



Universidad de Jaén

Escuela Politécnica Superior de Jaén

DESARROLLO DE UNA INTERFAZ DATAGLOVE

Autor: Carlos Artero Aragón

Grado: Ingeniería Electrónica Industrial

Fecha: 04/06/2024

Licencia CC



CREA



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don LUIS MIGUEL NIETO NIETO , tutor del Proyecto Fin de Carrera titulado: DESARROLLO DE UNA INTERFAZ DATAGLOVE, que presenta CARLOS ARTERO ARAGÓN, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, JUNIO de 2024

El alumno:

CARLOS ARTERO ARAGÓN

Los tutores:

LUIS MIGUEL NIETO NIETO

JOSÉ VICENTE MUÑOZ DÍEZ

AGRADECIMIENTOS

Gracias a mi familia, mis amigos y mi pareja por el apoyo y ánimo que me han brindado en todo momento durante el transcurso de mi etapa académica.

Gracias a la Universidad de Jaén y a los profesores que he tenido a lo largo de mi etapa universitaria, por la formación y la ayuda que me han procurado durante estos años.

Gracias a mis tutores Luis Miguel Nieto Nieto y José Vicente Muñoz Díez por su dedicación, ayuda y apoyo a lo largo del desarrollo del presente trabajo.

Índice de contenido

RESUMEN	10
INTRODUCCIÓN	12
Justificación.....	13
Objetivos	13
Organización de la memoria.....	14
ESTADO DEL ARTE	18
Introducción.....	18
Concepto de guante de datos	18
Sensores	19
Sensor Flex	19
Acelerómetro	21
Giroscopio	22
IMU.....	23
Trabajos previos.....	24
DISEÑO DE HARDWARE Y SOFTWARE	26
Organización de diseño.....	26
Elección del microcontrolador	27
Esquema hardware de la aplicación.....	28
Acondicionamiento de señal del sensor Flex	30
Divisor de tensión.....	30
Primer buffer.....	31
Buffer de salida.....	31
Sumador-restador.....	31
Elección de resistencia de pull-up (R_a)	32
Diseño de la etapa amplificadora	33
Caracterización del sensor Flex	37
Caracterización.....	38
Análisis de Errores de Medida.....	42
Procesado de datos del sensor Flex	43
Procesado de datos del sensor MPU6050	44
Desarrollo Software.....	46
Lectura de los valores de los sensores	46
Interfaz de LabVIEW	47
Descripción de la interfaz de usuario.....	48

CONCLUSIONES Y LÍNEAS FUTURAS	50
Conclusiones.....	50
Líneas futuras.....	50
Bibliografía	52
ANEXO.....	54
Operativa de usuario	54
Requisitos del hardware	54
Requisitos de software	55

Índice de figuras

Figura 1. Nintendo Power Glove	19
Figura 2. Sensor Flex	20
Figura 3. Guante de datos con sensores Flex.....	21
Figura 4. Principio de funcionamiento de un acelerómetro.....	22
Figura 5. Funcionamiento interno de un sensor MEMS	22
Figura 6. Ejes de trabajo del sensor IMU MPU6050	23
Figura 7. Diagrama de bloques del diseño	27
Figura 8. Placa Arduino UNO	28
Figura 9. Esquema hardware completo.....	30
Figura 10. Conversor ADC del microcontrolador ATmega328P. Fuente: Microchip.....	31
Figura 11. Curva de valores de Ra frente a la variación de V_{S0}	32
Figura 12. Gráfica de resistencia y voltaje en función de la flexión del sensor. 34	
Figura 13. Montaje para toma de datos de cada valor de flexión	38
Figura 14. Histograma asociado al ángulo de flexión de 0°	39
Figura 15. Histograma asociado al ángulo de flexión de 30°	40
Figura 16. Histograma asociado al ángulo de flexión de 60°	40
Figura 17. Histograma asociado al ángulo de flexión de 90°	41
Figura 18. Sensor MPU6050	44
Figura 19. Gráfica de inclinación sin filtro frente a con filtro complementario... 46	
Figura 20. Interfaz de usuario (Panel frontal)	48
Figura 21. Interfaz de usuario en funcionamiento	49
Figura 22. Flexión por la cara correcta del sensor	55
Figura 23. Gráfica de ejemplo 1	57
Figura 24. Gráfica de ejemplo 2	57
Figura 25. Gráfica de ejemplo 3	58
Figura 26. Gráfica de valores de inclinación de la mano	58
Figura 27. Primera parte del diagrama de bloques	59
Figura 28. Segunda parte del diagrama de bloques.....	60

