

---

# Prototipado de un dispositivo portátil para la medición de actividad muscular

---

Carlos Mauricio Molina López





UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



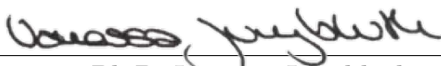
**Prototipado de un dispositivo portátil para la medición de  
actividad muscular**


Trabajo de graduación presentado por Carlos Mauricio Molina López  
para optar al grado académico de Licenciado en Ingeniería Electrónica

Guatemala,

2025

Vo.Bo.:

(f)   
Ph.D. Vanessa Jungbluth

(f)   
M.Sc. Carlos Esquit Hernández

Fecha de aprobación: Guatemala, 20 de noviembre de 2025.

Me es grato presentar el siguiente trabajo de graduación, el cual se ha realizado con gran entusiasmo en búsqueda de innovar utilizando la galga extensiométrica. El desarrollo de este trabajo de graduación busca establecer las bases para la utilización del sensor en un contexto biomédico. De esta manera, esto representa una base sólida para futuros proyectos de investigación en el ámbito de la instrumentación electrónica y biomédica de la Universidad del Valle de Guatemala.

Quiero tomar la oportunidad para agradecer a Dios y a María. Sin su guía y apoyo no podría haber realizado el siguiente trabajo. Su amor constante ha sido mi motivación para poder terminar mis estudios universitarios.

Quiero agradecer a mis padres, Carolina López y Lester Molina, quienes me han apoyado e incentivado incondicionalmente a lo largo de mi desarrollo académico y profesional. A mis hermanas, Paulina y María Fernanda Molina, por su apoyo constante. A mis abuelos, Carlos López y Soledad Morales, por su esfuerzo constante para asegurar la culminación de mis estudios universitarios.

Quiero agradecer a mi asesor de trabajo, Ph.D. Gabriel Barrientos, quien ha dedicado su tiempo y esfuerzo para el desarrollo correcto de este trabajo. A mi catedrático Pablo Mazariegos, por su valiosa orientación y a la Ph.D. Vanessa Jungbluth por su invaluable y constante guía durante este último año.

Finalmente, agradezco a mis amigos de la universidad Oscar Donis y Luis Pablo Carranza, por su constante aliento y reafirmación durante esta época universitaria.

<b>Prefacio</b>	<b>I</b>
<b>Índice de figuras</b>	<b>V</b>
<b>Índice de cuadros</b>	<b>VI</b>
<b>Resumen</b>	<b>VII</b>
<b>Abstract</b>	<b>VIII</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Antecedentes</b>	<b>3</b>
<b>3. Justificación</b>	<b>5</b>
<b>4. Objetivos</b>	<b>7</b>
4.1. Objetivo general . . . . .	7
4.2. Objetivos específicos . . . . .	7
<b>5. Alcance</b>	<b>8</b>
<b>6. Marco teórico</b>	<b>9</b>
6.1. Contracción muscular . . . . .	9
6.2. Biomecánica del músculo . . . . .	10
6.3. Galga extensiométrica . . . . .	12
6.4. Puente de Wheatstone . . . . .	16
6.5. Voltaje de excitación . . . . .	19
6.6. ADC . . . . .	20
6.7. Filtros digitales . . . . .	22
6.8. <i>Wifi</i> . . . . .	23

<b>7. Metodología</b>	<b>24</b>
7.1. Validación de circuito puente de Wheatstone . . . . .	24
7.2. <i>Post-processing</i> . . . . .	27
7.3. Caracterización . . . . .	30
7.4. Diseño de PCB . . . . .	31
7.5. Envío de data hacia computadora . . . . .	34
<b>8. Resultados</b>	<b>36</b>
<b>9. Discusión</b>	<b>45</b>
<b>10. Conclusiones</b>	<b>47</b>
<b>11. Recomendaciones</b>	<b>49</b>
<b>12. Referencias</b>	<b>51</b>
<b>13. Anexos</b>	<b>55</b>
13.1. Código en Matlab para el análisis de señal en archivo .csv . . . . .	55
13.2. Código para envío de lectura ADC utilizando un sensor mediante comunicación serial . . . . .	56
13.3. Código para obtención de lectura ADC mediante comunicación serial en Matlab	57
13.4. Código para envío de lectura ADC utilizando dos sensores mediante comunicación serial . . . . .	58
13.5. Código para caracterizar la señal como una gráfica de ADC vs Tiempo . . . .	60
13.6. Código para enviar la señal como data a través de <i>wifi</i> . . . . .	63
13.7. Código para la extracción de data de paquetes entrantes en la interfaz <i>wifi</i> . .	65

1.	Tipos de contracción muscular . . . . .	10
2.	Elementos principales de la estructura de un músculo . . . . .	11
3.	Estructura de una fibra muscular . . . . .	11
4.	Estructura de la miofibrilla . . . . .	12
5.	Deformación en una galga . . . . .	13
6.	Estructura de una galga . . . . .	13
7.	Mecanismo de una galga . . . . .	14
8.	Variables para deformación de una galga . . . . .	15
9.	Diferentes geometrías para las rejillas en una galga . . . . .	16
10.	Circuito puente de Wheatstone . . . . .	16
11.	Cuarto de puente de Wheatstone . . . . .	17
12.	Medio puente de Wheatstone . . . . .	18
13.	Frecuencia de muestreo y señal procesada . . . . .	20
14.	Resolución de 2 bits . . . . .	21
15.	Resolución de 4 bits . . . . .	21
16.	Diagrama de bloques de microcontrolador ESP32 . . . . .	23
17.	Galga adherida con agente adhesivo a viga de plástico . . . . .	25
18.	Galga adherida con cinta a viga de plástico . . . . .	25
19.	Circuito de cuarto de puente de Wheatstone en LTspice . . . . .	26
20.	Circuito de cuarto de puente de Wheatstone en <i>proto</i> board . . . . .	26
21.	Visualización de señal en osciloscopio con ajuste de escala . . . . .	27
22.	Esquemático de circuito para procesamiento digital de voltaje en puente Wheatstone . . . . .	28
23.	Dos galgas adheridas con cinta a viga de plástico . . . . .	29
24.	Circuito de medio puente de Wheatstone . . . . .	30
25.	Esquemático de circuito para procesamiento digital de voltaje con dos galgas	30
26.	Esquemático de circuito completo para lectura y caracterización de cambios en puente Wheatstone . . . . .	31
27.	Creación de esquemático y cuerpo 3D . . . . .	32
28.	PCB con componentes . . . . .	33
29.	<i>Routing</i> para los componentes en la PCB diseñada . . . . .	33
30.	Aplicación de filtro en Wireshark . . . . .	34

31.	Esquemático de circuito en PCB modificado para envío de señal por <i>wifi</i> . . .	35
32.	Lectura de voltaje en nodo $V_C$ para sensor adherido con cinta en estiramiento	37
33.	Lectura de voltaje en nodo $V_C$ para sensor adherido con cinta en compresión .	37
34.	Lectura de diferencia de voltaje entre nodos sin manipulación de galga extensiométrica en configuración de cuarto de puente de Wheatstone . . . . .	38
35.	Lectura del voltaje posprocesamiento digital sin manipulación de la galga extensiométrica . . . . .	38
36.	Lectura de voltaje procesada por microcontrolador con y sin manipulación de la galga extensiométrica sujeta con cinta a la viga . . . . .	39
37.	Lectura de voltaje procesada por microcontrolador con y sin manipulación de la galga extensiométrica utilizando dos galgas en paralelo adheridas a la viga con cinta . . . . .	40
38.	PCB con componentes en software Altium . . . . .	41
39.	PCB real con componentes soldados . . . . .	41
40.	PCB real con <i>display</i> integrado . . . . .	42
41.	Caracterización del sensor en deformación hacia abajo . . . . .	42
42.	Caracterización del sensor en deformación hacia arriba . . . . .	43
43.	Gráfica de tráfico entrante por interfaz <i>wifi</i> de computadora filtrado para paquetes provenientes de microcontrolador . . . . .	43
44.	Análisis de paquete entrante . . . . .	44
45.	Gráfica de data obtenida de paquetes en computadora en tiempo real . . . . .	44

---

## Índice de cuadros

---

1.	Declaración de valor de $\lambda$ en el código . . . . .	28
2.	Implementación de filtro digital como ecuación de diferencias . . . . .	29
3.	Declaración de límites para caracterización . . . . .	31
4.	Valor de resistencia obtenido en la galga bajo deformación para difentes métodos de fijación . . . . .	36
5.	Variabilidad de lecturas en unidades de voltaje para diferentes implementaciones del filtro digital . . . . .	39
6.	Medición de potencia . . . . .	40

La actividad muscular es una de las áreas de mayor campo de estudio debido a los diferentes acercamientos hacia su medición. Muchos sensores de actividad musculares se basan en biomarcadores, mientras otros centran su enfoque hacia los campos eléctricos. Técnicas modernas recaen en métodos invasivos o poco fiables para la recolección de datos biométricos. La gran mayoría de estos fallan en presentar la información adquirida. Esto representa una carencia para aquellas personas que busquen aumentar masa muscular y que busquen un dispositivo que facilite este proceso. Una de las perspectivas más prometedoras que surgen es la de un acercamiento mecánico, para el cual únicamente se recolecta data en base a cambios por la deformación del sensor debido a un estímulo externo.

A lo largo de este trabajo se busca explorar la aplicación de la galga extensiométrica simulando la superficie muscular utilizando una viga en voladizo de plástico. Para explorar la aplicabilidad del sensor, se estableció un sistema de medición, compuesto de un circuito puente de Wheatstone en conjunto con un microcontrolador con un convertidor de analógico a digital (ADC, por sus siglas en inglés) de 12 bits y la implementación del filtro digital *leaky integrator*. El resultado fue la transducción de una variabilidad en resistencia de  $1.1 \Omega$  hacia un cambio de voltaje de 2.9 mV. Implementado, además, una caracterización con un *display*, la deformación del sensor se manifestó como variaciones notorias en escala de mV. Finalmente, el sistema se presentaría en una placa de circuito impreso (PCB, por sus siglas en inglés) de  $5 \times 5$  cm. Además, se destacó la capacidad del microcontrolador de enviar la data de manera inalámbrica a 27.46 paquetes/segundo.

**Palabras clave:** Actividad muscular, galga extensiométrica, ingeniería, sensor, filtro digital, viga en voladizo, no invasivo.

The field of muscle activity is a major area of study due to different approaches toward its measurement. Many muscle activity sensors are based on biomarkers, while others shift their focus toward electrical fields. Modern techniques rely on invasive or unreliable methods for the collection of biometric data. The vast majority of these methods fail to present the acquired information effectively. This represents a deficiency for individuals seeking to increase muscle mass who require a device to facilitate this process. One of the most promising emerging perspectives is the mechanical approach. This method relies on collecting data based on changes due to the sensor's deformation from an external stimulus.

Throughout this work, the application of the strain gauge is explored by simulating the muscle surface using a plastic cantilever beam. To explore the applicability of the sensor, a system was established, consisting of a Wheatstone bridge circuit combined with a 12-bit analog-to-digital converter (ADC) microcontroller and the implementation of the digital Leaky Integrator filter. The result was the transduction of a resistance variability of  $1.1 \Omega$  into a voltage change of  $2.9 mV$ . Furthermore, implementing a characterization system with a display, the sensor's deformation was made perceptible as changes on the  $mV$  scale. Finally, the system was implemented on a  $5 \times 5 cm$  printed circuit board (PCB) to ensure its portability. Additionally, the microcontroller's capacity to send data wirelessly at 27.46 packets/second was demonstrated.

**Keywords:** Muscle activity, strain gauge, engineering, sensor, digital filter, cantilever beam, non invasive.

Existen diversos métodos propuestos para la transducción de la actividad muscular hacia una lectura de utilidad. En el capítulo 2 se indaga sobre los diferentes acercamientos hacia la medición de actividad muscular y sus limitaciones [1], tanto acercamientos modernos, como aquellos con mayor presencia comercial [2], [3]. En el capítulo 3 se detalla sobre la necesidad de realización de un dispositivo autónomo y portátil, con una caracterización útil para el usuario. Esto con el propósito de desarrollar un recurso atractivo e innovador para la facilidad de ganancia de masa muscular para aquellos que lo necesiten [4].

Para el capítulo 4 y 5, se desarrolla sobre el objetivo y alcance del trabajo, el cual es fundamentar el uso del sensor conocido como la galga extensiométrica, que se utiliza para medir la deformación ( $\epsilon$ ) de un material. Esto se realiza mediante la representación de la fibra muscular como una viga en voladizo [5], [6]. Además, se detallan las metas a cumplir para demostrar la relevancia del sensor en un prototipo para medición de actividad muscular.

Típicamente, el sensor se utiliza en estructuras como vigas, siendo adherido con un agente adhesivo para medir la deformación del material [7]. Es así que en el capítulo 6 se justifica la teoría detrás del por qué el sensor puede ser utilizado en un contexto biomecánico; la relación entre la estructura del músculo y la deformación que resulta en un alargamiento y compresión de un área muscular [8]. Además, se detalla cómo la topología del circuito de puente de Wheatstone es importante en acompañamiento del sensor para traducir la deformación mecánica a una señal de voltaje.

En adelante, el capítulo 7 relata la metodología que se llevó a cabo para medir la deformación mecánica del sensor. Para esto, se adhirió el sensor a una simulación de viga de plástico flexible. Los cambios en resistencia del sensor fueron traducidos mediante un cuarto de puente de Wheatstone y se procesaron con el microcontrolador ESP32-C3 Super Mini como una señal de voltaje. Esta señal fue posprocesada así mediante un filtro digital IIR conocido como *leaky integrator* para caracterizar los cambios del sensor en una *display* OLED. Con esto, se desarrolló una placa PCB para hacer del prototipo portátil y autónomo. Más adelante, también se exploró la capacidad del microcontrolador para el envío de datos de manera inalámbrica mediante *wifi*.

En el capítulo 8 se demuestra la respuesta del sensor comercial al ser deformado y ser adherido únicamente con cinta adhesiva. Además, se detalla el efecto del filtrado digital al tener una señal con cambios reducidos de voltaje. En el capítulo 9 se brinda una discusión sobre lo que representan estos resultados con miras a la implementación del sensor en una superficie muscular.

El capítulo 10 ofrece conclusiones sobre los descubrimientos del trabajo y el capítulo 11 recomendaciones sobre lo que se debe desarrollar a futuro para mejorar el prototipo.

La tecnología utilizada para el control de actividad muscular se basa principalmente en el uso de sensores para la obtención de datos [9]. Estos datos se producen como un efecto del movimiento del músculo, ya sean estos cambios internos en el cuerpo o externos [10]. Estas diferentes tecnologías sensoriales pueden variar dependiendo de varios factores, por ejemplo si se utilizan biomarcadores, cuál es la naturaleza de la señal, la información obtenida y cómo se recoge, cómo se comunica la información y qué señal [9].

Tecnologías particulares como la de los biosensores se basan en obtener el resultado entre la interacción de un analito —un marcador biológico que ocurre naturalmente dentro del cuerpo (y se transporta en la sangre, sudor u orina)— y una superficie tratada [11]. Para la actividad física, un analito tal como lactato se emplea como biomarcador confiable para cuantificar el esfuerzo hecho por un músculo [12]. Gran parte de estos métodos exitosos de detección de actividad muscular mediante biosensores son de naturaleza invasiva o mínimamente invasiva [13].

Así entonces, algunos biosensores actuales evitan el uso de análisis basados en la sangre para obtener una lectura de lactato. Por ejemplo, aquellos dispositivos de biodetección que pegan un parche de micro agujas al área muscular de interés [2]. Esto logra que el dispositivo acceda a una muestra de significancia con la menor profundidad posible, siendo así mínimamente invasivo. Así una vez las agujas penetren en la sección transdérmica, se utiliza el líquido intersticial para acceder al lactato. Este analito y la reacción proceden a ser transducida mediante un biosensor amperométrico a una corriente legible. Esta señal se procesa y se envía a una aplicación móvil donde un usuario puede ver toda esta transducción caracterizada como un gráfico de corriente con respecto al tiempo.

Aunque los dispositivos mínimamente invasivos se presentan como una opción prometedora, un enfoque no invasivo suele ser una preferencia entre los sujetos al portar un sensor portátil [14]. La mayoría de estos dispositivos no invasivos se basan en detección de electromiografía (EMG). La electromiografía funciona mediante la grabación de una serie de señales eléctricas que derivan del músculo, específicamente como respuesta muscular a la estimulación nerviosa [15]. Esta señal es recolectada por electrodos. El módulo BITalino es

una opción actual como sensor de electromiografía. Este tipo de sensor utiliza 3 electrodos: uno como electrodo de referencia y los últimos como electrodo positivo y negativo. Estos se colocan en la piel y luego se conectan con un cable micro-USB directamente a la placa. Luego, en base a la activación muscular, un cambio de potencial eléctrico entre los electrodos es procesado por el módulo y la información es enviada a una computadora. Además, un dispositivo reciente como el sensor Callibri es un ejemplo de una solución en curso que abarca tanto el concepto de ser inalámbrico como de ser no invasivo [16]. Este dispositivo de tamaño mediano está conectado al área muscular de interés y envía información del EMG entre otros datos cuantificables como velocidad angular y aceleración a través de Bluetooth hacia una computadora [3].

La tecnología actual para medición de actividad muscular se clasifica como de corto o alcance restringido, proporcionando al usuario final de información y datos sin validez técnica. La mayoría de los dispositivos disponibles se comprometen únicamente a mostrar una lectura precisa, tal como el de una gráfica de voltaje/corriente con respecto al tiempo, o bien, algún tipo de lectura más específica, como el cambio en concentración de lactosa con el tiempo [12]. Estas gráficas caracterizan la actividad muscular de manera eficiente, pero esta información numérica aún deja gran espacio a su interpretación. Agregando a esto, algunos de los métodos convencionales empleados para la recolección de datos biométricos para cuantificar la actividad muscular pueden ser afectados por procesos biológicos internos, presentando así inconsistencias o ruido en las lecturas [1]. Incluso más a detalle, algunos de estos dispositivos dependen de algún tipo de ecosistema tecnológico para funcionar, ya sea un *smartwatch*, *smartphone*; algún computador o dispositivo maestro el cual debe estar en la cercanía del individuo para analizar y procesar los datos recolectados por el sensor [17]. Todo esto evidencia la necesidad de desarrollar una herramienta autónoma, basada en un método de adquisición de datos más robusto y fiable, que optimice la caracterización del monitoreo de la actividad muscular. Un sistema con estas capacidades mejoraría la experiencia del usuario, facilitando el proceso de ganar músculo y haciendo así más eficiente el desarrollo de este proceso.

Esta necesidad se ve respaldada por el interés creciente en aumentar la masa muscular en distintos grupos objetivo [4]. Mientras que algunas personas buscan ganar masa muscular por motivos estéticos, otras lo hacen para compensar la pérdida natural de masa muscular que ocurre con la edad, o como consecuencia de una lesión o enfermedad [4], [18].

De acuerdo con estudios fisiológicos realizados en 2018, la masa muscular puede tardar hasta 20 semanas junto a la realización de entrenamiento intenso para poder incrementar hasta en un 20% [19]. Por lo que proporcionar una solución que de información de utilidad en vivo, sobre la reacción de un músculo (o grupo muscular), con respecto a un ejercicio de resistencia específico, es vital para optimizar la ejecución de un ejercicio. Además, una mayor eficiencia en la ejecución del ejercicio se traduce a un uso óptimo del tiempo, acelerando así

el proceso de ganancia de actividad muscular.

