xn-1;
end

function iz_Callback(source,eventdata)
    xs=xs+10;
end

function dr_Callback(source,eventdata)
    xs=xs-10;
end

end
```

Fuente elaboración propia, en lenguaje M de Matlab.

Apéndice 4. Presupuesto del osciloscopio digital

Material	Cantidad	Precio Unidad	Precio Total
Placa de fibra de Vidrio 10x12 cm	1	Q 60,00	Q 60,00
Cables de conexión de proyectos conectores hembra	9	Q 1,00	Q 9,00
Estaño por metro	1	Q 5,00	Q 10,00
Acido de oxido férrico	1	Q 15,00	Q 15,00
Resistencia 1 K Ω	1	Q 0,60	Q 0,60
Resistencia 5 K Ω	1	Q 0,60	Q 0,60
Resistencia 10 K Ω	5	Q 0,60	Q 3,00
Resistencia 11.5 K Ω	2	Q 0,60	Q 1,20
Resistencia 15 K Ω	1	Q 0,60	Q 0,60
Resistencia 120 K Ω	1	Q 0,60	Q 0,60
Resistencia 220 K Ω	1	Q 0,60	Q 0,60
Trimpot 1K Ω	1	Q 5,00	Q 5,00
Capacitor cerámico 100 nF	2	Q 1,00	Q 2,00
Capacitor cerámico 1 μ F	1	Q 3,50	Q 3,50
Capacitor cerámico 2 μ F	1	Q 3,50	Q 3,50
Capacitor Electrolítico 22 μ F	2	Q 1,00	Q 1,00
Capacitor Electrolítico 47 μ F	1	Q 2,00	Q 2,00
Amplificador operacional dual LM358	2	Q 10,00	Q 20,00
Convertidor de voltaje MAX660	1	Q 24,00	Q 24,00
Tira de pines de conexión 36 pines	1	Q 4,00	Q 4,00
Microcontrolador de placa integrada Texas Instrument Tiva C	1	Q 120,00	Q 120,00
Conector BNC	1	Q 10,00	Q 10,00
Punta de Osciloscopio	1	Q 150,00	Q 150,00
Caja de conexiones	1	Q 45,00	Q 45,00
Total, presupuesto			Q 491,2

Fuente: elaboración propia.

ANEXOS

Anexo 1. Hoja de datos integrados LM158

LMx58-N Low-Power, Dual-Operational Amplifiers

1 Features

- Available in 8-Bump DSBGA Chip-Sized Package, (See AN-1112, SNVA009)
- Internally Frequency Compensated for Unity Gain
- Large DC Voltage Gain: 100 dB
- Wide Bandwidth (Unity Gain): 1 MHz (Temperature Compensated)
- Wide Power Supply Range:
 - Single Supply: 3V to 32V
 - Or Dual Supplies: $\pm 1.5V$ to $\pm 16V$
- Very Low Supply Current Drain (500 μA)—Essentially Independent of Supply Voltage
- Low Input Offset Voltage: 2 mV
- Input Common-Mode Voltage Range Includes Ground
- Differential Input Voltage Range Equal to the Power Supply Voltage
- Large Output Voltage Swing
- Unique Characteristics:
 - In the Linear Mode the Input Common-Mode Voltage Range Includes Ground and the Output Voltage Can Also Swing to Ground, even though Operated from Only a Single Power Supply Voltage.
 - The Unity Gain Cross Frequency is Temperature Compensated.
 - The Input Bias Current is also Temperature Compensated.
- Advantages:
 - Two Internally Compensated Op Amps
 - Eliminates Need for Dual Supplies
 - Allows Direct Sensing Near GND and V_{OUT} Also Goes to GND
 - Compatible with All Forms of Logic
 - Power Drain Suitable for Battery Operation

2 Applications

- Active Filters
- General Signal Conditioning and Amplification
- 4- to 20-mA Current Loop Transmitters

3 Description

The LM158 series consists of two independent, high gain, internally frequency compensated operational amplifiers which were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage.

Application areas include transducer amplifiers, dc gain blocks and all the conventional op-amp circuits which now can be more easily implemented in single power supply systems. For example, the LM158 series can be directly operated off of the standard 3.3-V power supply voltage which is used in digital systems and will easily provide the required interface electronics without requiring the additional $\pm 15V$ power supplies.

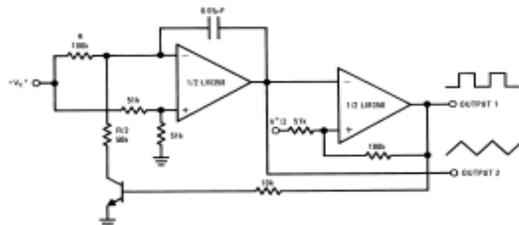
The LM358 and LM2904 are available in a chip sized package (8-Bump DSBGA) using TI's DSBGA package technology.