RESUMEN

El presente trabajo de fin de grado se enfoca en el estudio de la tecnología para fabricar un guante de datos capaz de detectar los movimientos característicos de una mano. Para lograr este objetivo, se hace uso de un sensor inercial MPU6050 y un sensor tipo Flex, los cuales se integran con una placa Arduino UNO.

El proceso se lleva a cabo utilizando la plataforma Arduino que, en conjunto con la interfaz de LabVIEW, permite el procesamiento y la visualización de los datos recopilados por los sensores. Este enfoque proporciona una herramienta eficaz para capturar y analizar los movimientos de la mano en tiempo real.

Como conclusión, se identifican varias aplicaciones potenciales para el guante de datos desarrollado, destacando su utilidad en campos como la rehabilitación física, la realidad virtual o la interfaz hombre-máquina. Estas aplicaciones muestran el valor y la versatilidad de esta tecnología en diversos contextos.

INTRODUCCIÓN

En los últimos años, se ha experimentado un notable avance en los dispositivos diseñados para la detección y seguimiento de los movimientos humanos. Al ser la mano la parte más móvil del cuerpo humano, entre estos dispositivos, los guantes de datos han recibido una considerable atención debido a su capacidad para capturar y transmitir información detallada sobre los movimientos, la posición, la fuerza y la presión aplicada por las manos y los dedos del usuario.

Para el desarrollo de estos guantes de datos, se han empleado una variedad de sensores, entre ellos acelerómetros, giroscopios, magnetómetros, sensores de flexión y sensores de presión. Esta combinación de sensores proporciona una amplia gama de aplicaciones en diversos campos. En el ámbito del entretenimiento y los videojuegos, por ejemplo, se utilizan para controlar personajes virtuales y manipular objetos en entornos de realidad virtual y aumentada. En medicina, estos guantes son utilizados para la manipulación y el control de robots y maquinaria, así como en investigaciones científicas para estudiar el movimiento humano y la biomecánica.

Aunque todo este tipo de guantes ofrecen variedad de posibilidades, también presentan desafíos técnicos. La precisión y la confiabilidad de los sensores son aspectos críticos, especialmente en aplicaciones médicas y de rehabilitación donde se requiere una captura precisa de movimientos. La integración de múltiples sensores y la calibración adecuada son fundamentales para garantizar la calidad de los datos capturados.

Un ejemplo de desarrollo de un guante de datos es el estudio que ha llevado a cabo la Universidad de Ciencias Islámicas de Malasia [1], que destaca la importancia y el potencial de estas tecnologías en el campo de la salud. En un contexto en el que el ictus ocupa el tercer lugar de causa de muerte en Malasia, se trata de un estudio realizado para desarrollar una herramienta que facilite la rehabilitación remota de pacientes que han sufrido un ictus. Este proyecto ilustra cómo los guantes de datos pueden aplicarse de manera innovadora para abordar

desafíos médicos, como la rehabilitación post-ictus, ofreciendo soluciones que pueden mejorar la atención médica y la calidad de vida de los pacientes.

En resumen, los guantes de datos ofrecen un potencial significativo en una amplia gama de aplicaciones, pero su éxito depende en gran medida de la selección adecuada y la integración efectiva de los sensores, así como de la implementación de técnicas de calibración y procesamiento de señales para garantizar la precisión y confiabilidad de los datos capturados.

En este contexto, el presente trabajo se centra en el diseño y desarrollo de un periférico de entrada en forma de guante de datos, con el objetivo de capturar movimientos y gestos realizados por un usuario con el fin de controlar otras aplicaciones, como una plataforma robótica o una aplicación de realidad aumentada.