Tecnología actual, como aquella en base a la electromiografía (EMG) pareciera ser la idea que engloba muchos aspectos importantes, como la portabilidad del dispositivo, proveer una lectura como la actividad eléctrica, y así ver los estímulos que se le da a un área muscular, entre otros. Su defecto recae en la confiabilidad de estos datos y su caracterización [1]. Las señales EMG resultantes presentan un problema. En su estado normal, sin implicar un procesamiento de señales, son variables debido a su naturaleza, existiendo así varios métodos de normalización [20]. Estas señales pueden depender en gran medida del individuo e incluso presentar cambios para el mismo individuo durante períodos, debido al metabolismo [21]. Esto plantea la última característica justificada del sensor a desarrollarse: Debe esforzarse por ser personalizado y global al mismo tiempo. Significa que, aunque considera diferentes respuestas por parte de los diferentes individuos, será capaz de dar una métrica significativa para cada uno de ellos.

El dispositivo final no solamente debe proveer información, sino que debe procesar las señales y datos involucrados, para así, mediante un posprocesamiento, mostrar un resultado enriquecedor para el usuario, más allá de una gráfica de voltaje vs tiempo, característica de tecnologías actuales [22].

El sensor de deformación, o galga extensiométrica, es capaz de transducir la deformación mecánica a un cambio en resistencia eléctrica. Debido a la naturaleza de la contracción muscular que sucede durante un ejercicio de resistencia, se puede modelar desde una perspectiva mecánica [23]. Esto permitiría utilizar el sensor para medir esta deformación mecánica como producto de la actividad muscular y generar una solución aplicable para cada individuo. Pues la deformación mecánica es una métrica universal, a diferencia de los biomarcadores que son específicos para cada usuario.

### 4.1. Objetivo general

Desarrollar un prototipo de dispositivo portátil para medir actividad muscular, utilizando una galga extensiométrica.

### 4.2. Objetivos específicos

- Diseñar un sistema de medición de voltaje para la galga extensiométrica, utilizando un circuito de puente de Wheatstone.
- Desarrollar un circuito impreso para integrar la solución, incluyendo un algoritmo embebido para caracterizar la actividad muscular.
- Transmitir las mediciones de voltaje a una computadora, por medio de comunicación inalámbrica.

El alcance de este trabajo está delimitado por el objetivo general de desarrollar un prototipo funcional para la medición de la actividad muscular mediante una galga extensiométrica. Para esto, se definieron ciertos límites.

Uno de ellos se centra en la aplicación biológica del prototipo. Este trabajo no aborda pruebas de campo sobre músculo esquelético real. El trabajo se limita a la medición de la deformación de una viga en voladizo de plástico flexible, sirviendo como un análogo del tejido muscular. Más adelante, el sensor utilizado para medir esta deformación fue la galga extensiométrica. Se utilizaron galgas lineales adquiridas comercialmente. Con lo cual se limita a la evaluación del recurso y no a su desarrollo. Además, en línea con el tipo de deformación limitada por la viga, no se evalúa la aplicabilidad de galgas con diferentes estructuras geométricas o patrones al *layout* lineal.

Ahora bien, en cuanto al diseño del sistema para la medición de la deformación del análogo al músculo, se limita este a la incorporación de un circuito con configuración de medio o cuarto puente de Wheatstone, un microcontrolador con un ADC de 12 bits y un filtro digital. El trabajo se enfoca en fundamentar los elementos mínimos y óptimos requeridos para la medición de deformación del sensor. *Hardware* adicional o diferente a este no es justificado por los hallazgos del trabajo.

Una vez se desarrolló el sistema, la caracterización de la deformación en función de cotas fue implementada. La caracterización desarrollada no refleja una correlación entre una lectura digital y una fuerza real ejercida por el músculo. La caracterización se basa en cotas definidas por un análisis de la variabilidad en voltaje presentado en experimentación.

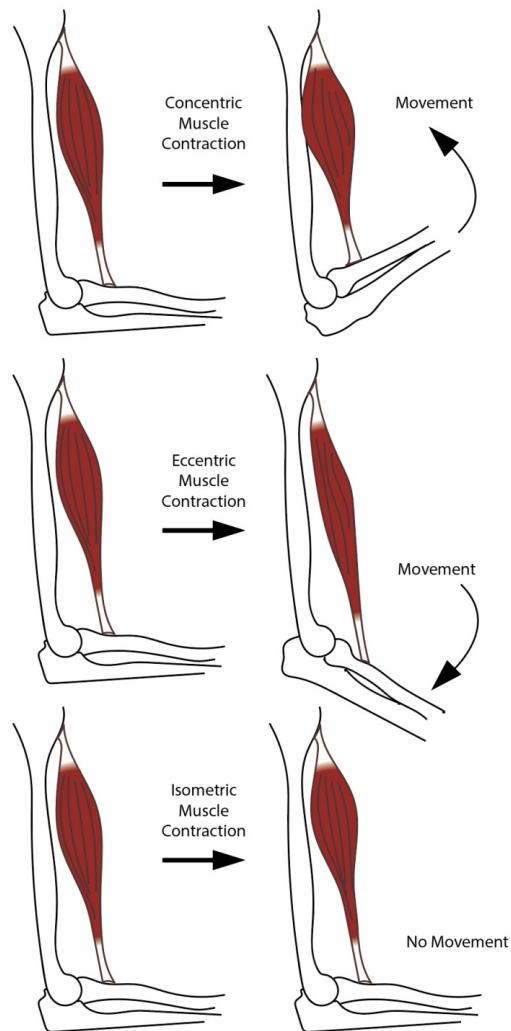
También, respecto al diseño de *hardware*, se debe remarcar que la PCB que abarca el sistema para medir la deformación del sensor se centra en la funcionalidad y el diseño compacto. No se consideran requerimientos de certificación o seguridad.

Finalmente, es importante presentar que la comunicación inalámbrica del microcontrolador se limita a una evaluación mediante protocolo UDP en una red local aislada.

## 6.1. Contracción muscular

La contracción muscular se define como el alargamiento o acortamiento de una fibra muscular, produciendo tensión dentro del músculo, para tirar o empujar una resistencia. Se puede dividir en 4 tipos: Isométrica, isotónica, concéntrica y excéntrica. La contracción concéntrica y excéntrica comprenden un tipo de contracción muscular en la que el músculo genera una fuerza modificando su longitud (Figura 1). De un lado la contracción muscular concéntrica ocurre cuando el músculo se acorta, generando una fuerza capaz de superar una carga determinada. Esta carga puede ser, por ejemplo, una mancuerna. Mientras que, del otro lado, la contracción muscular excéntrica, se genera una fuerza contra una carga mediante el alargamiento del músculo; sin embargo, en este caso la carga es mayor que la fuerza ejercida por el músculo [15].

**Figura 1.** Tipos de contracción muscular



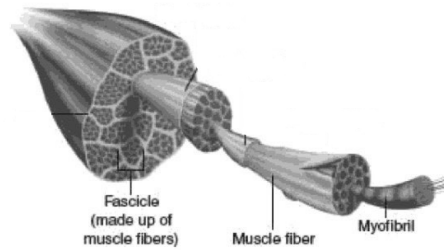
Nota. Esta imagen fue obtenida de [15].

Estos dos tipos de contracción muscular describen las contracciones habituales realizadas durante ejercicios de gimnasio, como el *curl* de bíceps o sentadillas con barra.

## 6.2. Biomecánica del músculo

El músculo esquelético se compone de conjuntos de fibras musculares (posicionadas en paralelo) agrupadas en los denominados fascículos [8]. Estos fascículos, específicamente aquellos que se ubican hacia la parte externa del músculo, cubren cada uno cierto ancho, con lo que cada uno de estos cubra un área superficial (Figura 2).

**Figura 2.** Elementos principales de la estructura de un músculo

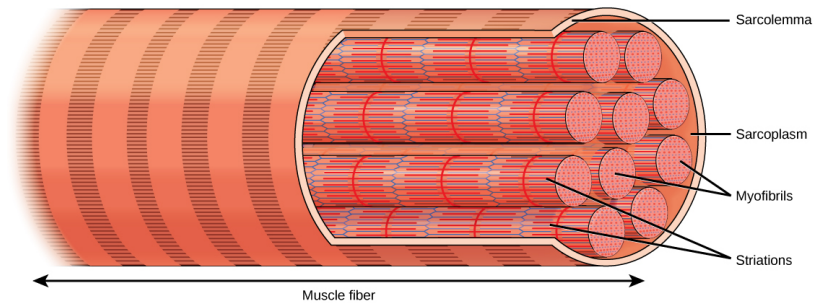


Nota. Esta imagen fue modificada de [8].

En muchas ocasiones, estos fascículos se encuentran agrupados de manera arbitraria, pero dependiendo del grupo y la ubicación, la fuerza del músculo puede ser en la misma dirección u opuesta entre grupos de fascículos [8].

Más a detalle, cada fibra muscular se compone de un conjunto de miofibrillas agrupadas en paralelo (Figura 3).

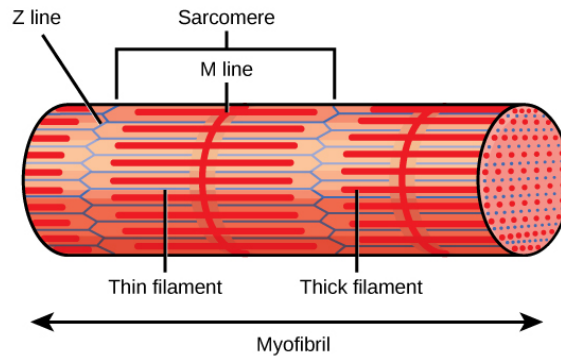
**Figura 3.** Estructura de una fibra muscular



Nota. Esta imagen fue obtenida de [24].

Y es adentro de cada una de estas miofibrillas que se encuentra el material contráctil. Este material se organiza en unidades denominadas sarcómeros (Figura 4).

**Figura 4.** Estructura de la miofibrilla



Nota. Esta imagen fue obtenida de [24].

Por lo que, cuando un músculo se acorta o alarga durante la contracción muscular, es la longitud del sarcómero la que se reduce en una cantidad proporcional. Dando a entender que esta es la unidad en la que se produce el cambio de longitud del músculo activamente [25]. Y estos, a pesar de tener un rango de operación, aún permiten que el músculo sea extremadamente plástico [26]. Sea entonces que se genere una tensión mediante una contracción muscular concéntrica o excéntrica, este un resultado directo del cambio en longitud en los sarcómeros.

Esto da lugar así a que las fibras musculares logren ejercer una deformación axial, siendo esta un estiramiento u acortamiento [8]. Este alargamiento como encogimiento de las fibras musculares es estudiado ampliamente por la biomecánica. En ocasiones, se modela el músculo esquelético como una viga [5]. Incluso se modela la mecánica de algunos músculos como simples objetos geométricos [8] dependiendo del músculo en cuestión.

En ocasiones también se utiliza la contracción muscular como una guía de diseño para las vigas en voladizo [6]. La deflexión de una viga puede ser un modelo simplificado de la contracción de las fibras musculares.

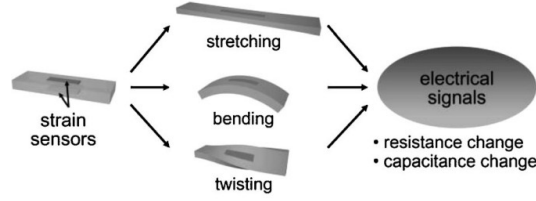
### 6.3. Galga extensiométrica

La fuerza de tracción generada por los músculos se transfiere a una deformación superficial medible.

El sensor de deformación conocido como la galga extensiométrica deriva su funcionamiento en torno a la deflexión de un material conductor. Este sensor tiene como propósito identificar diferentes tipos de deformación. La deformación ( $\epsilon$ ) se define como la relación entre la longitud de un material afectada por una fuerza aplicada con respecto a su longitud inicial [27]. La deformación es entonces el cambio que se da por alargamiento o contracción de un material, ya sea que el sensor sea estirado, torcido, o bien, en el caso de ser doblado (Figura 5). Por lo mismo, es necesario que el sensor esté adherido fijamente a una superficie, como lo es la de un área muscular, para poder proveer información relevante; puesto que la

lectura proviene de cambios al sensor directamente, y no por cambios externos a este.

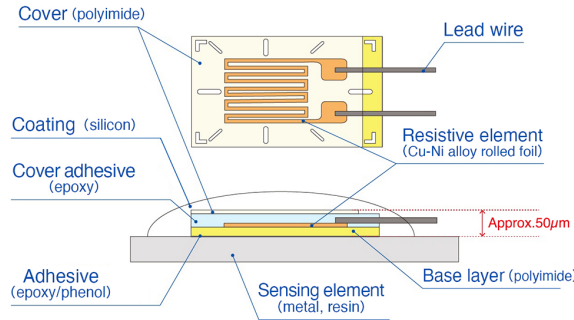
**Figura 5.** Deformación en una galga



Nota. Esta imagen fue obtenida de [28].

El sensor está compuesto por tres partes importantes: Un material base o sustrato, como la poliimida, una aleación de metal (cobre y níquel) que compone la cuadrícula del sensor (siendo este el material conductor) y dos cables que permiten conectar este sensor hacia un circuito [29] (Figura 6). Esta cuadrícula que se forma en el sensor es la organización de un conductor único en trazas en *zig-zag*, lo que posibilita maximizar su longitud efectiva.

**Figura 6.** Estructura de una galga



Nota. Esta imagen fue obtenida de [29].

Estos sensores de deformación son característicos por tener una resistencia eléctrica nominal, la cual varía a medida que se aplique una deformación al sensor. Los sensores adquiridos comercialmente tienden a tener una resistencia nominal de 120  $\Omega$ , 350  $\Omega$  y 1000  $\Omega$  [27].

Este valor de resistencia ( $R$ ) se rige por la siguiente ecuación:

$$R = \rho \frac{L}{A} \quad (1)$$

donde  $\rho$  es la resistividad del material del conductor (en  $\Omega \cdot m$ ),  $l$  es la longitud del conductor (en m) y  $A$  es la sección transversal del conductor (en  $m^2$ ).

Debido a que  $\rho$  es una constante propia del material, esto permite hacer la siguiente relación matemática, donde el valor de resistencia de la galga es directamente proporcional

a la longitud total de la cuadrícula e inversamente proporcional a la sección transversal del conductor:

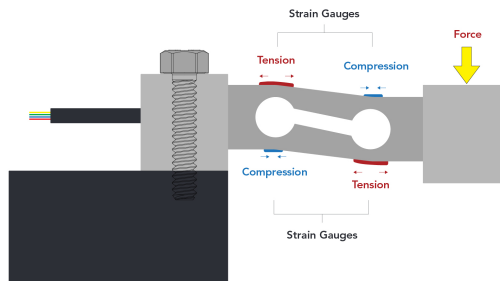
$$\Delta R \propto \Delta l \quad (2)$$

$$\Delta R \propto \frac{1}{\Delta A} \quad (3)$$

Además, hay que tomar en cuenta que el cambio en la longitud del conductor durante la deformación del sensor es considerablemente más alta que el cambio en la sección transversal del conductor. Esto se debe a la fabricación del sensor, donde se tiene como objetivo minimizar la sección transversal de la rejilla ( $A$ ) [30]. Por lo que, para mayor practicidad, se puede reducir la dependencia de la resistencia únicamente a la longitud del conductor ( $l$ ) como se presenta en (2).

La galga extensiométrica puede experimentar cambios de alargamiento como de contracción al estar adherida a una superficie que experimente una deformación. Esto se traduce al incremento y decremento en la resistencia del sensor continuamente durante periodos de tensión y compresión. Esto ya que al estar adherida la base del sensor a una superficie, este se doblaría en una dirección u otra (Figura 7). Esto se explora a detalle en estructuras rígidas, como vigas o estructuras metálicas, para medir deformaciones en estas.

**Figura 7.** Mecanismo de una galga



Nota. Esta imagen fue obtenida de [31].

Al estirar el sensor, se incrementa la longitud del conductor, y por ende, se tiene una mayor resistencia. Del otro lado, al comprimir el sensor, se estaría disminuyendo la longitud del conductor, resultando así en una menor resistencia.

Alargamiento de Galga :  $\uparrow R$

Contracción de Galga :  $\downarrow R$

Este cambio en resistencia del sensor puede ser de hasta un máximo de 2% en incremento y disminución con respecto a su valor nominal [32]. Así entonces, al utilizar una galga extensiométrica de p. ej. 350  $\Omega$ , esta podría variar al deformarse, entre aproximadamente 346.5  $\Omega$  y 353.5  $\Omega$ .

### 6.3.1. Factor de galga (GF)

El factor de galga ( $GF$ ) es un parámetro clave que se utiliza para reconocer la sensibilidad del sensor hacia su deformación [30]. A la vez esto se traduce en la cantidad de variabilidad de resistencia que presentará el sensor en función de la deformación que se le aplique.

Típicamente los sensores tienden a tener un  $GF$  entre 2 y 5, pero esto no es necesariamente la constante para todos los sensores [33].

El factor de galga se rige entonces por la siguiente ecuación:

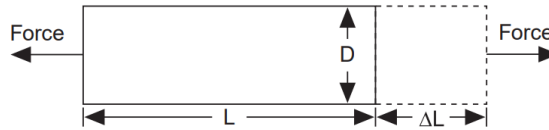
$$GF = \frac{\Delta R/R}{\Delta L/L} \quad (4)$$

para la cual los cambios en la longitud del sensor respecto a su longitud original se refieren a la deformación ( $\epsilon$ ), o bien:

$$\epsilon = \frac{\Delta L}{L} \quad (5)$$

Tomando en cuenta una galga lineal, este cambio en longitud se puede dar ya sea porque el sensor experimente dos tipos de deformación: de tensión, lo que representa un alargamiento del sensor, o de compresión, que representa un acortamiento (Figura 8).

**Figura 8.** Variables para deformación de una galga

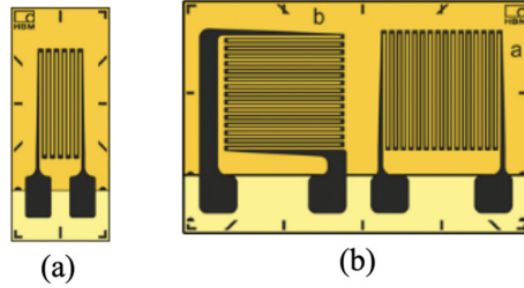


Nota. Esta imagen fue obtenida de [30].

### 6.3.2. Gauge pattern

La geometría detrás de la rejilla de una galga es importante, considerando el tipo de deformación esperado para la superficie y material en el que se emplea. Para esto, se fabrican galgas extensiométricas con distintos tipos de *layout* para la rejilla (Figura 9). Una rejilla lineal puede medir deformación en un eje, específicamente la que ocurre a lo largo del eje horizontal formado por los cables de la rejilla [34]. Por otro lado, una rejilla en forma de T es capaz de medir la deformación en dos ejes simultáneamente, para una deformación que se da tanto en el eje horizontal como en el eje vertical.