Device Information⁽¹⁾

PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM158-N	TO-CAN (8)	9.08 mm x 9.09 mm
	CDIP (8)	10.16 mm x 6.502 mm
LM258-N	TO-CAN (8)	9.08 mm x 9.09 mm
LM2904-N	DSBGA (8)	1.31 mm x 1.31 mm
	SOIC (8)	4.90 mm x 3.91 mm
	PDIP (8)	9.81 mm x 6.35 mm
LM358-N	TO-CAN (8)	9.08 mm x 9.09 mm
	DSBGA (8)	1.31 mm x 1.31 mm
	SOIC (8)	4.90 mm x 3.91 mm
	PDIP (8)	9.81 mm x 6.35 mm

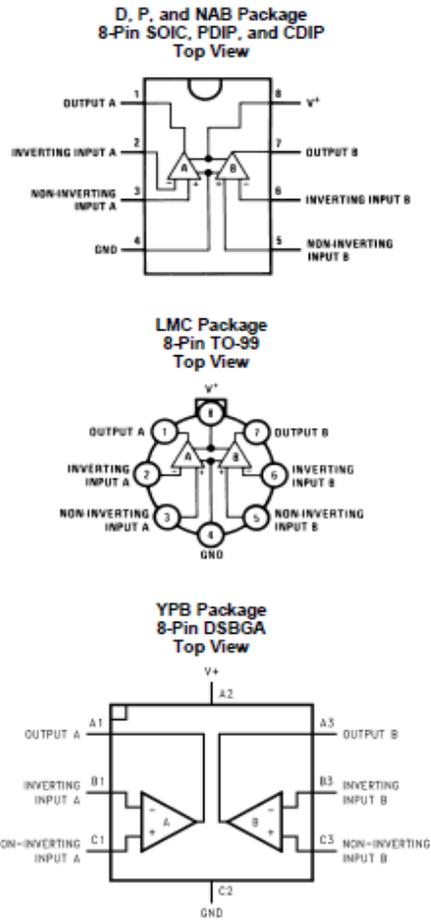
(1) For all available packages, see the orderable addendum at the end of the datasheet.

Voltage Controlled Oscillator (VCO)



Continuación anexo 1.

5 Pin Configuration and Functions



Pin Functions

DiP/LMC NO.	PIN		TYPE	DESCRIPTION
	DSBGA NO.	NAME		
1	A1	OUTA	O	Output, Channel A
2	B1	-INA	I	Inverting Input, Channel A
3	C1	+INA	I	Non-Inverting Input, Channel A
4	C2	GND / V-	P	Ground for Single supply configurations. negative supply for dual supply configurations
5	C3	+INB	I	Output, Channel B
6	B3	-INB	I	Inverting Input, Channel B
7	A3	OUTB	O	Non-Inverting Input, Channel B
8	A2	V+	P	Positive Supply

Fuente: LM158. <http://www.ti.com/lit/ds/symlink/lm158-n.pdf>. Consulta: 25 de abril de 2019.

Anexo 2. Hoja de datos integrado Max660

MAX660 Switched Capacitor Voltage Converter

1 Features

- Inverts or Doubles Input Supply Voltage
- Narrow SO-8 Package
- 6.5-Ω Typical Output Resistance
- 88% Typical Conversion Efficiency at 100 mA
- Selectable Oscillator Frequency: 10 kHz/80 kHz

2 Applications

- Laptop Computers
- Cellular Phones
- Medical Instruments
- Operational Amplifier Power Supplies
- Interface Power Supplies
- Handheld Instruments

3 Description

The MAX660 CMOS charge-pump voltage converter is a versatile unregulated switched-capacitor inverter or doubler. Operating from a wide 1.5-V to 5.5-V supply voltage, the MAX660 uses two low-cost capacitors to provide 100 mA of output current without the cost, size and EMI related to inductor-based converters. With an operating current of only 120 μ A and operating efficiency greater than 90% at most loads, the MAX660 provides ideal performance for battery-powered systems. MAX660 devices can be operated directly in parallel to lower output impedance, thus providing more current at a given voltage.

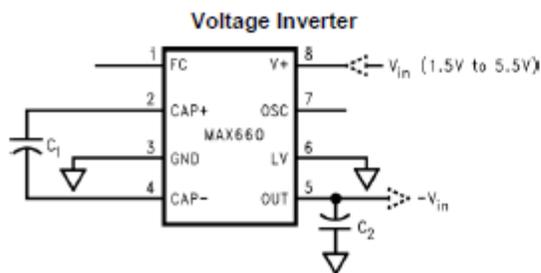
The FC (frequency control) pin selects between a nominal 10-kHz or 80-kHz oscillator frequency. The oscillator frequency can be lowered by adding an external capacitor to the OSC pin. Also, the OSC pin may be used to drive the MAX660 with an external clock up to 150 kHz. Through these methods, output ripple frequency and harmonics may be controlled.