Justificación

Este trabajo se realiza para finalizar los estudios del Grado en Ingeniería Electrónica Industrial. Se sitúa al final del plan de estudios y está orientado a evaluar competencias asociadas al título que se desea adquirir [2]. El Trabajo Fin de Grado se ha desarrollado respetando la normativa del plan de estudios vigente en la Escuela Politécnica Superior de Jaén. El estudio y desarrollo de un periférico de entrada para interacción hombre-máquina con formato de guante de datos estará enmarcado en la mención de Automática de la titulación del Grado en Ingeniería Electrónica Industrial.

La elección de este proyecto viene dada a través de la asignatura Sistemas de Adquisición de Datos, donde se estudiaron los fundamentos y aplicaciones de la adquisición de datos en tiempo real. Aprovechando los conceptos adquiridos y a la motivación por un proyecto innovador y vistoso, surgió la idea del trabajo.

Objetivos

El objetivo de este Trabajo Fin de Grado en Ingeniería Electrónica Industrial tiene como principal propósito desarrollar un periférico de entrada con forma de guante de datos de manera que capte movimientos y gestos realizados por un usuario. Estos movimientos serán captados por el periférico a través de una interfaz basada en un microcontrolador con comunicaciones en serie con un computador.

A continuación, se enumeran los objetivos específicos incluidos en este trabajo:

1. **Selección y Caracterización de Sensores.** Se llevará a cabo una investigación para seleccionar los sensores más apropiados que permitan la captura precisa de los movimientos realizados por el usuario con el guante de datos. Además, se realizará una caracterización detallada de estos sensores para comprender su comportamiento y desempeño en diversas condiciones.
2. **Desarrollo de Hardware y Software.** Se desarrollarán tanto el hardware como el software necesarios para el funcionamiento integral del guante de datos. Esto incluirá el diseño y montaje de los componentes físicos del guante, así como la programación del firmware del Arduino UNO y la interfaz de usuario en LabVIEW para la comunicación y visualización de datos.
3. **Implementación y Filtrado de Datos.** Se implementarán algoritmos avanzados de filtrado y procesamiento de señales para mejorar la precisión y fiabilidad de los datos capturados por el guante de datos. Estos algoritmos se aplicarán tanto en el dispositivo de adquisición de datos como en el software de procesamiento posterior.
4. **Evaluación y Resultados.** Se llevarán a cabo pruebas para evaluar el rendimiento y la eficacia del guante de datos en escenarios de aplicación relevantes. Se analizarán los resultados obtenidos para validar la precisión y la utilidad práctica del dispositivo en situaciones del mundo real.
5. **Conclusiones y Futuras Direcciones.** Se presentarán conclusiones detalladas basadas en los resultados obtenidos de las pruebas y evaluaciones realizadas. Además, se discutirán posibles líneas de trabajo futuro para mejorar y ampliar las capacidades del guante de datos, incluyendo investigaciones adicionales, mejoras en el diseño y la implementación de nuevas características y funcionalidades.

Organización de la memoria

La memoria que aquí se presenta está dividida en las siguientes partes:

1. Introducción.

En esta sección se presenta una introducción detallada al tema central del proyecto: el guante de datos. Este dispositivo es capaz de captar y registrar múltiples parámetros físicos y ambientales a través de diferentes sensores. Se describirán algunos de los sensores más comúnmente integrados en estos

guantes, tales como los sensores de flexión, acelerómetros, giroscopios y sensores de presión. Además, se proporciona un ejemplo de aplicación de un guante de datos en el ámbito de la medicina, demostrando su utilidad en el seguimiento y rehabilitación de pacientes con trastornos motores.

Posteriormente, se expone la justificación de la elección del tema, que surge a partir de los conceptos adquiridos en la asignatura de Sistemas de Adquisición de Datos y debido a la motivación por un proyecto innovador y visualmente creativo.

Finalmente, se establecen los objetivos específicos que se pretenden alcanzar con este proyecto. Estos objetivos incluyen la selección y caracterización de los sensores, el desarrollo hardware y software, la implementación y el filtrado de datos, la evaluación y resultados, y conclusiones y líneas futuras.

2. Estado del Arte.

El objetivo de esta sección es proporcionar un conocimiento del entorno relacionado con el tema del guante de datos. Se comienza con una introducción explicativa sobre los aspectos que se abordarán en el apartado, proporcionando un marco conceptual.