**Figura 9.** Diferentes geometrías para las rejillas en una galga



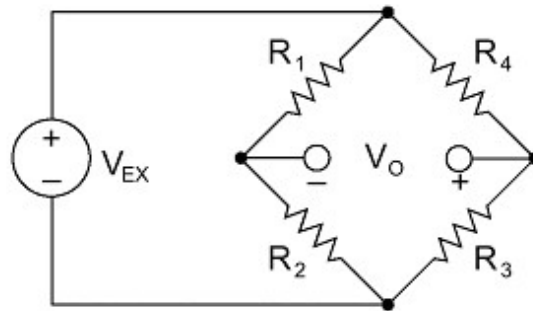
Nota. La imagen muestra las diferentes estructuras geométricas utilizadas para la rejilla de la galga en diferentes aplicaciones. (a) Rejilla lineal y (b) rejilla en forma de T. Esta imagen fue obtenida de [34].

## 6.4. Puente de Wheatstone

El puente de Wheatstone es una topología de circuito que permite convertir variaciones en resistencia eléctrica hacia cambios en voltaje [27]. Este circuito se usa ampliamente en conjunto con resistencias variables, como las galgas extensiométricas (sensores de deformación), para procesar los cambios en resistencia como señales eléctricas en un microcontrolador. A manera simple, el circuito se compone de dos divisores de voltaje en paralelo, en un arreglo de rombo (Figura 10). Los divisores se componen de las resistencias  $R_1$ ,  $R_2$ ,  $R_3$  y  $R_4$ . Estas pueden ser fijas y/o variables, dependiendo del número de galgas a utilizar.

Al tener dos divisores de voltaje, se cuenta con dos nodos principales, a través de los cuales se obtienen dos lecturas diferentes de potencial eléctrico. Los cambios en las lecturas de potencial se obtendrán en unidades de mV debido al cambio en unidades de  $\Omega$  por parte de la galga (sensor). Finalmente, se cuenta con un voltaje de entrada  $V_{EX}$  el cual tiene como propósito alimentar el circuito en cuestión, además de establecer la cota superior de voltaje que se estará dividiendo por los divisores.

**Figura 10.** Circuito puente de Wheatstone



Nota. Esta imagen fue obtenida de [35].

El voltaje de salida ( $V_o$ ) se puede expresar mediante la siguiente ecuación:

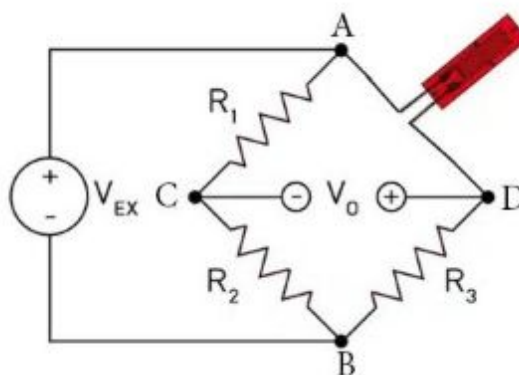
$$V_o = \left( \frac{R_3}{R_3 + R_4} - \frac{R_2}{R_1 + R_2} \right) \cdot V_{EX} \quad (6)$$

De la ecuación anterior es importante notar que ambos divisores de voltaje tienen un aporte para el voltaje de salida  $V_o$  resultante; siendo parte del factor que divide el voltaje de entrada  $V_{EX}$ . Es importante entonces diseñar en función de obtener un valor positivo para el voltaje de salida. Esto ya que este voltaje de interés es comúnmente amplificado en una segunda etapa empleando amplificadores operacionales (OPAMPS) [36]. Y considerando que los microcontroladores no soportan el procesamiento de voltajes negativos, es importante hacer un diseño de acuerdo a rangos operativos de un microcontrolador. Para cumplir con esto, la siguiente relación debe tomar efecto, donde:

$$\frac{R_3}{R_4} \geq \frac{R_2}{R_1}$$

Existen múltiples tipos de puentes Wheatstone, entre ellos la configuración de cuarto, medio y puente completo. El tipo de configuración que se debe escoger depende del número de sensores que se utilice activamente. En caso de utilizar un único sensor, se escogería la configuración de cuarto de puente de Wheatstone, donde se tendría una resistencia variable (galga) y 3 resistencias fijas (Figura 11).

**Figura 11.** Cuarto de puente de Wheatstone



Nota. Esta imagen fue obtenida de [35].

El valor de las resistencias fijas debe contemplarse en función de amplificar la sensibilidad de lectura del sensor. Esto significa, lograr la mayor variabilidad para el divisor de voltaje al tomar en cuenta el cambio de resistencia esperado para la galga ( $\leq 2\%$ ). En el caso de un medio puente de Wheatstone, esta sería la ecuación que rige para el divisor que incluye al sensor:

$$V_D = \frac{R_3}{R_3 + R_G} \cdot V_{EX} \quad (7)$$

donde  $R_G$  es la resistencia del sensor. Tomando en cuenta una variación de resistencia de  $100 \Omega$  para la galga extensiométrica. Variación significativamente superior a la esperada, y que se emplea con fines ilustrativos para simplificar los cálculos teóricos. Se pueden presentar los siguientes escenarios al tomar resistencias fijas de  $350 \Omega$ ,  $470 \Omega$  y  $1000 \Omega$ . Los voltajes en el nodo  $V_D$  utilizando una galga de resistencia nominal de  $350 \Omega$  y un  $V_{EX}$  de  $3.3 \text{ V}$  se muestran a continuación:

$$350 \Omega \text{ resistencia fija: } 1.540 \text{ V} \xleftarrow{\text{alargamiento}} 1.650 \text{ V} \xrightarrow{\text{contracción}} 1.777 \text{ V}$$

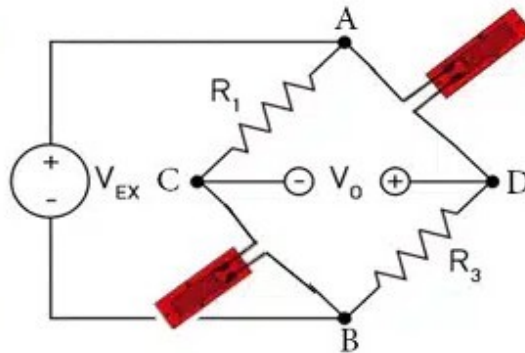
$$470 \Omega \text{ resistencia fija: } 1.783 \text{ V} \xleftarrow{\text{alargamiento}} 1.891 \text{ V} \xrightarrow{\text{contracción}} 2.014 \text{ V}$$

$$1000 \Omega \text{ resistencia fija: } 2.357 \text{ V} \xleftarrow{\text{alargamiento}} 2.444 \text{ V} \xrightarrow{\text{contracción}} 2.538 \text{ V}$$

El cambio en unidades de mV es de 237, 231 y 181, respectivamente en  $V_D$ . Esto aplica también para  $V_o$ , puesto que el voltaje en el otro divisor  $V_C$  se mantiene fijo al estar compuesto de resistencias fijas. Con esto en mente, la resistencia fija en los divisores de voltaje debe ser seleccionada lo más cercana al valor nominal de la resistencia de la galga para tener la mayor ganancia, y así asegurar la mayor sensibilidad captada por cambios al sensor.

Para incrementar la señal de voltaje obtenida por la deformación del sensor en el circuito de Wheatstone, se puede emplear la siguiente configuración, el medio puente de Wheatstone. Este se compone de dos sensores (colocados diagonalmente) y de dos resistencias fijas (Figura 12). Esta configuración duplica la señal de voltaje diferencial entre nodos, mejorando la resolución de la medición.

**Figura 12.** Medio puente de Wheatstone



Nota. Esta imagen fue obtenida de [35].

Al tener esta configuración, se tendrá un comportamiento inversamente proporcional en ambos divisores de voltaje. Con esta configuración, se busca que ambos sensores experimenten el mismo tipo de deformación, ya sea tensión o compresión. Lo cual significa que a medida que el voltaje  $V_D$  incremente, el voltaje en el otro nodo  $V_C$  disminuirá, y viceversa.

Esto debido a que la resistencia que ambos sensores tendrán (en condiciones ideales) será simétrica. A comparación del cuarto de puente de Wheatstone donde únicamente existía variación en un nodo, ahora se amplifica la sensibilidad de lectura al habilitar el otro nodo, y así contar también con esta ganancia.

Tomando en cuenta el ejemplo para el medio puente de Wheatstone, la lectura en  $V_o$  con resistencias fijas de  $350 \Omega$  y  $470 \Omega$  (junto a un  $V_{EX}$  de  $3.3 \text{ V}$ ), tendría los siguiente cambios en voltaje al tener ambos sensores experimentando la misma deformación:

$$350 \Omega \text{ resistencia fija: } -0.220 \text{ V} \xleftarrow{\text{alargamiento}} 0 \text{ V} \xrightarrow{\text{contracción}} 0.260 \text{ V}$$

$$470 \Omega \text{ resistencia fija: } 0.266 \text{ V} \xleftarrow{\text{alargamiento}} 1.464 \text{ V} \xrightarrow{\text{contracción}} 0.728 \text{ V}$$

Esto resulta en cambios de  $480 \text{ mV}$  y  $462 \text{ mV}$ , respectivamente. Esto también representa una sensibilidad del doble (aproximadamente) con respecto a los cambios de voltaje para el medio puente de Wheatstone. Además, cabe recalcar que al obtener un voltaje negativo en el caso de las resistencias fijas siendo igual a las nominales de la galga, se debe hacer una corrección a lo establecido para las resistencias fijas. En este caso, se tendría que seguir la noción de utilizar valores de resistencias fijas cercanos al valor nominal de la galga, mas no igual a la de la galga.

## 6.5. Voltaje de excitación

El voltaje de excitación  $V_{EX}$  es una variable importante dentro del puente de Wheatstone. Este es el que se verá dividido y amplificado para obtener un voltaje de salida. Este se toma empíricamente como  $3.3 \text{ V}$  para los ejemplos provistos en las configuraciones de puentes de Wheatstone debido a que es el voltaje de operación en la mayoría de microcontroladores [37]. En muchas aplicaciones, se busca utilizar este voltaje no solo para alimentar el circuito puente de Wheatstone, sino también para alimentar un microcontrolador, el cual a la vez integra diferentes componentes como pantallas *display*, entre otros [36].

Debido al perfil biomédico de la aplicación de los sensores de deformación, se utilizan baterías, logrando hacer del dispositivo autónomo y no dependiente de una fuente de voltaje. Las baterías de iones de litio (LI-ON) son una opción relevante ahora en día para circuitos de bajo rendimiento [38]. Estas baterías disponibles comercialmente pueden proveer de un voltaje de  $3.7 \text{ V}$ , con capacidades de  $\text{mAh}$  diferentes, dependiendo de la batería que se obtenga.

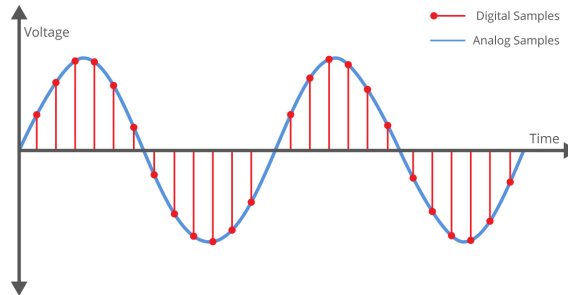
El voltaje de entrada de  $3.3 \text{ V}$  es un voltaje estándar para la alimentación de microcontroladores de baja potencia como el microcontrolador ESP32 [39]. Por esto, la utilización de circuitos integrados o módulos amplificadores/reductores de voltaje son importantes para poder utilizar la batería de LI-ON a pesar de tener circuitos que operen a diferentes voltajes.

## 6.6. ADC

El módulo *analog-to-digital converter* (ADC) de un microcontrolador se refiere al Conversor Analógico-Digital. El propósito de un ADC es poder transformar una señal analógica (es decir, continua en el tiempo) hacia valores digitales que puedan almacenarse temporalmente o permanentemente en la memoria del microcontrolador. Permitiendo así el procesamiento de estos valores, y permitiendo trabajar con ellos. El ADC funciona en base a dos principios, la frecuencia de muestreo y la resolución de bits [40].

La frecuencia de muestreo ( $f_s$ ) se refiere a qué tan fiel será la señal digital a la señal analógica original. Esta dicta con qué periodicidad se estará coleccionando el valor de la señal analógica en varios momentos del tiempo para almacenarlos [41] (Figura 13). Al tener una frecuencia de muestreo alta, la fidelidad será mayor, mientras que una frecuencia de muestreo baja tiende a categorizarse como una simulación de la señal analógica y no una representación exacta.

**Figura 13.** Frecuencia de muestreo y señal procesada



Nota. La imagen muestra el proceso de muestreo de una señal continua. La crua azul representa la señal analógica continua. Los puntos rojos son las muestras digitales que el ADC adquiere con frecuencia  $f_s$ . Esta secuencia de valores es la representación digital de la señal original. Esta imagen fue obtenida de [42].

Esta frecuencia de muestreo depende del microcontrolador. La frecuencia máxima de muestreo está limitada por el reloj (oscilador) interno que impulsa el microcontrolador.

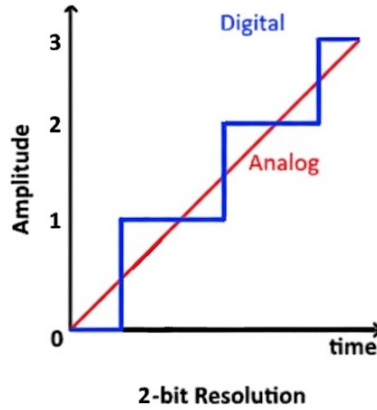
Ahora bien, la resolución de bits toma más relevancia en el ADC, ya que, a diferencia del tiempo de muestreo, que se puede configurar mediante *software*, este tiende a ser un valor ya integrado en el *hardware*. Esto significa que se debe adquirir un microcontrolador considerando la resolución de bits de su ADC. La resolución permite establecer con cuántos puntos decimales se podrá guardar el valor (continuo) correspondiente a la señal analógica. Esta resolución se rige por la siguiente fórmula:

$$\text{Resolución} = \frac{V_{IN}}{2^N - 1} \quad (8)$$

donde  $V_{IN}$  es el voltaje con el que se alimenta el microcontrolador y  $N$  la resolución de bits del ADC.

La resolución es, a manera de visualizarlo, el mínimo cambio en  $V$  que se debe dar en la señal analógica para que se pueda digitalizar como un valor diferente al previo. Existen resoluciones bajas como en el caso de un ADC de 2 bits (Figura 14).

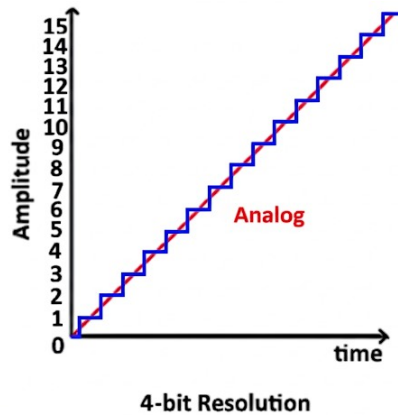
**Figura 14.** Resolución de 2 bits



Nota. Esta imagen fue modificada de [43].

A medida que incremente la resolución del ADC, mayor será la precisión de una lectura de una señal analógica, como en el caso de incrementar a 4 bits (Figura 15). Dependiendo de la necesidad de precisión de la lectura, ya sea que se quieran percibir cambios en escalas pequeñas como mV o  $\mu\text{V}$ , la resolución del ADC debe tener un mínimo de bits.

**Figura 15.** Resolución de 4 bits



Nota. Esta imagen fue modificada de [43].

La resolución se puede interpretar de mejor manera mediante un ejemplo con valores de práctica, tomando en cuenta que muchos de los microcontroladores actuales cuentan con ADC de 10 o 12 bits de resolución [37]. Así, al considerar un voltaje de entrada común como

lo son 5 V, se obtiene lo siguiente:

$$\text{Resolución} = \frac{5 \text{ V}}{2^{12} - 1} = 1.22 \text{ mV}$$

Esto significa que los cambios en voltaje más pequeños que podrá detectar este ADC son por cada 1 mV. Por lo que al tener en mente cambios en escalas de mV, es necesario por lo menos un ADC de 12 bits.

## 6.7. Filtros digitales

Un filtro digital se implementa como una alternativa a la utilización de un filtro analógico. Estos filtros se diseñan en base a la respuesta frecuencial deseada para una señal entrante a un *hardware* y se representan principalmente como una función de transferencia  $H(z)$  [44]. Específicamente, uno de los tipos de filtros digitales son los filtros IIR. En fundamento, estos son la contraparte en tiempo discreto (entiéndase para una señal continua muestreada digitalmente) de los filtros analógicos clásicos [44].

Tomando en cuenta así que una señal DC opera fundamentalmente en una frecuencia de 0 Hz, frecuencias considerablemente arriba de 10 Hz por lo tanto, se consideran como ruido significativo [45]. Para esto, se puede utilizar el filtro IIR conocido como el *leaky integrator*, el cual fue diseñado para ser un filtro pasa-bajas [44].

El filtro digital *leaky integrator* es así un filtro pasa-bajas de primer orden que se describe mediante la siguiente ecuación de diferencias:

$$H(z) = \frac{1 - \lambda}{1 - \lambda z^{-1}} \quad (9)$$

Esto, al ser una función de transferencia, se puede representar como la siguiente ecuación de diferencias.

$$y[n] = (1 - \lambda) \cdot x[n] + \lambda \cdot y[n - 1] \quad (10)$$

donde  $y[n]$  representa el valor de la señal de salida (*output*) que se desea,  $x[n]$  la señal obtenida en ese instante (*input*) e  $y[n - 1]$  la señal de salida en el instante previo. Debido a que el *output* actual o  $y[n]$  únicamente depende del valor de la entrada en el presente  $x[n]$  y del valor del *output* del pasado  $y[n - 1]$ , el sistema se denomina como implementable; a esto se le conoce como un sistema causal. Esto significa que se puede implementar como un algoritmo a nivel de *software*.

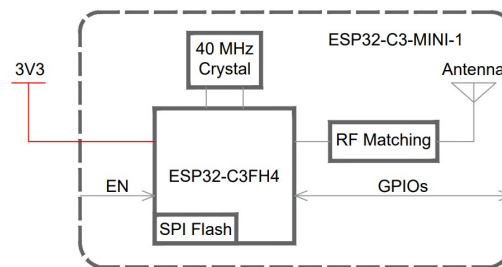
Además, el valor de  $\lambda$  describe el factor de suavizado, o bien, la severidad con la que se suavizará la señal analógica entrante.

## 6.8. Wifi

Los dispositivos compatibles con *wifi* son cada vez más comunes. Muchos microcontroladores actuales son compatibles con comunicación *wifi*, destacando los microcontroladores *espressif systems product* (ESP). Permitiendo así que se logre enviar información hacia una computadora de manera inalámbrica, a diferencia de tener que recurrir a una comunicación física. Para aplicaciones simples, se emplea la banda de 2.4 GHz, permitiendo la transmisión de datos a distancias de hasta 74 m, considerando un uso en el interior de un edificio [46].

La compatibilidad o función *wifi* se obtienen debido a la antena ya integrada en el microcontrolador (Figura 16). Esta permite captar como enviar datos mediante la programación del *software* en el microcontrolador.

**Figura 16.** Diagrama de bloques de microcontrolador ESP32



Nota. Esta imagen fue obtenida de [47].

Se dividió la metodología en las siguientes etapas: Validación del circuito con galga extensiométrica, *post-processing* y caracterización de valores digitales con algoritmo embebido, envío de data hacia la computadora.