Additionally, the MAX660 may be configured to divide a positive input voltage precisely in half. In this mode, input voltages as high as 11 V may be used.

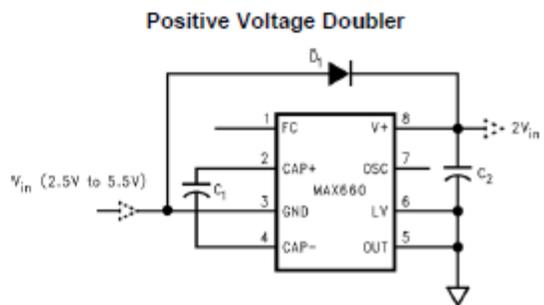
Device Information⁽¹⁾

PART NUMBER	PACKAGE	BODY SIZE (NOM)
MAX660	SOIC (8)	4.90 mm × 3.91 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.



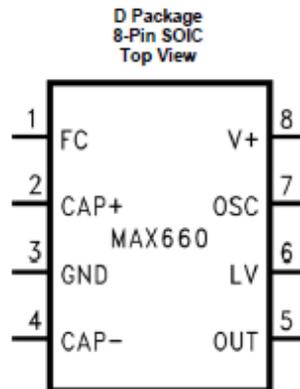
Copyright © 2016, Texas Instruments Incorporated



Copyright © 2016, Texas Instruments Incorporated

Continuación anexo 2.

6 Pin Configuration and Functions



Pin Functions

PIN		I/O	DESCRIPTION	
NAME	NO.		VOLTAGE INVERTER	VOLTAGE DOUBLER
CAP+	2	Power	Connect this pin to the positive terminal of charge-pump capacitor.	Same as inverter
CAP-	4	Power	Connect this pin to the negative terminal of charge-pump capacitor.	Same as inverter
FC	1	Input	Frequency control for internal oscillator: FC = open, $f_{osc} = 10$ kHz (typical); FC = V+, $f_{osc} = 80$ kHz (typical); FC has no effect when OSC pin is driven externally	Same as inverter
GND	3	Ground	Power supply ground input.	Power supply positive voltage input
LV	6	Input	Low-voltage operation input. Tie LV to GND when input voltage is less than 3.5 V. Above 3.5 V, LV can be connected to GND or left open. When driving OSC with an external clock, LV must be connected to GND.	LV must be tied to OUT.
OSC	7	Input	Oscillator control input. OSC is connected to an internal 15-pF capacitor. An external capacitor can be connected to slow the oscillator. Also, an external clock can be used to drive OSC.	Same as inverter except that OSC cannot be driven by an external clock
OUT	5	Power	Negative voltage output	Positive supply ground input
V+	8	Power	Power supply positive voltage input	Positive voltage output

Fuente: MAX660. <http://www.ti.com/lit/ds/symlink/max660.pdf>. Consulta: 24 de abril de 2019.

Anexo 3. Mapa de pines Tiva C

LaunchPads LM4F TM4C - Pins Maps 11-32



Flash	256	KB
SRAM	32	KB
Serial hardware		
ADC	12	bits
Use pins numbers only		
Quadrature Encoder - TTL level only on TM4C123GH6PM		



Rel Wins, 2012-2017
embedded.com/embedded.com/ek-tm4c123gx/
 version 2.1 2018-10-20

LaunchPad with LM4F120H5QR LaunchPad with TM4C123GH6PM



Hardware	Pin number	Other pin number
IC	Serial UART	SPI
analogRead()	digitalRead() and digitalWrite()	digitalWrite(), digitalWrite() and analogWrite()



VBUS_detection PD 7



Flash	256	KB
BRAM	32	KB
Serial hardware		
ADC	12	bits
Use pins numbers only		



Rel Wins, 2012-2017
embedded.com/embedded.com/ek-tm4c123gx/
 version 2.1 2018-10-20

StellarPad with LM4F120H5QR LaunchPad with TM4C123GH6PM



Hardware	Pin number	Other pin number
IC	Serial UART	SPI
analogRead()	digitalRead() and digitalWrite()	digitalWrite(), digitalWrite() and analogWrite()



VBUS_detection PD 7

LaunchPads LM4F TM4C - Pins Maps 12-33

Fuente: *Distribución de pines Tiva C.* <http://energia.nu/pinmaps/ek-tm4c123gx/>. Consulta: 24 de abril de 2019.