En primer lugar, se presenta el concepto de guante de datos, ofreciendo una definición precisa para contextualizar su importancia en la implementación de guantes de datos. Se describen los diferentes tipos de sensores que pueden ser integrados en estos dispositivos, como sensores de flexión o acelerómetros, entre otros. Cada tipo de sensor es analizado, destacando sus características principales, principios de funcionamiento y aplicaciones específicas. Aunque este trabajo se centra en dos tipos concretos de sensores, se presentan los más tipos de sensores para facilitar una comprensión más completa del proyecto.

Finalmente, se revisan algunos trabajos previos relevantes que sirven como fundamento y referencia para la realización del presente proyecto. Estos estudios y desarrollos previos aportan un contexto histórico y métodos utilizados en este trabajo, demostrando continuidad y evolución del conocimiento en el campo del guante de datos.

3. Diseño de hardware y software.

En esta sección se describe en detalle la organización general del diseño del sistema, comenzando por la elección del microcontrolador y los sensores utilizados en este proyecto. Se explica la justificación de la selección del microcontrolador y se proporciona una descripción exhaustiva de los sensores empleados, alineados con los objetivos del trabajo.

Se presenta el esquema hardware de la aplicación, ilustrando la integración de los sensores y otros componentes electrónicos. Este esquema proporciona una visión clara de cómo se interconectan los distintos elementos del sistema para cumplir con los requisitos funcionales del proyecto.

A continuación, se detalla el circuito de acondicionamiento de la señal del sensor analógico Flex, describiendo cada una de sus partes y su función específica. Se realiza una caracterización completa del sensor Flex, evaluando su respuesta y comportamiento en distintas condiciones operativas. Además, se explica el procesamiento de los datos obtenidos de este sensor, abarcando desde la adquisición de la señal hasta su interpretación y utilización en el sistema.

Seguidamente, se describen las características del giroscopio-acelerómetro MPU6050 integrado en la aplicación. Se analiza el procesamiento de los datos proporcionados por este sensor, incluyendo técnicas de filtrado para la obtención de medidas precisas y confiables.

Finalmente, se detalla el desarrollo del software, abarcando la programación del microcontrolador para la lectura y procesamiento de los valores de los sensores. Se ofrece una explicación sobre la interfaz en LabVIEW, describiendo su estructura y funcionalidad. Aunque se proporciona una descripción general en esta sección, se remite al anexo correspondiente para una explicación más detallada en el manual de usuario.

4. Conclusiones y Líneas Futuras.

En este apartado se presentan las conclusiones obtenidas tras la realización del proyecto, evaluando el cumplimiento de los objetivos planteados y el desempeño del sistema desarrollado.

Además, se proponen posibles líneas de investigación y desarrollo futuro basadas en el trabajo realizado. Estas líneas futuras pueden incluir mejoras en el diseño de hardware, optimización del software, incorporación de nuevos tipos de sensores, y aplicaciones adicionales del guante de datos en diversos campos como la medicina, la robótica y la realidad virtual. Se identifican áreas específicas donde el proyecto puede ser ampliado o refinado, con el objetivo de maximizar su impacto y utilidad práctica.

5. Bibliografía.

En este apartado se recogen todas las fuentes de información que han sido consultadas y citadas a lo largo del desarrollo del proyecto, siguiendo el formato estandarizado de las normas IEEE.

6. Anexos

En este apartado se presenta un manual de usuario detallado, que abarca tanto los requisitos de hardware como de software necesarios para operar correctamente el sistema desarrollado. Este manual está diseñado para proporcionar un entendimiento completo y facilitar el uso eficiente del guante de datos.

ESTADO DEL ARTE

Introducción

En este capítulo se presentan diferentes sensores que pueden ser usados para la detección de gestos y movimientos que pueden ser implementados en un guante de datos, del que se ilustrará y se explicará cómo funciona y para que puede servir. Además, se presentarán diferentes trabajos relacionados con el proyecto, en los que se han usado los sensores para otro tipo de estudios.

Concepto de guante de datos

Los guantes de datos son periféricos de entrada utilizados para la interacción hombre-máquina que han sido concebidos como una extensión natural del cuerpo humano en el ámbito tecnológico. Surgieron por primera vez en 1977 gracias a Thomas de Fanti y Daniel Sandi, quienes desarrollaron el "Sayre Glove". Este dispositivo pionero empleaba tecnología de fibra óptica para medir la flexión de los dedos que funcionaba emitiendo un haz de luz en un extremo y detectando la intensidad de dicho haz en el otro extremo mediante un sensor (diodo fotoeléctrico).