## 7.1. Validación de circuito puente de Wheatstone

En esta etapa inicial, se midió la variación de resistencia de la galga extensiométrica adherida a una viga en voladizo con 2 métodos de fijación. Posteriormente, estos valores fueron simulados en LTspice para determinar el voltaje diferencial esperado en una configuración de cuarto de puente de Wheatstone. Finalmente, se observó y cuantificó el ruido analógico del voltaje diferencial obtenido desde el circuito mediante un osciloscopio.

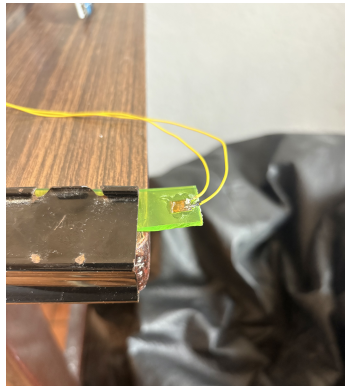
### 7.1.1. Examinación de galga extensiométrica

Inicialmente, se preparó la galga extensiométrica (Electrónica DIY, GUA) para su utilización. Para usar un sensor, de tamaño  $7.4 \times 4.4$  mm y con un factor de galga  $GF = 2$ , se soldaron cables terminales a los *pads* de soldadura de la galga, para su fácil conexión.

Para la primera prueba se adhirió la galga con adhesivo instantáneo comercial (Loctite, CT, USA) al extremo de una pieza delgada de plástico; esto de acuerdo con el material que se quiera examinar para simular el comportamiento del músculo (Figura 17). Además se sostuvo con una abrazadera para poder deformar el extremo de la pieza con la galga, como viga en voladizo. A continuación, utilizando un multímetro, se midió la resistencia al deformar la viga. Esta se deformó ejerciendo una fuerza manual, sin hacer contacto con la galga, hasta alcanzar los  $+90^\circ$  respecto a la horizontal. Luego la medición se repitió hasta

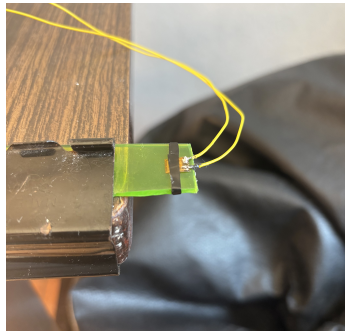
alcanzar los  $-90^\circ$ .

**Figura 17.** Galga adherida con agente adhesivo a viga de plástico



Para comparar el cambio de resistencia obtenido bajo distintos métodos de fijación, se continuó utilizando un pedazo de cinta de aislar para adherir la galga a la viga (Figura 18). Nuevamente se realizaron mediciones con el multímetro al deformar la viga a  $+90^\circ$  y  $-90^\circ$ .

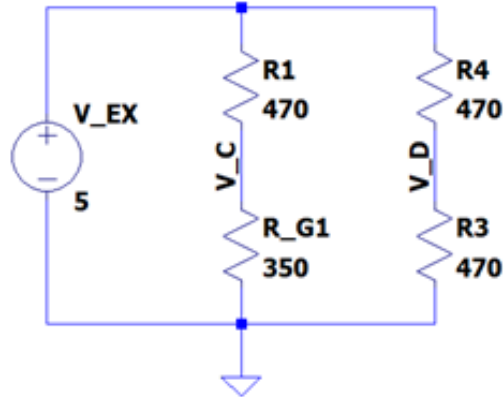
**Figura 18.** Galga adherida con cinta a viga de plástico



### 7.1.2. Simulación de cambios de resistencia

Una vez se cuantificó la variabilidad en resistencia del sensor, se prosiguió a simular los cambios en voltaje que estos representarían en el programa LTspice 1.17.9 (Analog Devices, MA, USA). Utilizando el circuito de cuarto de puente de Wheatstone, se realizó un análisis transitorio para obtener el voltaje en el nodo  $V_C$  al cabo de 10 s (Figura 19).

**Figura 19.** Circuito de cuarto de puente de Wheatstone en LTspice



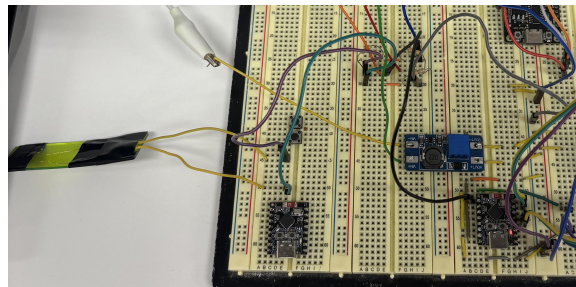
Se efectuaron dos simulaciones del análisis. Una utilizando la resistencia máxima obtenida para el sensor durante su estiramiento al deformar la viga a  $-90^\circ$ , y la otra con la resistencia mínima obtenida durante su compresión al deformar la viga a  $+90^\circ$ .

Los dos voltajes obtenidos se registraron para tener una referencia del cambio en voltaje esperado a nivel experimental.

### 7.1.3. Ruido analógico

En una segunda instancia, se recreó el circuito de cuarto de puente de Wheatstone en una *protoboard*. El circuito se alimentó mediante una fuente de poder, aprovisionando un voltaje de  $5\text{ V}$ . Las resistencias restantes del circuito se seleccionaron de manera práctica, utilizando el valor comercial más cercano al nominal de la galga, en este caso de  $470\ \Omega$ . Así entonces, se diseñó un circuito con voltaje de entrada de  $5\text{ V}$ , 3 resistencias fijas de  $470\ \Omega$  y la galga extensiométrica de  $350\ \Omega$  (Figura 20).

**Figura 20.** Circuito de cuarto de puente de Wheatstone en *protoboard*



Una vez conectado el circuito y alimentado, se continuó a observar su comportamiento mediante un osciloscopio TBS2000 (Tektronix, OR, USA). Debido al cambio en unidades de mV obtenido previamente en simulación, al ser esta una escala muy baja, se utilizaron los dos nodos de salida del puente de Wheatstone, donde la punta de medición se colocó en

el nodo  $V_D$  y la punta de referencia en el nodo  $V_C$ . Se ajustó así la escala de división de tiempo y la escala de voltaje de acuerdo con un criterio visual (Figura 21). Una vez esto configurado y empleando nuevamente el montaje experimental previamente establecido de la galga junto a la viga y abrazadera, se prosiguió a deformar la viga manualmente a ángulos aleatorios con tal de ver picos/cambios en la señal del osciloscopio. Se observó si los picos eran notorios, o bien, si eran opacados por ruido eléctrico.

**Figura 21.** Visualización de señal en osciloscopio con ajuste de escala



Al observar ruido, se continuó a analizar la naturaleza de este sin manipular la galga. Para esto, se extrajo la señal observada en el osciloscopio como un archivo *.csv* directamente desde este, para así analizar estos datos en el *software* Matlab R2023b (The MathWorks, MA, USA). En este último, se graficó la señal obtenida y se determinó la diferencia entre las muestras de la señal con mayor y menor magnitud (ver Anexo 13.1). Los datos del voltaje diferencial extraídos del osciloscopio permitieron estimar el nivel de ruido para así comprender la variabilidad de la señal que sería convertida posteriormente por el ADC del microcontrolador.

## 7.2. *Post-processing*

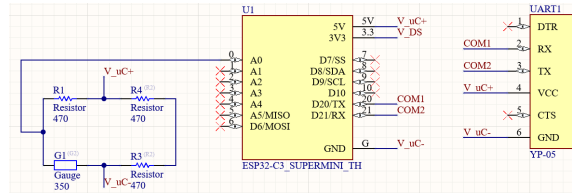
En la siguiente etapa experimental, se utilizó el conversor analógico-digital (ADC) del microcontrolador para comparar la señal del circuito puente de Wheatstone procesada sin filtro con la señal suavizada mediante un filtro digital IIR, atenuando el ruido previamente analizado. Una vez implementado el filtro, se aplicó una deformación angular a la viga en voladizo para procesar y capturar los cambios de resistencia de la galga adherida como cambios de voltaje.

### 7.2.1. Ruido digital

Se utilizó nuevamente el circuito cuarto de puente de Wheatstone junto a la galga adherida a la viga, y se conectó el nodo  $V_C$  del circuito Wheatstone hacia el pin A0 (pin ADC) de

un microcontrolador de selección - ESP32 C3 Supermini (Espressif Systems, CHN). Además, también se agregó al circuito un módulo adaptador USB a Serial UART TTL para poder obtener datos desde el microcontrolador por cable (Figura 22).

**Figura 22.** Esquemático de circuito para procesamiento digital de voltaje en puente Wheatstone



Nuevamente, sin ejercer ninguna fuerza sobre la viga, se analizó el ruido ahora procesado por el microcontrolador. Para esto, se cargó un código en el microcontrolador utilizando la plataforma Arduino IDE 2.3.6 (Arduino Project, USA), obtenida directamente desde la web del proveedor Arduino; en esta, se seleccionó la placa ESP32C3 Dev Module como el *board* a programar. En el código cargado se utilizó la función ADC y se leyó el voltaje en el pin A0 continuamente. Enviando estas lecturas digitales hacia el módulo UART mediante la librería *HardwareSerial.h* (ver Anexo 13.2).

El módulo UART se conectó directamente con cable USB a la computadora. Desde esta se ejecutó un código en el programa Matlab, el cual recibía y graficaba 10 s de los datos recibidos por comunicación serial (ver Anexo 13.3).

### 7.2.2. Filtro digital

Utilizando el mismo circuito y procedimiento experimental que en el paso previo, se continuó ahora aplicando un filtro digital a la señal procesada en el pin A0. Esto para comparar la señal digital con ruido a las señales con un filtro. Más a detalle, al aplicar el filtro digital, conocido como *leaky integrator*, se buscaba suavizar y estabilizar la señal del puente de Wheatstone con el sensor aún en reposo.

Para esto, se realizó un único cambio al código cargado al microcontrolador, en el cual se establecía que el valor de  $\lambda$  pasaría a tener un valor mayor a 0, permitiendo así que el filtro digital tuviera un efecto sobre la señal procesada (Cuadro 1).

**Cuadro 1.** Declaración de valor de  $\lambda$  en el código

```
float lambda = 0.8;
```

El filtro digital se implementó en el código provisto como una ecuación de diferencias (Cuadro 2). Como tal,  $y_n$  era el valor corregido (el de interés), y  $\lambda$  es una constante con valor entre 0 y 1. Dependiendo de la severidad con la que se quiere que actúe el filtro, se colocaba un valor cercano a 0 para menor efecto y un 1 para un mayor efecto.

Se realizaron 2 repeticiones del experimento, obteniendo dos señales nuevas en la compu-

## Cuadro 2. Implementación de filtro digital como ecuación de diferencias

```
y_n = (1 - lambda) * x_n + lambda * y_n_1;  
y_n_1 = y_n;
```

tadora. Para estas pruebas, los ensayos se llevaron a cabo con un  $\lambda = 0.8$ , y posteriormente con un  $\lambda = 0.99$ . Las señales resultantes se graficaron en el programa Matlab junto a la señal digital con ruido previamente guardada; para cual  $\lambda = 0$ .

### 7.2.3. Evaluación del sensor con filtro digital

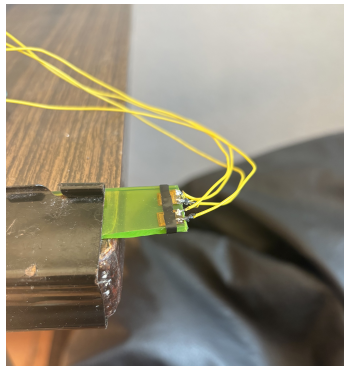
Una vez implementado el filtro digital, se realizó experimentación manipulando la galga junto al filtro. Para esto nuevamente se empleó el sistema compuesto por el circuito puente de Wheatstone junto a la galga adherida a la viga (con cinta), microcontrolador y módulo UART (Figura 22).

Para la primera parte de la prueba, se cargó el código al microcontrolador para que este fuese nuevamente capaz de obtener la lectura del pin A0, aplicar el filtro a esta señal con  $\lambda = 0.99$ , y enviar esta data al módulo UART continuamente (Anexo 13.2). En la computadora, se recibió esta información durante 10 s al ejecutar el código de Matlab (Anexo 13.3).

Esta captura de información se realizó en dos ejecuciones. Durante la primera ejecución, se deformó la viga hacia  $-90^\circ$  por 5 s y se deformó la viga hacia  $+90^\circ$  por los otros 5 s restantes. En la segunda ejecución no se manipuló la viga, capturando así la señal sin interferencia.

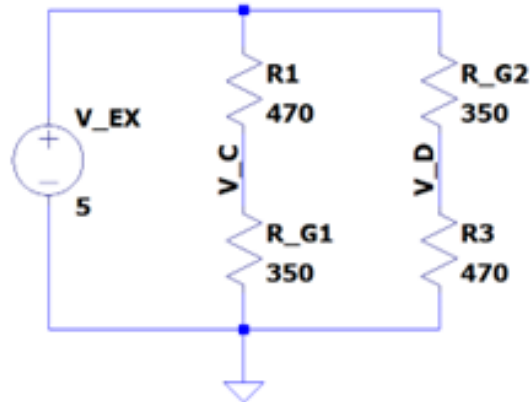
En la segunda parte de la prueba, se repitió el procedimiento anterior, con la excepción de que ahora se utilizaron dos galgas extensiométricas en simultáneo. Para esto, se adhirieron dos galgas extensiométricas a una misma viga con un pedazo de cinta de aislar (Figura 23).

**Figura 23.** Dos galgas adheridas con cinta a viga de plástico



Esto también supuso un cambio al sistema utilizado anteriormente, puesto que para leer cambios en dos sensores a la vez, se utilizó un medio puente de Wheatstone (Figura 24).

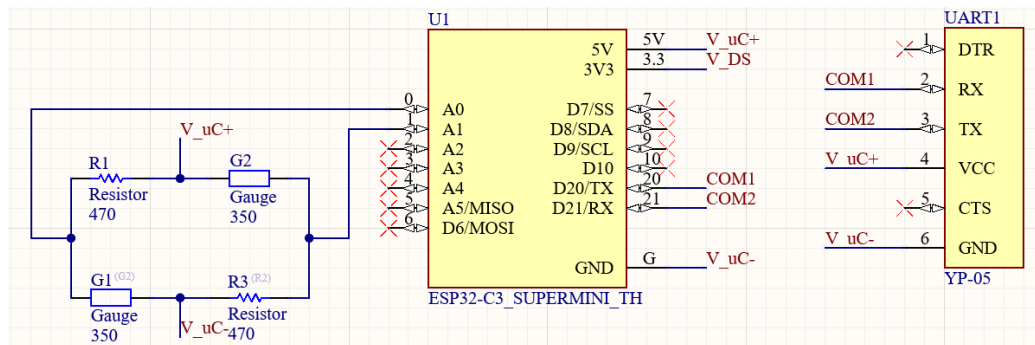
**Figura 24.** Circuito de medio puente de Wheatstone



Al tener el circuito configurado, se cargó al microcontrolador un código diferente. En este caso, el microcontrolador leía las señales provenientes de los nodos  $V_C$  y  $V_D$  en los pines A0 y A1 (Figura 25). Se realizaba una resta entre ambas señales y a esta señal resultante se le aplicaba el filtro digital con  $\lambda = 0.99$ , enviando así esta continuamente al módulo de comunicación serial (ver Anexo 13.4). Esta señal se obtuvo durante 10 s en la computadora para graficarla.

Se realizaron dos ejecuciones. Durante la primera, se deformó la viga a  $-90^\circ$  por 5 s y seguido a  $+90^\circ$  por los otros 5 s. En la segunda, se capturó la señal sin manipulación alguna de la galga.

**Figura 25.** Esquemático de circuito para procesamiento digital de voltaje con dos galgas

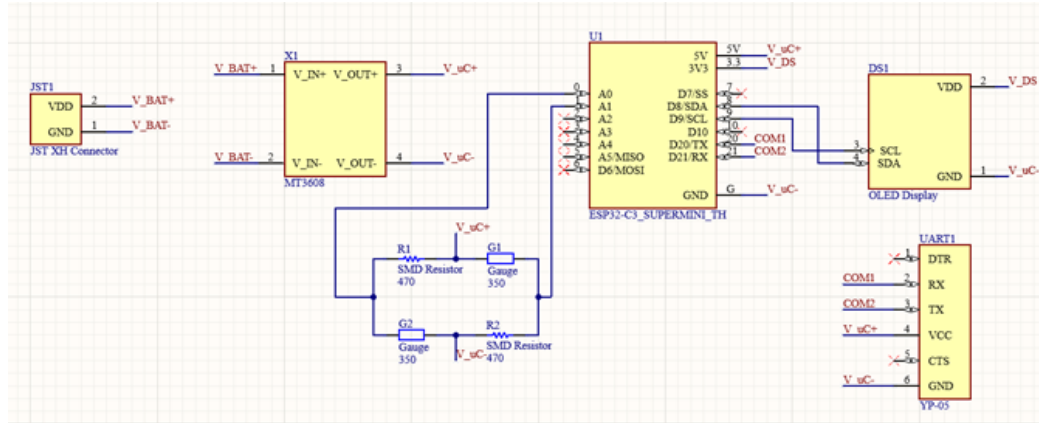


### 7.3. Caracterización

Habiendo obtenido gráficas demostrando los cambios en pasos ADC/voltaje obtenidos con el sensor y la lectura de valores procesados en el microcontrolador, proseguimos a diseñar un circuito que incluyó una pantalla OLED; para este paso ya no se utilizó el módulo UART, además de que se agregó una pantalla OLED 1.3" SH1106 I2C (Electrónica DIY, GUA).

Así también se agregó una batería de 3.7 V y 1000 mAh (Tettsa, GUA), y un módulo amplificador de voltaje MT3608 (La Electrónica, GUA) con el propósito de introducir una alimentación independiente para el circuito (Figura 26).

**Figura 26.** Esquemático de circuito completo para lectura y caracterización de cambios en puente Wheatstone



Se programó nuevamente un código a cargar al microcontrolador, buscando que la diferencia entre los límites alcanzados experimentalmente y la lectura se pudieran mostrar en una gráfica ADC vs tiempo, en tiempo real. Los límites utilizados en la programación se obtuvieron desde las gráficas obtenidas previamente y se insertaron como una suma y resta de unidades respecto al ADC promedio calculado cada 10 s; dependiendo del cambio en voltaje percibido experimentalmente, estos se pueden ajustar (Cuadro 3).

**Cuadro 3.** Declaración de límites para caracterización

```
adcMin = adcAvg - 1.5;
adcMax = adcAvg + 1.5;
```

Entonces, mediante la función I2C del microcontrolador, se mostró la caracterización en la pantalla. Para utilizar esta comunicación I2C, se emplearon las librerías *U8g2lib.h* y *Wire.h*, las cuales permitieron una mayor practicidad para programar la información a desplegar en la pantalla (ver Anexo 13.5).

Una vez cargado el código, se realizaron pruebas deformando una única viga, con la galga adherida con cinta, a  $+90^\circ$  y  $-90^\circ$ , para apreciar los cambios en la señal ahora en la pantalla.

## 7.4. Diseño de PCB

Con base en la caracterización obtenida, se midió el consumo de potencia al registrar la corriente del circuito bajo deformación continua de la viga y con el *display* en funcionamiento. Posteriormente, utilizando esta medición de corriente y adhiriéndose a las reglas de diseño para el ancho de pista (*track width*) así como reglas de diseño del fabricante, se diseñó y se fabricó una PCB que integró el prototipo desarrollado en la *protoboard*.