Desde entonces, la evolución de los guantes de datos ha sido notable, dando lugar a nuevas versiones como el "Digital Entry Data Glove" en 1983, el "Power Glove" en 1989, mostrado en la Figura 1, y el "Super Glove" en 1995 [3]. Estos avances tecnológicos han ampliado significativamente las aplicaciones de los guantes de datos, desde su uso inicial en la interpretación del lenguaje de signos hasta campos más recientes como la medicina.



Figura 1. Nintendo Power Glove

Sensores

Un sensor es un dispositivo que detecta una determinada acción externa, temperatura, presión, etc., y la transmite adecuadamente [4]. Para este proyecto, los sensores se encargarán de detectar una acción de la mano y transmitirla de la manera más precisa posible. Para ello se podrán utilizar sensores de los siguientes tipos.

Sensor Flex

Un sensor Flex o sensor flexible es una resistencia variable en función del grado en que esté doblado, es decir, convierte la curvatura en valores de resistencia eléctrica. Son por lo general una delgada tira de unos 5 cm de largo que varía la resistencia de aproximadamente 10 a 50 k Ω , aunque hay otros modelos de distintos tamaños y resistencias, en la *Figura 2* se muestra el utilizado en el presente proyecto [5].

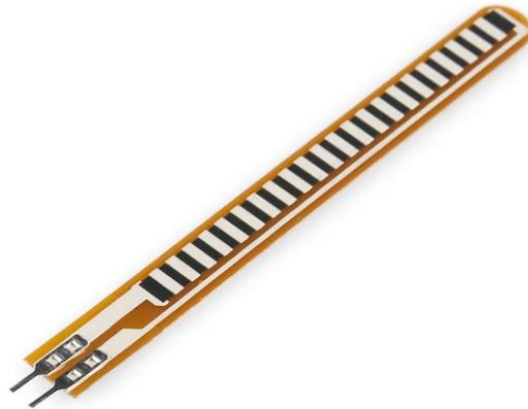


Figura 2. Sensor Flex

Los Sensores Flex son sensores resistivos que varían de manera proporcional a la flexión del sensor. Dentro de su diseño, se encuentran elementos resistivos de carbono embebidos en un sustrato flexible y delgado, en nuestro caso poliamida. La cantidad de carbono presente en el sensor determina su resistencia, siendo una relación inversamente proporcional. En otras palabras, a mayor cantidad de carbono, menor resistencia.

La elección de este sensor para implementarlo en el guante de datos se ha realizado por diversos motivos, entre ellos su bajo coste o su respuesta adecuada a las necesidades mostradas por el propio guante.

En primer lugar, su versatilidad es una de las características destacadas. Este sensor se presenta como una opción accesible y fácil de implementar ya que, como se ha expuesto anteriormente, al funcionar como una resistencia variable se puede medir la flexión de los dedos y la mano en general. Su simplicidad de uso facilita su integración en el diseño del guante y su operación por parte del usuario lo que lo convierte en una solución práctica y funcional.

Otro aspecto importante a tener en cuenta es, en comparación con otros tipos de sensores disponibles en el mercado, su bajo precio. Este factor resulta crucial, especialmente para este proyecto donde se busca optimizar el presupuesto si comprometer la calidad del prototipo final. Además del costo reducido del propio sensor, su integración en el diseño del guante requiere componentes adicionales igualmente económicos, como resistencias y amplificadores operacionales.

Finalmente, cabe destacar que el sensor Flex ofrece una respuesta adecuada a las necesidades específicas del guante de datos. Su capacidad para medir la flexión de los dedos y a su bajo coste lo convierte en una herramienta ideal para capturar los movimientos de la mano con fines de interacción hombre-máquina.

Por estos motivos, la elección de un sensor que sea capaz de medir el grado de flexión de los dedos se hace sencilla. Tanto es así, que Spectra Symbol, empresa que ha patentado esta tecnología, afirma que estos sensores se utilizaron en el guante original de Nintendo Power Glove [6]. En la actualidad puede que nos hayamos acostumbrado a este tipo de tecnologías, pero Nintendo fue pionera en el uso de este tipo de sensores, siendo el lanzamiento de dicho dispositivo en 1989.

En la Figura 3 se muestran los sensores Flex en un guante de datos.

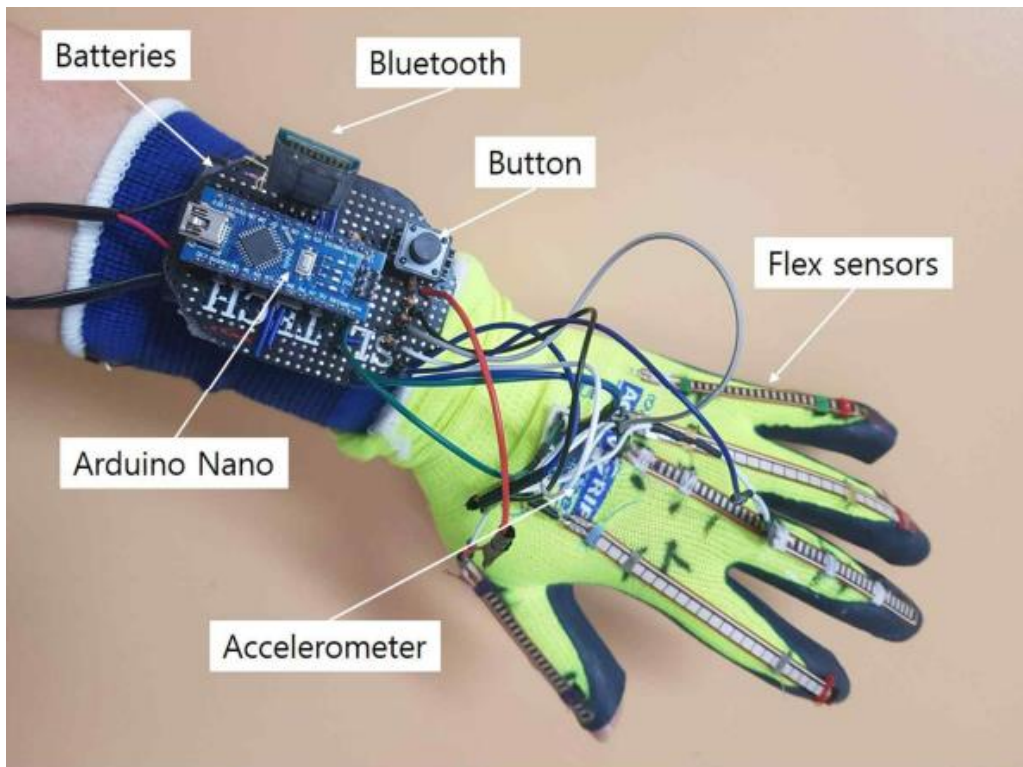


Figura 3. Guante de datos con sensores Flex

Acelerómetro

Los acelerómetros son dispositivos electromecánicos que detectan las fuerzas de aceleración, ya sea estática o dinámica. Las fuerzas estáticas incluyen la gravedad, mientras que las fuerzas dinámicas pueden incluir vibraciones y movimiento. Los acelerómetros pueden medir la aceleración en uno, dos o tres ejes [7].

Generalmente, los acelerómetros contienen placas capacitivas internamente. Algunos de estos son fijos, mientras que otros están unidos a resortes minúsculos que se mueven internamente conforme las fuerzas de aceleración actúan sobre el sensor, como se puede ver en la Figura 4. Como estas placas se mueven en relación el uno al otro, la capacitancia entre ellos cambia. A partir de estos cambios en la capacitancia, la aceleración se puede determinar [8].