### 7.4.1. Análisis de potencia

Al tener la caracterización, se utilizó el circuito previamente diseñado, con excepción de la batería y el módulo rectificador de voltaje. En su reemplazo, se empleó una fuente de alimentación que proporcionó en tiempo real la información de voltaje y corriente siendo aprovisionados al circuito.

Así, deformando la galga continuamente a ángulos mayores a  $45^\circ$  y  $-45^\circ$  de manera continua, se registró la corriente de mayor magnitud reportada por la fuente. Esta corriente correspondería a tener la deformación del sensor y la caracterización de esta en tiempo real en simultáneo.

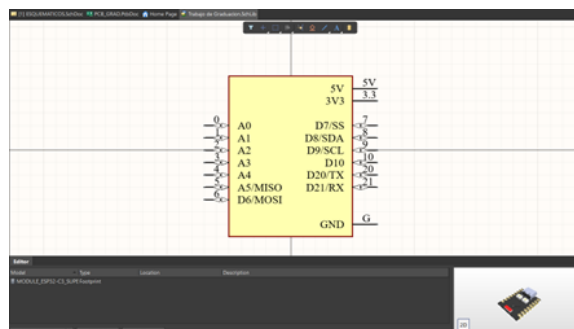
### 7.4.2. Diseño y fabricación de PCB

Una vez completado el circuito de la Figura 26 en físico, y habiendo logrado validar el circuito de Wheatstone, la implementación del filtro digital *leaky integrator* y la caracterización de la lectura, se prosiguió a traducir este circuito, hasta el momento empleado en una *protoboard*, hacia una PCB.

Para esto, se utilizó el software de diseño de PCB Altium Designer Professional 23.2.1 (Altium, CA, USA). Lo primero fue crear los componentes que se utilizaban en el circuito como el microcontrolador, *headers* para las galgas, resistencias SMD, *headers* para el *display*, conector JST-XH para la batería, *pads* para el elevador de voltaje y *pads* para la posibilidad de conexión del módulo UART. Para esto se creó el esquemático y el cuerpo 3D de cada componente.

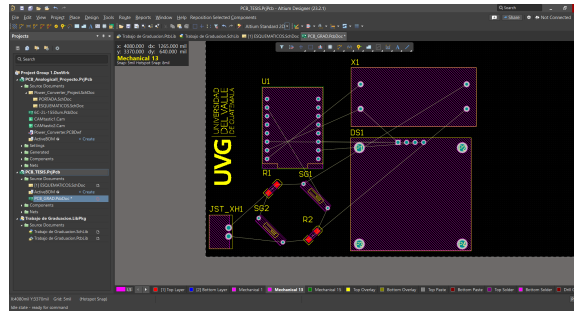
Una vez se crearon los componentes, se creó un archivo de esquemáticos (*.SchDoc*) para en este unir los esquemáticos de los componentes hacia su cuerpo 3D (Figura 27).

**Figura 27.** Creación de esquemático y cuerpo 3D



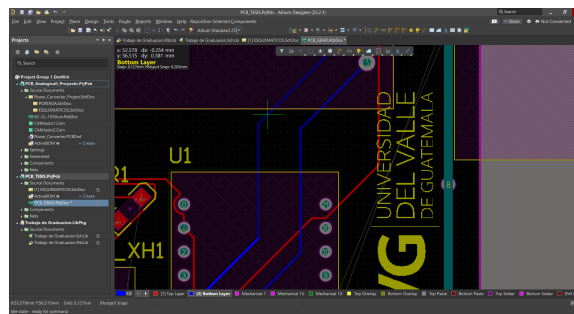
Una vez unidos, se creó la PCB en un archivo *.PcbDoc* con la función automática del programa. Ahora solo quedó organizar los componentes como se quisieran en la placa, así como definir el tamaño y forma de la placa, la cual fue una placa de  $5 \times 5 \text{ cm}$  rectangular (Figura 28).

Figura 28. PCB con componentes



Una de las últimas etapas fue cablear los componentes ya en la PCB (Figura 29). Para esto, se tuvo que tomar en cuenta la corriente que consumió el circuito. Con la corriente medida, se utilizó la calculadora ANSI PCB Trace Width Calculator (disponible en la web) para poder estimar el ancho que debían tener los *tracks* de la placa.

Figura 29. Routing para los componentes en la PCB diseñada



Finalmente, se seleccionó a JLCPCB como el fabricante de PCB y se leyeron las reglas de diseño para poder fabricar en dicho sitio. Para esto, se pudo recurrir a visualizar las reglas de diseño establecidos por el fabricante en la página web. Estas reglas se ingresaron manualmente al software mediante la opción *design rules*, y se corrió un *design rule check* (DRC) para asegurarse que todas las reglas se cumplieran.

Con el DRC aprobado, se prosiguió a exportar los archivos *gerber* y *drill*. Para que así se pudieran enviar estos a la fábrica JLCPCB y se pudiera realizar la PCB. Como último paso, habiendo obtenido la PCB, se soldaron los componentes a esta, exceptuando el módulo UART, cuyos pines también se utilizaron para implementar un *push button* al circuito cuando no se utilizara este. Se validó entonces que la operación fuera la misma a la que se obtuvo a nivel de *proto board*. Esto se hizo corriendo el código *.ino* de la caracterización en la pantalla OLED y observando que la caracterización tuviese el mismo comportamiento que al mostrado anteriormente en pruebas.

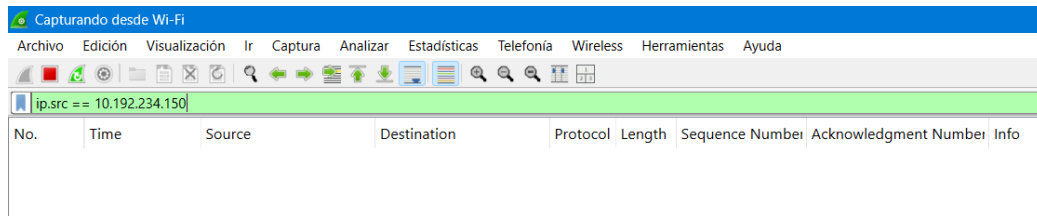
## 7.5. Envío de data hacia computadora

Así entonces, llegado a este punto, se prosiguió a utilizar la capacidad *wifi* del microcontrolador mediante las librerías *WiFi.h* y *WiFiUdp.h* para el envío de datos hacia la computadora. Para esto también se realizó un código de Python en complemento al código del microcontrolador, esto para la obtención y lectura de los paquetes enviados por el microcontrolador en la computadora vía la librerías *pyqtgraph*.

Inicialmente, se estableció la utilización de un *hotspot* móvil para crear una red de área local (LAN, por sus siglas en inglés) aislada, permitiendo la comunicación exclusiva entre el microcontrolador y la computadora. Se programó al microcontrolador para que pudiera conectarse a una red *wifi* utilizando como parámetros el nombre de la red (SSID, por sus siglas en inglés) y la contraseña de esta. Una vez conectado a la red, el microcontrolador mostraría en la pantalla OLED la IP obtenida dinámicamente. Esto sería importante para que, en el código de Python ejecutado en la computadora, se pudiera definir desde qué IP obtener la información. Una vez mostrada la IP, el microcontrolador empezaría a enviar la lectura de la señal obtenida en el pin A0 y filtrada con un  $\lambda = 0.99$  continuamente hacia la IP de la computadora mediante el protocolo UDP (ver Anexo 13.6).

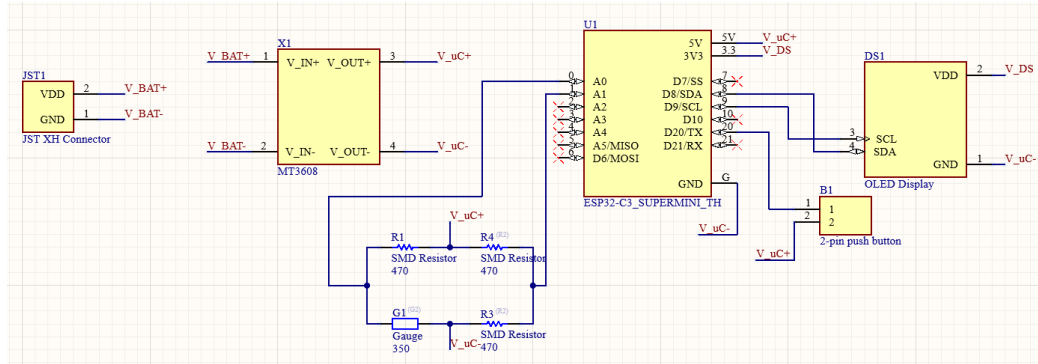
Ahora bien, del lado de la computadora, se validó una comunicación correcta de los datos transmitidos desde el microcontrolador. Para esto, se utilizó el programa Wireshark 4.4.7 (Wireshark Foundation, USA). En este, se capturó la comunicación en la interfaz *wifi* y se colocó un filtro para únicamente monitorear los paquetes provenientes de la IP del microcontrolador (Figura 30).

**Figura 30.** Aplicación de filtro en Wireshark



Tomando en cuenta además la posibilidad de la implementación del *push button* en la placa, se utilizó este en los pines VCC y RX reservados hasta el momento para comunicación serial (Figura 31).

**Figura 31.** Esquemático de circuito en PCB modificado para envío de señal por *wifi*



Utilizando el botón, ya definido en el código cargado al microcontrolador, se logró que durante un periodo de 30 s se enviaran paquetes por 10 s desde el microcontrolador, y al presionar el botón, se pausara la comunicación por 5 s. Se realizó esta operación dos veces.

Utilizando la función **gráficas de E/S** del programa Wireshark, se obtuvo a manera de gráfica la recepción de paquetes sobre tiempo durante esta experimentación. Esta gráfica también se exportó como un archivo *.csv* para el análisis del promedio de paquetes obtenidos en la computadora durante el envío de data en los 30 s.

Además, se seleccionó un paquete arbitrariamente de todos los obtenidos durante estos 30 s para analizar la data que contenía. Se tradujo esta data de código ASCII a un valor numérico para corroborar si se estaba obteniendo un valor de voltaje razonable por parte del microcontrolador. Para esto, se utilizó durante la prueba de los 30 s una resistencia fija de 330  $\Omega$  en el circuito de cuarto de puente de Wheatstone en lugar de la galga.

Al asegurar una comunicación confiable de los datos, se ejecutó el código *.py* que escudaría información de la IP proveniente del microcontrolador, extraería el valor de la señal de los paquetes y se graficarían estos datos como una gráfica de voltaje vs tiempo en tiempo real (ver Anexo 13.7). Para esto, se utilizó la librería *pyqtgraph* de Python.

Finalmente, se deformó la viga con la galga adherida con cinta nuevamente a  $+90^\circ$  y a  $-90^\circ$  durante un lapso de 10 s, y se capturó la imagen de la señal obtenida desde la computadora.

## Resultados

Para examinar el rango de variación de resistencia obtenido por parte de la galga extensiométrica, se llevó a cabo una prueba inicial. Aplicando fuerza a la viga para deformar esta hacia  $90^\circ$  y  $-90^\circ$ , respectivamente, se obtuvieron los valores de resistencia que se detallan en el Cuadro 4. Al estar fijada con un adhesivo comercial, el sensor reporta una resistencia de  $348.0 \Omega$  al deformarse junto a la viga a  $90^\circ$  y una resistencia de  $354.0 \Omega$  al deformar la viga a  $-90^\circ$ . En contraste, al fijar la galga con cinta adhesiva y deformar la viga completamente hacia arriba, se obtiene una lectura de  $351.0 \Omega$ , mientras que al deformar la viga hacia abajo, se obtiene una lectura de  $352.1 \Omega$ .

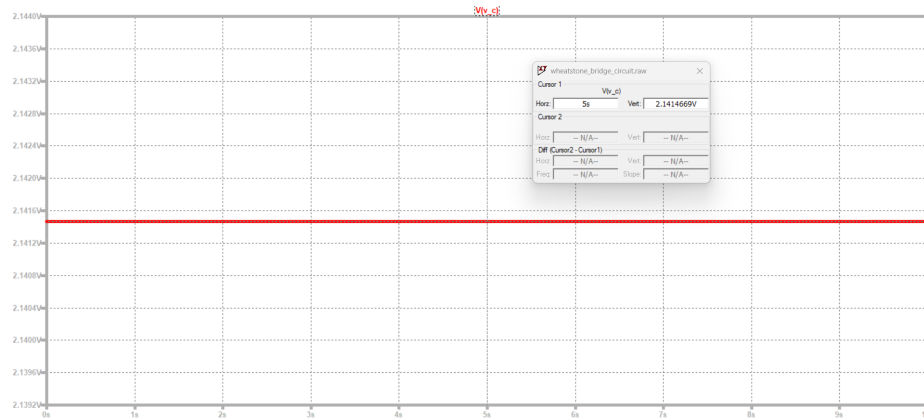
**Cuadro 4.** Valor de resistencia obtenido en la galga bajo deformación para diferentes métodos de fijación

Método de Fijación	Valor nominal ( $\Omega$ )	Compresión ( $\Omega$ )	Estiramiento ( $\Omega$ )
<b>Agente adhesivo</b>	350	347.7	352.7
<b>Cinta adhesiva</b>	350	351.0	352.1

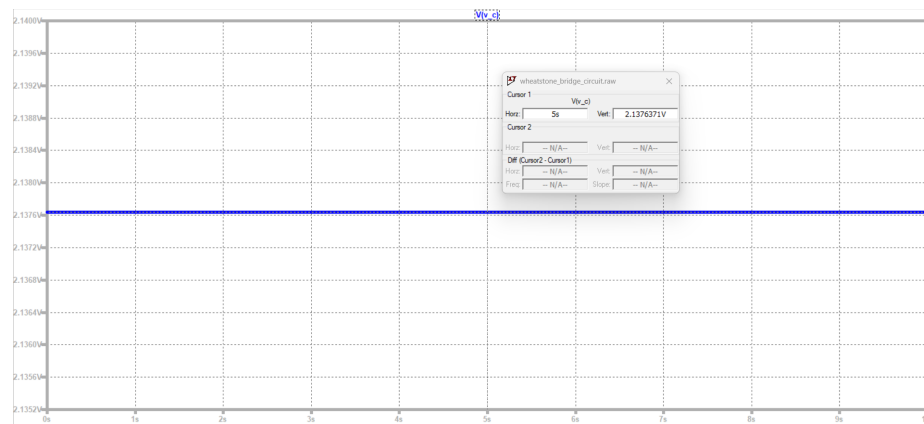
La diferencia entre el valor máximo de resistencia y el mínimo de resistencia obtenidos experimentalmente representa una variación entre  $0 - 1.1 \Omega$  a medida que se deforma el sensor estando sujeto con cinta a un material flexible como el plástico. Esto es un  $0.31 \%$  de variación respecto al valor nominal.

Al realizar simulaciones en el programa LTspice para el cuarto de puente de Wheatstone con  $V_{EX} = 5 \text{ V}$ , resistencias fijas de  $470 \Omega$  y tomando en cuenta las variaciones anteriores de resistencia, se tradujeron los valores de la resistencia mínima y máxima obtenidos hacia valores en términos de voltaje en el nodo del sensor  $V_C$ . Los voltajes resultantes son de  $2.1376 \text{ V}$  y  $2.1415 \text{ V}$  según se aprecia en las Figuras 32 y 33, respectivamente. Representando esto así, cambios en voltaje entre  $0 - 3.9 \text{ mV}$  a medida que se deformaba el pedazo de plástico.

**Figura 32.** Lectura de voltaje en nodo  $V_C$  para sensor adherido con cinta en estiramiento

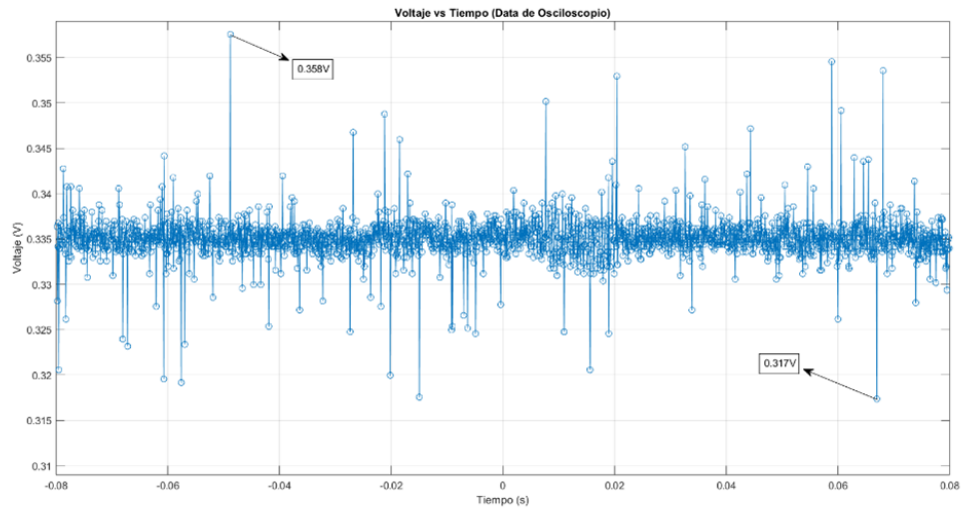


**Figura 33.** Lectura de voltaje en nodo  $V_C$  para sensor adherido con cinta en compresión



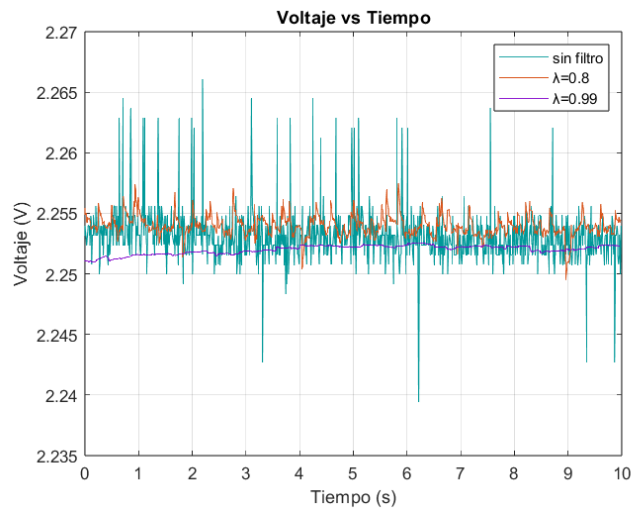
Al extraer la lectura desde el osciloscopio del voltaje diferencial obtenido en el cuarto de puente de Wheatstone, se pudo apreciar en la Figura 34 que, al tener el sensor en reposo, la variabilidad más grande en las lecturas en diferentes puntos de tiempo es de 0.317 V y 0.358 V. Esto representa un rango de variabilidad de 41 mV en la lectura analógica de la señal.

**Figura 34.** Lectura de diferencia de voltaje entre nodos sin manipulación de galga extensiométrica en configuración de cuarto de puente de Wheatstone



En la Figura 35 se detalla la diferencia entre las señales obtenidas desde el circuito puente de Wheatstone, ahora ya siendo procesadas por el microcontrolador, a medida que se aplicaba el filtro con una mayor magnitud para  $\lambda$ .

**Figura 35.** Lectura del voltaje posprocesamiento digital sin manipulación de la galga extensiométrica



Más adelante, también se pudieron contabilizar estos cambios a modo de apreciarlos numéricamente. Esta permitió diferenciar entre el uso del filtro digital y cuando no se utilizaba (Cuadro 5). Al no implementar el filtro, el microcontrolador procesa la señal con un rango de variabilidad de 26.6 mV para la salida del puente de Wheatstone, mientras que al usar

el filtro digital con  $\lambda = 0.99$ , esta variabilidad se reduce a 1.6 mV.

**Cuadro 5.** Variabilidad de lecturas en unidades de voltaje para diferentes implementaciones del filtro digital

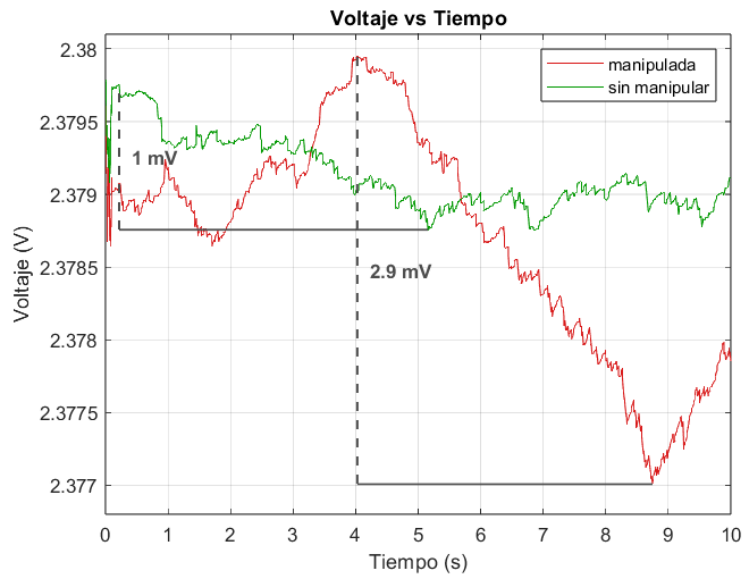
$V_{pp}$ ( <i>sin filtro</i> )	$V_{pp}$ ( $\lambda = 0.80$ )	$V_{pp}$ ( $\lambda = 0.99$ )
26.6 mV	8.0 mV	1.6 mV

Nota. Los valores representan la diferencia entre el valor máximo y mínimo registrado para cada una de las señales con diferentes condiciones de filtrado.

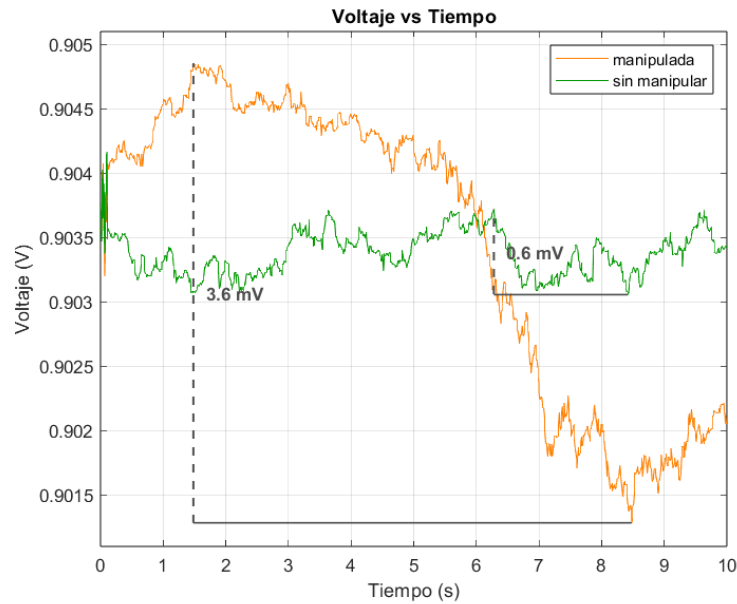
En detalle, el cuadro muestra que el filtro de  $\lambda = 0.99$  presenta una variabilidad en pasos de aproximadamente 1 mV.

Una vez comprobado el filtro digital y su implementación para la lectura eficiente de la señal obtenida del puente de Wheatstone, se evaluó el comportamiento del sensor al manipularlo controladamente. En la Figura 35 se muestra la variación de la señal de voltaje procesada por el microcontrolador y filtrada digitalmente al manipular la viga de plástico mediante una fuerza manual; esta fuerza se aplicaba hasta que el material se doblase a  $90^\circ$  y  $-90^\circ$ . El rango de medición se puede aproximar entre 2.377 V y 2.380 V, dando a entender que se presenta un rango de operación de entre 0 – 2.9 mV al manipular el sensor. De otro lado, al no ser manipulada, la galga presentó cambios únicamente entre 0 – 0.9 mV.

**Figura 36.** Lectura de voltaje procesada por microcontrolador con y sin manipulación de la galga extensiométrica sujeta a la viga



**Figura 37.** Lectura de voltaje procesada por microcontrolador con y sin manipulación de la galga extensiométrica utilizando dos galgas en paralelo adheridas a la viga con cinta



Nota. La imagen ilustra el contraste entre la señal obtenida al deformar la viga y la de referencia. La señal verde representa la señal filtrada con un  $\lambda = 0.99$  sin deformar la viga. La señal naranja representa la señal filtrada con un  $\lambda = 0.99$  al deformar la viga con los dos sensores.

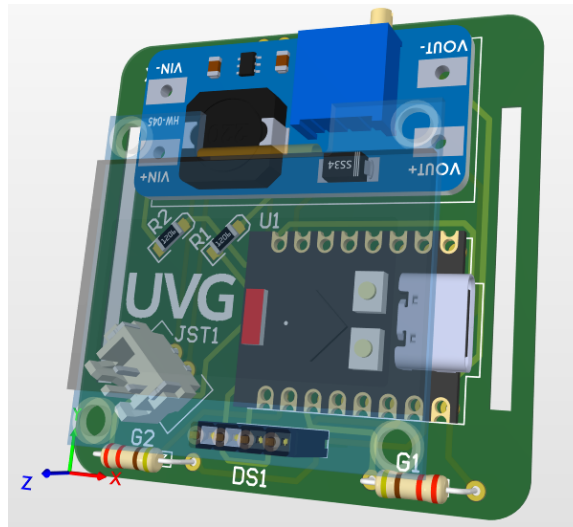
Tomando en cuenta los requerimientos del prototipo, se abordó el análisis de la potencia consumida. Integrando el circuito puente de Wheatstone con el procesamiento y filtrado digital del microcontrolador, junto a la caracterización de la señal del sensor en una pantalla OLED, se realizaron mediciones a partir de los datos entregados por la fuente de voltaje. El voltaje entregado fue de 5 V y la corriente máxima reportada de 39 mA, ver Cuadro 6. A la vez, esto equivale, utilizando la batería LI-ON de 3.7 V y 1000 mAh, a un uso de hasta 18.97 h.

**Cuadro 6.** Medición de potencia

Voltaje entregado	Corriente máxima reportada	Potencia	Tiempo de uso de batería
5 V	0.039 A	0.195 W	18.97 h

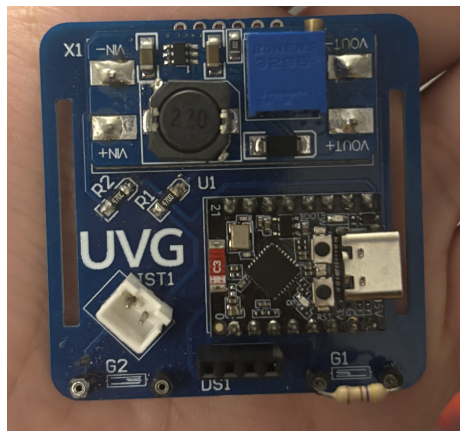
Continuando hacia el diseño del circuito impreso, se tendría de referencia un voltaje de 5 V y una corriente máxima de 39 mA, dejando un margen de hasta 100 mA. Considerando un espesor de cobre común de 1 oz/ft<sup>2</sup> y una temperatura de trabajo de 25 °C, se obtuvo un ancho de 0.61 mm para las pistas de la PCB de acuerdo con las reglas ANSI para el ancho de trazas. Agregando a esto, se diseñó la PCB con dimensiones mínimas, asegurando un espaciado eficiente para los varios componentes y la posibilidad de conexión entre pines de componentes requeridos. El resultado es la PCB que se detalla en la Figura 38.

**Figura 38.** PCB con componentes en software Altium



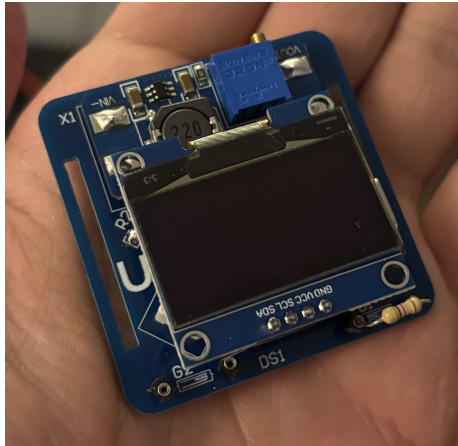
La PCB contó con un tamaño final de 5x5 cm, agrupando todos los componentes de una manera inteligente para hacer de este prototipo lo más compacto posible, como se ejemplifica en la Figura 39. Además, incluye una pantalla OLED modular para que se pueda quitar y poner de acuerdo con la preferencia del usuario como se muestra en la Figura 40. Adicionando, pines extra para la integración de un módulo UART para la lectura de información por medio serial, permitiendo siempre una comunicación física directa con el microcontrolador. Finalizando con la integración de *headers* para la adición fácil de los sensores/galgas; esto es, para que se pueda utilizar un cuarto de puente de Wheatstone, o bien, un medio puente de Wheatstone de acuerdo con la disponibilidad y empleo de sensores.

**Figura 39.** PCB real con componentes soldados



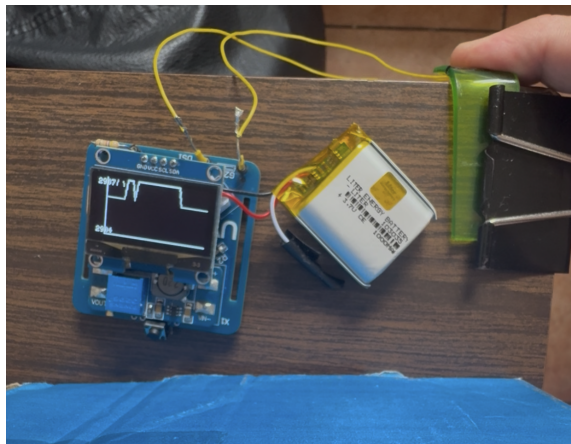
Nota. La imagen muestra la PCB final con cada uno de los componentes soldados.

**Figura 40.** PCB real con *display* integrado

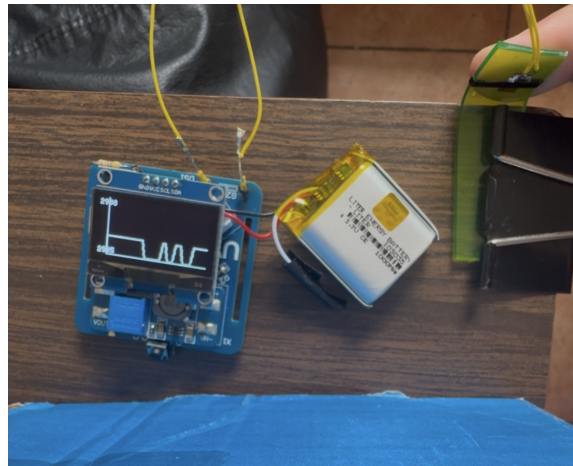


Soldados los componentes a la placa y validando el funcionamiento autónomo del prototipo, se realizaron pruebas para validar el funcionamiento de la PCB directamente con la caracterización en la pantalla OLED. La deformación de la galga se pudo apreciar entre los valores ADC de 2904 y 2907 como se aprecia en las Figuras 41 y 42 al manipularla entre  $90^\circ$  y  $-90^\circ$ . Esto corresponde a valores de 2.3402 V y 2.3426 V, y por ende un cambio apreciable visualmente entre 0 – 2.4 mV en tiempo real.

**Figura 41.** Caracterización del sensor en deformación hacia abajo

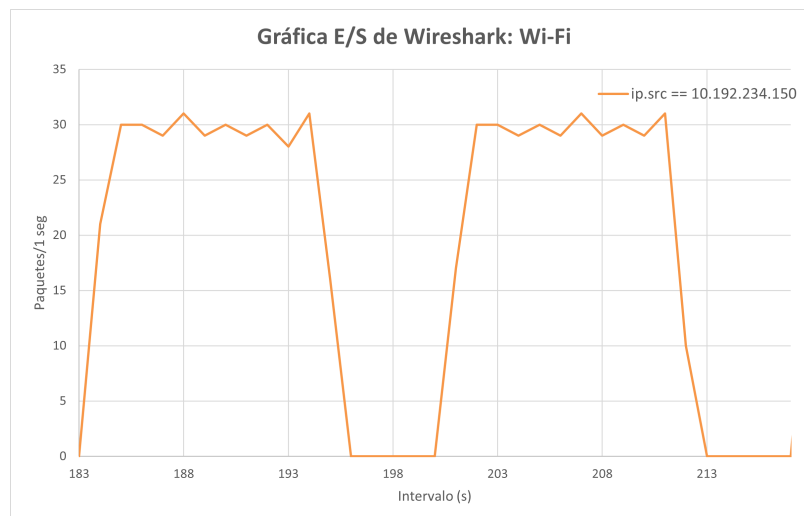


**Figura 42.** Caracterización del sensor en deformación hacia arriba



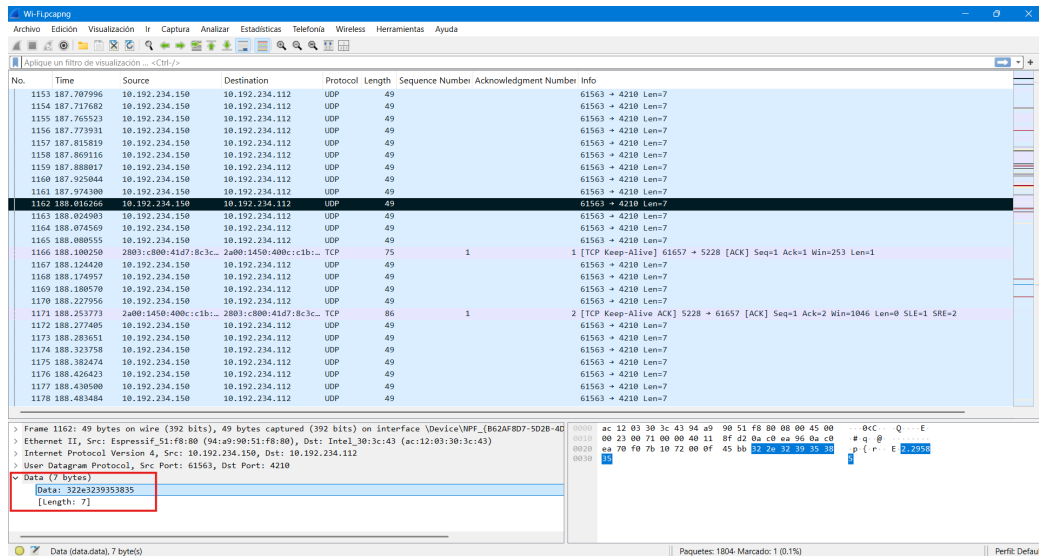
Respecto al envío de información de forma inalámbrica, se estableció una conexión de la computadora y microcontrolador ESP32 hacia una misma red *wifi*. Los paquetes que contenían las lecturas del sensor fueron enviados desde el microcontrolador mediante protocolo UDP hacia la computadora. En la Figura 43 se detalla el tráfico continuo durante 30 segundos de paquetes recibidos por la computadora, filtrados mediante la IP del microcontrolador en el programa Wireshark. En la gráfica se detalla una transmisión continua de 27.46 paquetes/segundo.

**Figura 43.** Gráfica de tráfico entrante por interfaz *wifi* de computadora filtrado para paquetes provenientes de microcontrolador



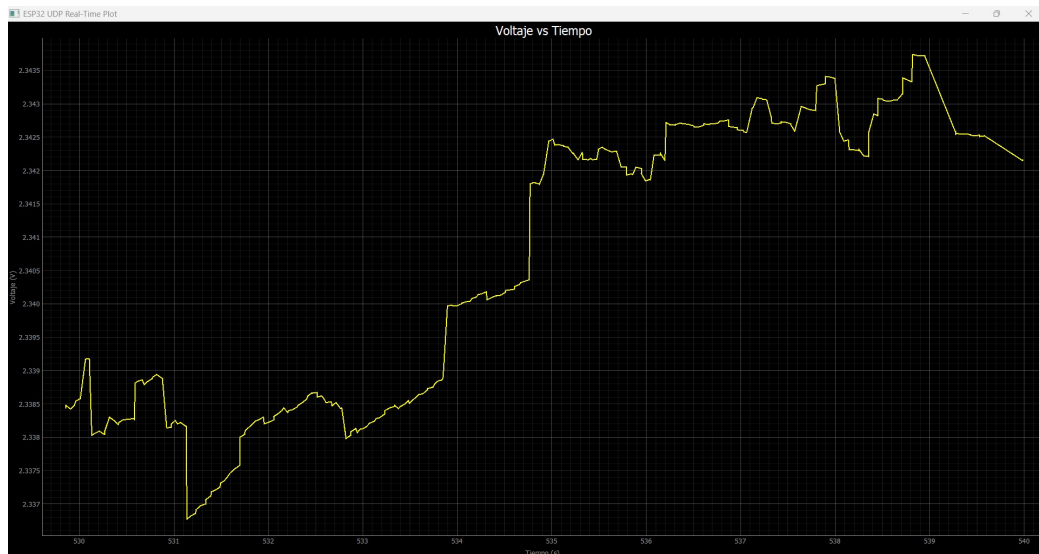
Al analizar uno de estos paquetes por individual, del grupo de paquetes siendo enviado cada segundo, se pudo determinar que se envía cada paquete con un tamaño total de 49 bytes. Para el cual, 7 bytes de estos pertenecen al valor de voltaje de interés, enviado como un valor ASCII (Figura 44).

**Figura 44.** Análisis de paquete entrante



Al realizar el código de Python (desarrollado con asistencia de inteligencia artificial) capaz de obtener estos paquetes y extraer el valor de voltaje dentro de ellos, se realizó un gráfico de voltaje en tiempo real. En el resultado se puede ver en la Figura 45, para la cual se aprecian los cambios al deformar el sensor adjunto a la viga de plástico durante 10 s.

**Figura 45.** Gráfica de data obtenida de paquetes en computadora en tiempo real



La utilización de la galga extensiométrica junto a la viga de plástico en voladizo demuestra tener una deformación que se representa como un cambio de resistencia de  $1.1 \Omega$  con respecto al valor nominal del sensor. Esto permite considerar que, al tener un material flexible como el plástico, se tiene una superficie adecuada para percibir una deformación de compresión y alargamiento. Tomando en cuenta las propiedades plásticas del músculo esquelético [26], así como la representación de las fibras musculares como vigas [5], se tiene una idea sobre la posibilidad de utilización sobre una superficie muscular.

Esta variabilidad del 0.31 % en la resistencia del sensor sugiere que se es capaz de leer la deformación de un material con la galga al estar únicamente adherida esta con un adhesivo como la cinta aislante. En adición, el uso de la cinta adhesiva resulta en una disminución en la eficiencia de la transferencia de deformación, obteniendo un 22 % de la variación en resistencia de aquella registrada con el agente adhesivo. Esto permite comprender el cambio a esperar utilizando el sensor únicamente con un método de fijación temporal, el cual sigue concordando con el máximo de 2 % de cambio en resistencia planteado por experimentación previa sobre el uso del sensor [32]. Esto también conduce a la realización de que la galga no necesita estar pegada con un pegamento especializado al material para presentar una deformación significativa, sino que una sujeción con cinta a la superficie es suficiente para transferir la deformación. Este resultado provee, entonces, conocimiento sobre la aplicabilidad de la galga en un uso no industrial.

Analizando los resultados del circuito, la señal obtenida del puente de Wheatstone con la galga en reposo evidencia un ruido considerable, de 41 mV a nivel analógico y 26.6 mV a nivel digital, superando los valores de cambio previstos al deformar la galga, estimados en hasta 3.83 mV.

Empleando el sistema propuesto, compuesto por la galga extensiométrica en conjunto con el circuito de cuarto de puente de Wheatstone, conectado a un módulo ADC de 12 bits

de un microcontrolador ESP32-C3 Super Mini y aplicando el filtro digital *leaky integrator* con la mayor severidad de  $\lambda$  (0.99), se demuestra confiabilidad en la detección de variaciones en la escala de mV; arreglo para el cual la mayor variabilidad de lectura obtenida es de 1.6 mV. Esto permite así detectar los cambios generados al deformar la viga, los cuales se obtuvieron en un rango de hasta 2.9 mV. Estos resultados contribuyen así a una comprensión de los requisitos mínimos necesarios para la identificación fiable de la deformación de una galga extensiométrica.

Al desarrollar la caracterización para la deformación del material, se utilizó una pantalla OLED. Esta se logró adecuar correctamente junto al sistema de medición de la galga, integrándola en una placa PCB compacta con alimentación independiente. Con lo mostrado en la pantalla, es evidente que, al emplear el filtro digital, se obtiene una señal limpia y estable, y de comportamiento en tiempo real detallando los cambios en la señal del circuito y, por ende, de la galga. Este cambio es así apreciable como una gráfica de valor ADC vs tiempo, donde el eje y en cuestión refleja un rango de cambio de 3 pasos ADC, o bien, 2.42 mV, concordando con lo obtenido en experimentación previa.

En cuanto a la comunicación de los datos inalámbricos, se obtiene una comunicación mediante *wifi* fluida con un promedio de obtención de 27.46 paquetes/segundo desde el microcontrolador hacia la computadora. Al analizar la data proveniente de los paquetes enviados, la cual corresponde a 7 bytes de información de los 49 bytes en total en cada paquete, se puede traducir este arreglo a un valor decimal congruente a lo estimado para el circuito puente de Wheatstone. Esto permite determinar que el microcontrolador ESP32-C3 Super Mini es capaz de procesar una lectura y enviar esta de manera inalámbrica de manera efectiva.

Ahora bien, los resultados del trabajo sientan las bases para el uso del sensor en un contexto biomédico, demostrando la viabilidad de utilización del sensor, galga extensiométrica, sobre una superficie plástica que asemeje una viga en voladizo. Al integrar este sensor junto a los componentes de un circuito puente de Wheatstone y un microcontrolador *mini* de ADC de 12 bits, se marca un gran avance hacia la utilización del sensor en un prototipo. Estos hallazgos demuestran los requerimientos mínimos de *hardware* para interpretar y caracterizar de manera confiable las deformaciones en una superficie. Más adelante, la caracterización en un *display* y la capacidad de enviar datos de manera alambicada, validan la portabilidad y autonomía del dispositivo alrededor de la utilización de la galga como sensor de actividad muscular.

Este trabajo propone un sistema para la medición del cambio de resistencia de una galga extensiométrica como cambios en voltaje mediante la implementación de una topología de circuito de puente de Wheatstone junto a un microcontrolador *mini* y la implementación de un filtro digital IIR. A la vez, se evaluó la implementación de una caracterización para apreciar estos cambios al mostrarlos con una pantalla OLED de protocolo I2C. Los datos de los cambios obtenidos también se enviaron de manera inalámbrica para proponer una comunicación alternativa, y, por ende, la fabricación de un dispositivo autónomo y portátil.

El sistema diseñado para medir la deformación de una superficie con el sensor se compuso por un cuarto de puente de Wheatstone junto a un microcontrolador con un módulo ADC de 12 bits y la implementación del filtro digital *leaky integrator*. Este sistema exhibió la capacidad de adquirir la deformación mecánica del sensor ( $\epsilon$ ) como pequeños cambios en resistencia ( $\Omega$ ) y traducirlos correctamente hacia un rango de variabilidad en voltaje (V) al aplicar el circuito puente de Wheatstone. Así entonces, adquiriendo este voltaje con la función ADC del microcontrolador y utilizando el filtro digital IIR con un factor  $\lambda$  alto, se obtuvo una deformación perceptible como una señal posprocesada.

Una experimentación inicial se llevó a cabo para determinar la aplicabilidad del uso del sensor para capturar la deformación de compresión y alargamiento de una viga de plástico flexible en voladizo al únicamente adherir el sensor con cinta. Los resultados indican cambios de deformación apreciables en la escala de mV. Esto permitió evaluar la posibilidad de escalar este sensor hacia superficies con propiedades de plasticidad como lo es la superficie muscular; para detectar así el nivel de actividad muscular en áreas corporales con una deformación de compresión o alargamiento.

Al aplicar un filtro digital mediante la implementación de una ecuación de diferencias en el *software* del microcontrolador, la señal proveniente del circuito puente de Wheatstone se aprecia en escala de los mV, lo cual permite distinguir cambios arriba de aproximadamente 1 mV. Esta diferenciación en la señal al deformar el material permite cuantificar los cambios en la galga con fiabilidad en tiempo real.

Así, se desarrolla un circuito impreso con un área total de  $5 \times 5$  cm con el propósito de integrar una caracterización de gráfica de datos en tiempo real con un *display* OLED de una manera compacta. Esto permitió la autonomía del prototipo al mostrar la deformación del sensor como una gráfica de voltaje o ADC vs tiempo en una pantalla fija al dispositivo. Además, tomando en cuenta una demanda de potencia baja en funcionamiento del sistema y un diseño óptimo para su portabilidad, la PCB resultante es de dimensiones mínimas. Esta contenía los siguientes elementos: Microcontrolador ESP32-C3 Super Mini, *headers* para las galgas, resistencias SMD, *display* OLED I2C, conectores JST-XH para la batería, *pads* para el elevador de voltaje y *pads* para la posibilidad de conexión del módulo UART.

El microcontrolador ESP32-C3 Super Mini además demuestra la capacidad de obtener una lectura analógica, procesar esta y enviarla mediante protocolo *wifi* de manera ininterrumpida, logrando hacer del dispositivo portátil al contar con un envío de datos de manera inalámbrica.

- El material propuesto durante el trabajo fue un plástico, al cual la galga únicamente estaba fijada con adhesivo. Una parte futura del trabajo recaerá en el acercamiento hacia una estructura de mayor semejanza a la del músculo, o bien, la implementación de la misma superficie muscular en la siguiente etapa.
- Desarrollar u obtener la galga extensiométrica con un patrón de rejilla diseñado específicamente para un área muscular de interés. Durante el trabajo, se utiliza una galga con una rejilla y de un tamaño pequeño. Al migrar hacia material muscular, se debe proponer una cuadrícula y tamaño que puedan maximizar los cambios de resistencia con la deformación de la galga en función del área a monitorear.
- Indagar sobre la mejora de los electrodos empleados en las galgas extensiométricas comerciales. Debido a la deformación constante, en numerosas ocasiones, los electrodos se desprendían del material base resistivo, dejando a la galga sin uso una vez sucedía esto.
- La implementación del filtro digital *leaky integrator* demuestra ser eficiente para la lectura en mV y así para la reducción de ruido. Se puede investigar e implementar otros filtros digitales para comparar el rendimiento y lograr minimizar el ruido. Aunque el filtro IIR implementado es simple, y fue propuesto a manera de usar poca capacidad computacional del microcontrolador, un filtro FIR es una opción prometedora para la mejora del filtrado. Este último tipo de filtros no es un equivalente digital a un filtro analógico, mas una mejora de estos, ya que sus propiedades son difíciles de replicar con viabilidad en un circuito analógico.
- La portabilidad del dispositivo es importante, tanto el uso de materiales no tan rígidos y una PCB de material flexible se pueden explorar en busca de mayor portabilidad del dispositivo. El peso del prototipo también se debe considerar respecto a los materiales y módulos empleados.
- La caracterización es un paso para seguir desarrollando. Durante el trabajo, se muestra una gráfica de ADC vs tiempo para que el usuario fácilmente distinga el efecto

de la deformación del sensor. Uno de los pasos a seguir es diseñar y codificar esta caracterización para una mayor interactividad para el usuario.

- 
- [1] J.-H. Sul et al., «Electromyography Signal Acquisition, Filtering, and Data Analysis for Exoskeleton Development,» *Sensors*, vol. 25, n.º 13, 2025, ISSN: 1424-8220. DOI: 10.3390/s25134004. dirección: <https://www.mdpi.com/1424-8220/25/13/4004>.
  - [2] Q. Wang et al., «Monitoreo intradérmico de lactato basado en un parche sensor de microneedles para mayor precisión in vivo,» *ACS Sensors*, vol. 9, n.º 6, págs. 3115-3125, 2024. DOI: 10.1021/acssensors.4c00337.
  - [3] S. Perilli et al., «Development of a Wearable Electromyographic Sensor with Aerosol Jet Printing Technology,» *Bioengineering*, vol. 11, n.º 12, 2024, ISSN: 2306-5354. DOI: 10.3390/bioengineering11121283. dirección: <https://www.mdpi.com/2306-5354/11/12/1283>.
  - [4] F. Landi et al., «Muscle loss: The new malnutrition challenge in clinical practice,» *Clinical Nutrition*, vol. 38, n.º 5, págs. 2113-2120, 2019, ISSN: 0261-5614. DOI: 10.1016/j.clnu.2018.11.021.
  - [5] J. Stelletta, R. Dumas e Y. Lafon, «Modeling of the thigh,» en *Biomechanics of Living Organs*, Elsevier, 2017, págs. 497-521. DOI: 10.1016/B978-0-12-804009-6.00023-7.
  - [6] R. J. McLaughlin, «Systematic design of cantilever beams for muscle research,» *Journal of Applied Physiology: Respiratory, Environmental and Exercise Physiology*, vol. 42, n.º 5, págs. 786-794, 1977. DOI: 10.1152/jappl.1977.42.5.786.
  - [7] N. Afanador-García, G. Guerrero-Gómez y R. Gallardo-Amaya, «Structural and physical evaluation of a reinforced beam using strain gauges,» *Journal of Physics: Conference Series*, vol. 2153, n.º 1, pág. 012003, 2022. DOI: 10.1088/1742-6596/2153/1/012003.
  - [8] V. M. Zatsiorsky y B. I. Prilutsky, *Biomechanics of skeletal muscles*. Human Kinetics, 2012, págs. 1-20.
  - [9] X. Wang, H. Yu, S. Kold, O. Rahbek y S. Bai, «Wearable sensors for activity monitoring and motion control: A review,» *Biomimetic Intelligence and Robotics*, vol. 3, n.º 1, pág. 100089, 2023, ISSN: 2667-3797. DOI: 10.1016/j.birob.2023.100089. dirección: <https://www.sciencedirect.com/science/article/pii/S2667379723000037>.

- [10] D. Esposito et al., «A Piezoresistive Sensor to Measure Muscle Contraction and Mechanomyography,» *Sensors*, vol. 18, n.º 8, pág. 2553, 2018. DOI: 10.3390/s18082553.
- [11] A. Koyun, E. Ahlatcolu, Y. Koca y S. Kara, «Biosensors and their principles,» *A roadmap of biomedical engineers and milestones*, vol. 5, págs. 115-117, 2012.
- [12] D. C. Z. Ríos, A. J. M. Miramar, Y. M. Paz e I. C. R. Padilla, «Lactato: un marcador biológico de la actividad física en atletas colombianos de levantamiento de pesas,» *Revista Brasileira de Medicina do Esporte*, vol. 27, n.º 1, págs. 65-69, 2021. DOI: 10.1590/1517-8692202127012019\_0047.
- [13] P. C. Pandey, G. Pandey y R. J. Narayan, «Minimally Invasive Platforms in Biosensing,» *Frontiers in Bioengineering and Biotechnology*, vol. 8, ago. de 2020. DOI: 10.3389/fbioe.2020.00894.
- [14] L. Hadady, T. Robinson, E. Bruno, M. Richardson y S. Beniczky, «Users' perspectives and preferences on using wearables in epilepsy: A critical review,» *Epilepsia*, vol. 66, 2025, ISSN: 0013-9580. DOI: 10.1111/epi.18280. dirección: <https://onlinelibrary.wiley.com/doi/10.1111/epi.18280>.
- [15] M. C. Gash y colleagues, «Physiology, Muscle Contraction,» en *Treasure Island (FL): StatPearls Publishing*, 2023, Disponible en: National Library of Medicine. Consultado el 17 de marzo de 2025.
- [16] Callibri. «Especificación de Callibri.» Consultado el 17 de marzo de 2025. dirección: <https://callibri.com/specification/>.
- [17] T. Stuart, J. Hanna y P. Gutruf, «Wearable devices for continuous monitoring of bio-signals: Challenges and opportunities,» *APL Bioengineering*, vol. 6, n.º 2, pág. 021 502, 2022. DOI: 10.1063/5.0086935.
- [18] B. T. Corona, J. C. Wenke y C. L. Ward, «Pathophysiology of volumetric muscle loss injury,» *Cells, Tissues, Organs*, vol. 202, n.º 3-4, págs. 180-188, 2016. DOI: 10.1159/000443925.
- [19] M. Francaux y L. Deldicque, «Exercise and the control of muscle mass in human,» *Pflügers Archiv - European Journal of Physiology*, vol. 471, págs. 397-411, 2018. DOI: 10.1007/s00424-018-2217-x.
- [20] F. De Engenharia. «Surface electromyographic amplitude normalization methods: A review.» Consultado el 4 de junio de 2025. dirección: <https://repositorio-aberto.up.pt/handle/10216/644300>.
- [21] O. M. Blake y J. M. Wakeling, «Estimating changes in metabolic power from EMG,» *SpringerPlus*, vol. 2, pág. 229, 2013. DOI: 10.1186/2193-1801-2-229.
- [22] C. Fang, B. He, Y. Wang, J. Cao y S. Gao, «EMG-Centered Multisensory Based Technologies for Pattern Recognition in Rehabilitation: State of the Art and Challenges,» *Biosensors*, vol. 10, n.º 8, pág. 85, 2020. DOI: 10.3390/bios10080085.
- [23] L. Marcucci y L. Truskinovsky, «Muscle contraction: A mechanical perspective,» *European Physical Journal E*, vol. 32, págs. 411-418, 2010. DOI: 10.1140/epje/i2010-10641-0.
- [24] C. Molnar y J. Gair, *Concepts of Biology – 1st Canadian Edition*. BCcampus, 2015, Retrieved from the Open Text BCcampus repository, ISBN: 978-1-989623-99-2. dirección: <https://opentextbc.ca/biology/>.

- [25] J. Squire, *The Structural Basis of Muscular Contraction*, 1.<sup>a</sup> ed. New York, NY: Springer New York, NY, 2012, pág. 716, Edición original publicada por Plenum Press en 1981, ISBN: 978-1-4613-3183-4. DOI: 10.1007/978-1-4613-3183-4.
- [26] R. L. Lieber, T. J. Roberts, S. S. Blemker, S. S. M. Lee y W. Herzog, «Skeletal muscle mechanics, energetics and plasticity,» *Journal of NeuroEngineering and Rehabilitation*, vol. 14, n.º 1, pág. 108, 2017. DOI: 10.1186/s12984-017-0318-y. dirección: <https://doi.org/10.1186/s12984-017-0318-y>.
- [27] National Instruments. «Medir tensión con galgas extensiométricas.» Consultado el 16 de agosto de 2025. dirección: <https://www.ni.com/es/shop/data-acquisition/sensor-fundamentals/measuring-strain-with-strain-gages.html?srsltid=AfmB0oqIc0WxKKWjBP6aQoHYBGgqFTXEoJN7QyiGgwbrpIsyQ0oPPQhS>.
- [28] J. Park, I. You, S. Shin y U. Jeong, «Material approaches to stretchable strain sensors,» *ChemPhysChem: A European Journal of Chemical Physics and Physical Chemistry*, vol. 16, n.º 6, págs. 1155-1163, 2015. DOI: 10.1002/cphc.201402810.
- [29] MinebeaMitsumi. «What is a strain gage?» Consultado el 15 de agosto de 2025. dirección: [https://product.minebeamitsumi.com/en/technology/sensing\\_devices/s-gages\\_sg\\_about.html](https://product.minebeamitsumi.com/en/technology/sensing_devices/s-gages_sg_about.html).
- [30] National Instruments. «Strain Gauge Measurement – A Tutorial.» Application Note 078. Consultado el 17 de agosto de 2025. dirección: [http://elektron.pol.lublin.pl/elekp/ap\\_notes/ni\\_an078\\_strain\\_gauge\\_meas.pdf](http://elektron.pol.lublin.pl/elekp/ap_notes/ni_an078_strain_gauge_meas.pdf).
- [31] ANYLOAD Weigh & Measure, *How does a load cell work?* Consultado el 17 de agosto de 2025. dirección: <https://www.anyload.com/how-does-a-load-cell-work>.
- [32] J. Fraden, «Force and Strain,» en *Handbook of Modern Sensors*, Springer International Publishing, 2016, págs. 413-428.
- [33] S. Yang y N. Lu, «Gauge Factor and Stretchability of Silicon-on-Polymer Strain Gauges,» *Sensors*, vol. 13, n.º 7, págs. 8577-8594, 2013, ISSN: 1424-8220. DOI: 10.3390/s130708577. dirección: <https://www.mdpi.com/1424-8220/13/7/8577>.
- [34] Y. Zhao, Y. Liu, Y. Li y Q. Hao, «Development and Application of Resistance Strain Force Sensors,» *Sensors*, vol. 20, n.º 20, pág. 5826, 2020. DOI: 10.3390/s20205826. dirección: <https://doi.org/10.3390/s20205826>.
- [35] Bestech Australia. «Understanding a Wheatstone Bridge strain gauge circuit.» Blog post. dirección: <https://www.bestech.com.au/blogs/understanding-a-wheatstone-bridge-strain-gauge-circuit>.
- [36] O. O. Oluwole, A. T. Olanipekun y O. O. Ajide, «Design, Construction and Testing of a Strain Gauge Instrument,» *International Journal of Scientific & Engineering Research*, vol. 6, págs. 1825-1829, 2015.
- [37] Z. Czaja, «An implementation of a compact smart resistive sensor based on a microcontroller with an internal ADC,» *Metrology and Measurement Systems*, vol. 23, n.º 2, págs. 225-238, 2016. DOI: 10.1515/mms-2016-0020.
- [38] J. Wang et al., «Challenges and progresses of lithium-metal batteries,» *Chemical Engineering Journal*, vol. 420, págs. 129-739, 2021. DOI: 10.1016/j.cej.2021.129739.
- [39] *ESP32 Technical Reference Manual*, Consultado el 17 de agosto de 2025, Espressif Systems, 2025. dirección: [https://documentation.espressif.com/esp32\\_technical\\_reference\\_manual\\_en.pdf](https://documentation.espressif.com/esp32_technical_reference_manual_en.pdf).

- [40] S. Jamuna, P. Dinesha y K. P. Shashikala, «A brief review on types and design methods of ADC,» *Journal of Engineering Research and Applications*, vol. 8, n.º 6, págs. 85-91, 2018. DOI: 10.9790/9622-0806028591.
- [41] Analog-to-digital converters basics. «Analog-to-digital converters basics.» Consultado el 17 de agosto de 2025. dirección: <https://www.arrow.com/en/research-and-events/articles/engineering-resource-basics-of-analog-to-digital-converters>.
- [42] AlazarTech. «Variable frequency ADC clock technology - AlazarTech PCI digitizers. PC Oscilloscope PC Scope card and systems.» Consultado el 17 de agosto de 2025. dirección: <https://www.alazartech.com/en/technology/variable-frequency-adc-clock>.
- [43] Electrical Technology. «Analog to Digital Converter (ADC) - block Diagram, factors & applications.» Consultado el 17 de agosto de 2025. dirección: <https://www.electricaltechnology.org/2019/02/analog-to-digital-converter-adc.html>.
- [44] P. Prandoni y M. Vetterli, *Signal Processing for Communications* (Communication and Information Sciences). Lausanne, Switzerland: EPFL Press, 2008, ISBN: 978-2-940222-20-9.
- [45] Analog Devices. «Understanding and Eliminating 1/f Noise.» Consultado el 8 de noviembre de 2025. dirección: <https://www.analog.com/en/resources/analog-dialogue/articles/understanding-and-eliminating-1-f-noise.html>.
- [46] T. Janssen, N. BniLam, M. Aernouts, R. Berkvens y M. Weyn, «LoRa 2.4 GHz communication link and range,» *Sensors*, vol. 20, n.º 16, pág. 4366, 2020. DOI: 10.3390/s20164366.
- [47] Espressif Systems, *ESP32-C3-MINI-1 Datasheet*, Consultado el 17 de agosto de 2025, 2025. dirección: [https://www.espressif.com/sites/default/files/documentation/esp32-c3-mini-1\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-c3-mini-1_datasheet_en.pdf).

En estos anexos se incluye información adicional para entender la metodología del trabajo. Se proporcionan los códigos empleados para el microcontrolador ESP32-C3 Super Mini; para la obtención, análisis y procesamiento de las señales provenientes del puente de Wheatstone. Así como los códigos empleados para Matlab, para el análisis de archivos *.csv* y la captura de información enviada mediante comunicación serial. Además, se incluyen los archivos de Python utilizados para la extracción de data en paquetes enviados mediante *wifi* junto a su caracterización.

### 13.1. Código en Matlab para el análisis de señal en archivo *.csv*

```
L = readlines("nombre_archivo.csv");
offset = 0.334; % (en caso de haber utilizado)
max_v = offset+0.025;
min_v = offset-0.025;

L = L(strlength(L) > 0);

tok = regexp(L, '[-+]?\d*\.\d+(?:E[-+]?\d+)?', 'match');

data = cellfun(@(t) [str2double(t{1}), str2double(t{2})], tok, ...
              'UniformOutput', false);
data = vertcat(data{:});

time = data(:,1);
voltage = data(:,2);

plot(time, voltage, 'o-');
grid on;
xlabel('Tiempo (s)');
ylabel('Voltaje (V)');
title('Voltaje vs Tiempo (Data de Osciloscopio)');
```

```
ylim([min_v max_v]);
```

## 13.2. Código para envío de lectura ADC utilizando un sensor mediante comunicación serial

```
#include <HardwareSerial.h>

HardwareSerial MySerial(1);

const int adcPin = 0;
const int txPin = 20;
const int rxPin = 21;

float x_n = 0.0;
float y_n = 0.0;
float y_n_1 = 0.0;
float lambda = 0; // lambda = 0, lambda = 0.8, lambda = 0.99

int adcMin = 0;
int adcMax = 0;
bool calibrated = false;

unsigned long calibStart = 0;
const unsigned long calibDuration = 5000;

unsigned long sampleCount = 0;
unsigned long adcSum = 0;

void setup() {
  MySerial.begin(115200, SERIAL_8N1, rxPin, txPin);

  analogReadResolution(12);
  analogSetAttenuation(ADC_11db);

  calibStart = millis();
}

void loop() {
  x_n = analogRead(adcPin);

  if (sampleCount == 0){
    y_n = x_n;
    y_n_1 = y_n;
  }

  if (!calibrated) {
    adcSum += x_n;
    sampleCount++;

    if (millis() - calibStart >= calibDuration) {
      calibrated = true;
      int adcAvg = adcSum / sampleCount;

      adcMin = adcAvg - 1;
      adcMax = adcAvg + 1;
    }
  }
}
```

```

}

delay(10);
return;
}

if (sampleCount != 0){
    y_n = (1 - lambda) * x_n + lambda * y_n_1;
    y_n_1 = y_n;
}

MySerial.println(y_n);

delay(10);
}

```

### 13.3. Código para obtención de lectura ADC mediante comunicación serial en Matlab

```

%% IMPORTACION DE DATOS

% Parametros
port = "COM13";      % Puerto COM
baud = 115200;      % Baud rate
waitTime = 5;      % Tiempo a esperar antes de grabar
duration = 10;      % Duracion para obtencion de data (s)

% Crear objeto
s = serialport(port, baud);

% Flush de datos viejos
flush(s);

% Espera
disp("Esperar por " + waitTime + " segundos antes de grabar...");
pause(waitTime);
disp("Inicio de captura de datos...");

% Inicilizar almacenamiento
t0 = tic;
ADC_reading_array_name = [];
time_array_name = [];

% Captura de senal de voltaje
while toc(t0) < duration
    if s.NumBytesAvailable > 0
        str = readline(s); % Leer data en puerto COM
        val = str2double(str);
        if ~isnan(val)
            ADC_reading_array_name(end+1) = val;
            time_array_name(end+1) = toc(t0);
        end
    end
end
end

```

```

%% GRAFICO DE RESULTADOS

% Grafica ADC vs TIEMPO
figure;
plot(time_array_name, ADC_reading_array_name, '-','Color', [0.0 0.6 0.0]);
xlabel('Tiempo(s)');
ylabel('Valor ADC (Unidad Digital)');
title('Valor ADC vs Tiempo');
grid on;
legend('sin manipular','manipulada');
xlim([0 10]); % rango de eje x

% Grafica VOLTAJE vs TIEMPO
figure;
plot(time_array_name, (ADC_reading_array_name/4095)*3.3, '-','Color', [0.0 0.6 0.0]);
xlabel('Tiempo(s)');
ylabel('Voltaje(V)');
title('Voltaje vs Tiempo');
grid on;
legend('sin manipular','manipulada');
xlim([0 10]); % rango de eje x

% Guardar los resultados en archivo mat
save RUNS.mat ADC_reading_array_name time_array_name

% Clean up
clear s

```

### 13.4. Código para envío de lectura ADC utilizando dos sensores mediante comunicación serial

```

#include <HardwareSerial.h>

HardwareSerial MySerial(1);

const int adcPin0 = 0;
const int adcPin1 = 1;
const int txPin = 20;
const int rxPin = 21;

float x_n = 0.0;
float y_n = 0.0;
float y_n_1 = 0.0;
float lambda = 0.99;

float v_n = 0.0;
float w_n = 0.0;
float w_n_1 = 0.0;

float z_n = 0.0;
float a_n = 0.0;
float a_n_1 = 0.0;

int adcMin = 0;

```

```

int adcMax = 0;
bool calibrated = false;

unsigned long calibStart = 0;
const unsigned long calibDuration = 5000;

unsigned long sampleCount = 0;
unsigned long adcSum = 0;

void setup() {
  MySerial.begin(115200, SERIAL_8N1, rxPin, txPin);

  analogReadResolution(12);
  analogSetAttenuation(ADC_11db);

  calibStart = millis();
}

void loop() {
  x_n = analogRead(adcPin0);
  v_n = analogRead(adcPin1);

  if (sampleCount == 0){
    y_n = x_n;
    y_n_1 = y_n;

    w_n = v_n;
    w_n_1 = w_n;

    z_n = v_n-x_n;
    a_n = z_n;
  }

  if (!calibrated) {

    adcSum += v_n-x_n;
    sampleCount++;

    if (millis() - calibStart >= calibDuration) {
      calibrated = true;
      int adcAvg = adcSum / sampleCount;

      adcMin = adcAvg - 1;
      adcMax = adcAvg + 1;
    }

    delay(10);
    return;
  }

  if (sampleCount != 0){
    z_n = v_n-x_n;
    a_n = (1 - lambda) * z_n + lambda * a_n_1;
    a_n_1 = a_n;
  }

  MySerial.println(a_n);

  delay(10);
}

```

```
}
```

## 13.5. Código para caracterizar la señal como una gráfica de ADC vs Tiempo

```
#include <Arduino.h>
#include <U8g2lib.h>
#include <Wire.h>

U8G2_SH1106_128X64_NONAME_F_HW_I2C oled(U8G2_R0, /* reset=*/ U8X8_PIN_NONE);

const int adcPin = 0;
const int calibButton = 20;

int adcMin = 0;
int adcMax = 4095;
float x_n = 0.0;
float y_n = 0.0;
float y_n_1 = 0.0;
float lambda = 0.99;

unsigned long calibStart = 0;
const unsigned long calibDuration = 5000;
const unsigned long autoRecalInterval = 10000;
unsigned long lastAutoRecal = 0;
long adcSum = 0;
int sampleCount = 0;
bool calibrated = false;

const unsigned long targetHz = 200;
const unsigned long sampleInterval = 1000000UL / targetHz;

const unsigned long displayWindowMs = 200;
#define MAX_SAMPLES 256
int SAMPLES = (displayWindowMs * targetHz) / 1000;
int adcBuffer[MAX_SAMPLES];
int writeIndex = 0; // circular buffer index
unsigned long lastDisplay = 0;
const unsigned long displayInterval = 50; // ~20 FPS

unsigned long lastSampleMicros = 0;

void drawAxes();
void startCalibration();
void autoRecalibrate();
void drawWaveform();

void drawAxes() {
  oled.drawLine(10, 0, 10, 63);
  oled.drawLine(10, 63, 127, 63);
  oled.setFont(u8g2_font_5x8_tf);
  oled.setCursor(0, 8);
  oled.print(adcMax);
  oled.setCursor(0, 62);
```

```

    oled.print(adcMin);
}

void setup() {
    analogReadResolution(12);
    analogSetAttenuation(ADC_11db);
    pinMode(calibButton, INPUT_PULLDOWN);

    oled.begin();
    oled.clearBuffer();
    oled.setFont(u8g2_font_6x10_tf);
    oled.drawStr(5, 32, "Calibrando...");
    oled.sendBuffer();

    startCalibration();
}

void startCalibration() {
    calibrated = false;
    adcSum = 0;
    sampleCount = 0;
    calibStart = millis();

    oled.begin();
    oled.clearBuffer();
    oled.setFont(u8g2_font_6x10_tf);
    oled.drawStr(5, 32, "Calibrando...");
    oled.sendBuffer();

    Serial.println("Calibracion reiniciada!");
}

void autoRecalibrate() {
    if (millis() - lastAutoRecal >= autoRecalInterval) {
        if (sampleCount > 0) {
            int adcAvg = adcSum / sampleCount;
            adcMin = adcAvg - 1.5;
            adcMax = adcAvg + 1.5;
        }
        adcSum = 0;
        sampleCount = 0;
        lastAutoRecal = millis();
    }
}

void drawWaveform() {
    oled.clearBuffer();
    drawAxes();

    int oldestIndex = (writeIndex + 1) % SAMPLES;

    int prevY = map(adcBuffer[oldestIndex], adcMin, adcMax, 62, 0);
    prevY = constrain(prevY, 0, 62);

    for (int i = 1; i < SAMPLES; i++) {

```

```

    int index = (oldestIndex + i) % SAMPLES;
    int x1 = map(i - 1, 0, SAMPLES - 1, 11, 127);
    int x2 = map(i, 0, SAMPLES - 1, 11, 127);
    int y = map(adcBuffer[index], adcMin, adcMax, 62, 0);
    y = constrain(y, 0, 62);
    oled.drawLine(x1, prevY, x2, y);
    prevY = y;
}

oled.sendBuffer();
}

void loop() {

    if (digitalRead(calibButton) == HIGH) {
        startCalibration();
        delay(500);
    }

    if (micros() - lastSampleMicros >= sampleInterval) {
        lastSampleMicros = micros();
        x_n = analogRead(adcPin);

        if (!calibrated) {
            adcSum += x_n;
            sampleCount++;
            if (millis() - calibStart >= calibDuration) {
                calibrated = true;
                int adcAvg = adcSum / sampleCount;
                adcMin = adcAvg - 1.5;
                adcMax = adcAvg + 1.5;
                adcSum = 0;
                sampleCount = 0;
                lastAutoRecal = millis();
                Serial.println("Calibracion completa");
            }
            return;
        }

        y_n = (1 - lambda) * x_n + lambda * y_n_1;
        y_n_1 = y_n;

        adcBuffer[writeIndex] = y_n;
        writeIndex = (writeIndex + 1) % SAMPLES;

        adcSum += x_n;
        sampleCount++;
    }

    if (calibrated)
        autoRecalibrate();

    if (calibrated && millis() - lastDisplay >= displayInterval) {
        lastDisplay = millis();
        drawWaveform();
    }
}

```

## 13.6. Código para enviar la señal como data a través de *wifi*

```
\begin{lstlisting}
#include <WiFi.h>
#include <WiFiUdp.h>
#include <U8g2lib.h>
#include <Wire.h>

const char* ssid = "moto_TAS";
const char* password = "ojito123";

const char* hostIP = "10.192.234.112"; // IP de computadora
const int udpPort = 4210;

WiFiUDP udp;

U8G2_SH1106_128X64_NONAME_F_HW_I2C oled(U8G2_R0, U8X8_PIN_NONE);

const int adcPin = 0; // GPIO0 para ADC
const int txPin = 20; // TX
const int rxPin = 21; // RX
const int calibButton = 20; // GPIO20 - Input digital

const float refVoltage = 3.3; // referencia de voltaje - ESP32
const int adc_bits = 12;
float x_n = 0.0;
float y_n = 0.0;
float y_n_1 = 0.0;
float lambda = 0.99;
float voltage = 0.0;

const float sampleRate = 200.0; // Hz
const unsigned long sampleInterval = 1000000UL / sampleRate;
unsigned long lastSampleMicros = 0;

unsigned long sampleCount = 0;

void setup() {

  pinMode(calibButton, INPUT_PULLDOWN);

  analogReadResolution(adc_bits);
  analogSetAttenuation(ADC_11db); // para rangos de 0 a 3.3 V

  oled.begin();
  oled.clearBuffer();
  oled.setFont(u8g2_font_6x10_tf);
  oled.drawStr(0, 20, "Conectando a WiFi...");
  oled.sendBuffer();

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }

  oled.clearBuffer();
  oled.setFont(u8g2_font_6x10_tf);
  oled.drawStr(0, 15, "Conectado a WiFi!");
}
\end{lstlisting}
```

```

delay(1000); // Mostrar mensaje por 1s
oled.drawStr(0, 35, "IP asignada:");
oled.setCursor(0, 55);
oled.print(WiFi.localIP().toString());
oled.sendBuffer();
delay(5000); // Mostrar IP por 5s

oled.clearBuffer();
oled.drawStr(0, 30, "Empezando a enviar...");
oled.sendBuffer();
delay(1000); // Mostrar mensaje por 1s
}

void restartOLED() {

// Reiniciar pantalla OLED - I2C
oled.begin();
oled.clearBuffer();
oled.setFont(u8g2_font_6x10_tf);

}

void loop() {

if (sampleCount == 0){
  x_n = analogRead(adcPin);
  y_n = x_n;
  y_n_1 = y_n;
  sampleCount++;
}

if (digitalRead(calibButton) == HIGH) {
  restartOLED();
  delay(500); // antirrebote
  oled.drawStr(0, 30, "Pausa de 5s...");
  oled.sendBuffer();
  delay(5000);
}

unsigned long nowMicros = micros();
if (nowMicros - lastSampleMicros >= sampleInterval) {
  lastSampleMicros = nowMicros;

  x_n = analogRead(adcPin);

  y_n = (1 - lambda) * x_n + lambda * y_n_1;
  y_n_1 = y_n;

  voltage = (y_n / float((1 << adc_bits) - 1)) * refVoltage;

  String msg = String(voltage, 5); // 5 decimales

  udp.beginPacket(hostIP, udpPort);
  udp.write((uint8_t*)msg.c_str(), msg.length());
  udp.endPacket();
}

oled.clearBuffer();
oled.setFont(u8g2_font_6x10_tf);

```

```

oled.drawStr(0, 10, "Voltaje ADC:");
oled.setCursor(0, 20);
oled.print(voltage);
oled.drawStr(0, 50, "IP asignada:");
oled.setCursor(0, 60);
oled.print(WiFi.localIP().toString());
oled.sendBuffer();
}

```

## 13.7. Código para la extracción de data de paquetes entrantes en la interfaz *wifi*

```

import socket
import threading
from collections import deque
from PyQt5 import QtWidgets, QtCore
import pyqtgraph as pg
import sys

UDP_IP = "0.0.0.0"
UDP_PORT = 4210

MAX_SAMPLES = 1000

class UDPReceiver(threading.Thread):
    def __init__(self, data_buffer, limits):
        super().__init__(daemon=True)
        self.data_buffer = data_buffer
        self.limits = limits
        self.sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        self.sock.bind((UDP_IP, UDP_PORT))
        print(f"Listening for UDP packets on port {UDP_PORT}...")

    def run(self):
        while True:
            try:
                data, addr = self.sock.recvfrom(1024)
                message = data.decode().strip()
                parts = message.split()

                if len(parts) == 3:
                    adc_min, adc_max, voltage = map(float, parts)
                    self.limits['min'] = adc_min
                    self.limits['max'] = adc_max
                    self.data_buffer.append(voltage)
                else:
                    print(f"Invalid data format: {message}")

            except Exception as e:
                print(f"UDP error: {e}")

class PlotWindow(QtWidgets.QMainWindow):
    def __init__(self):
        super().__init__()

```

```

self.setWindowTitle("ESP32_UDP_Real-Time_Plot")
self.setGeometry(200, 200, 800, 400)

self.data_buffer = deque(maxlen=MAX_SAMPLES)
self.limits = {'min': 0, 'max': 3.3} # initial voltage range

self.receiver = UDPReceiver(self.data_buffer, self.limits)
self.receiver.start()

self.graphWidget = pg.PlotWidget()
self.setCentralWidget(self.graphWidget)
self.graphWidget.setBackground('k')
self.graphWidget.setTitle("Voltaje vs Tiempo", color='w', size="14pt")
self.graphWidget.setLabel('left', 'Voltaje (V)')
self.graphWidget.setLabel('bottom', 'Muestras')
self.graphWidget.showGrid(x=True, y=True)

self.curve = self.graphWidget.plot(pen=pg.mkPen('y', width=2))

self.timer = QtCore.QTimer()
self.timer.timeout.connect(self.update_plot)
self.timer.start(20) # 50 FPS refresh

def update_plot(self):
    if len(self.data_buffer) > 0:
        self.curve.setData(list(self.data_buffer))

        y_min = self.limits['min']
        y_max = self.limits['max']

        padding = (y_max - y_min) * 0.1 if y_max != y_min else 0.1
        self.graphWidget.setYRange(y_min - padding, y_max + padding)

def main():
    app = QtWidgets.QApplication(sys.argv)
    win = PlotWindow()
    win.show()
    sys.exit(app.exec_())

```