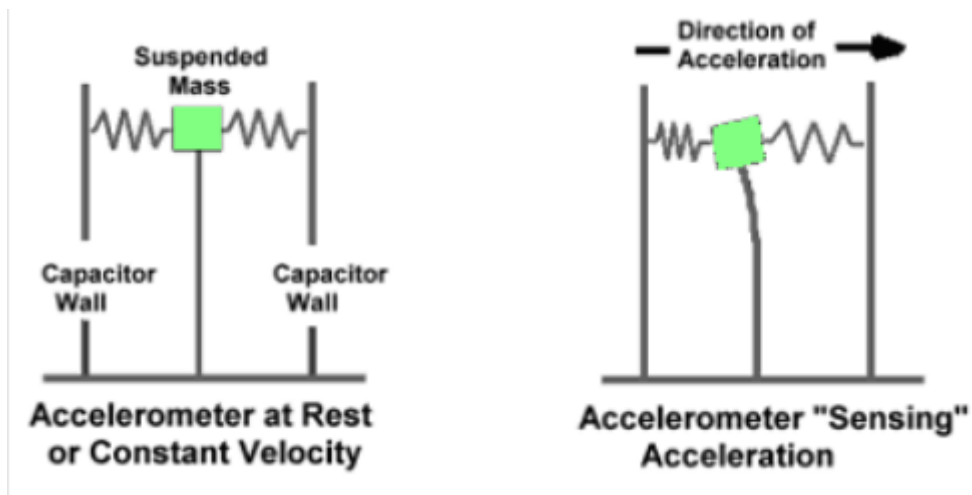


Figura 4. Principio de funcionamiento de un acelerómetro

Giroscopio

El giroscopio es un instrumento capaz de medir las velocidades angulares. Puede ser analógico o electrónico, pero en lo que concierne a este proyecto solo es el electrónico. Por lo tanto, de ahora en adelante cuando se hable de giroscopio se refiere a un sensor electrónico capaz de medir la velocidad angular.

El giroscopio tiene un sensor MEMS (MicroElectroMechanical Systems) muy pequeño dentro. Cuando se hace girar el giroscopio como se ve en la *Figura 5*, una pequeña masa de resonancia se desplaza con los cambios de velocidad angular. Este movimiento se convierte en señales eléctricas de muy bajas corrientes que se pueden amplificar para ser leídas por un microcontrolador [9].

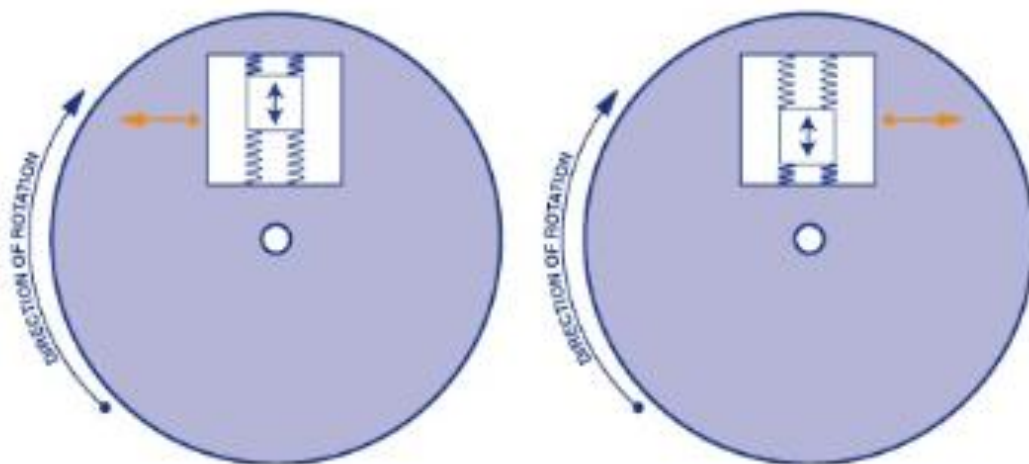


Figura 5. Funcionamiento interno de un sensor MEMS

IMU

IMU, que significa Unidad de Medición Inercial, es un dispositivo electrónico que mide e informa de la aceleración, la orientación, las velocidades angulares y otras fuerzas gravitatorias.

Existen varios tipos de sensores IMU: basados en FOG (giroscopio de fibra óptica), RLG (giroscopio láser de anillo) y, por último, basado en tecnología MEMS, que serán del tipo que tenemos en este proyecto [10].

En definitiva, esta opción es mucho más completa que las anteriores, pues al constar de un acelerómetro y un giroscopio que trabajan en conjunto, puede obtener medidas más reales que uno de los sensores trabajando individualmente. En la Figura 6 se muestran los ejes de trabajo del sensor IMU MPU6050 de 6 DOF (Degrees of Freedom), concretamente el que se usará en este trabajo.

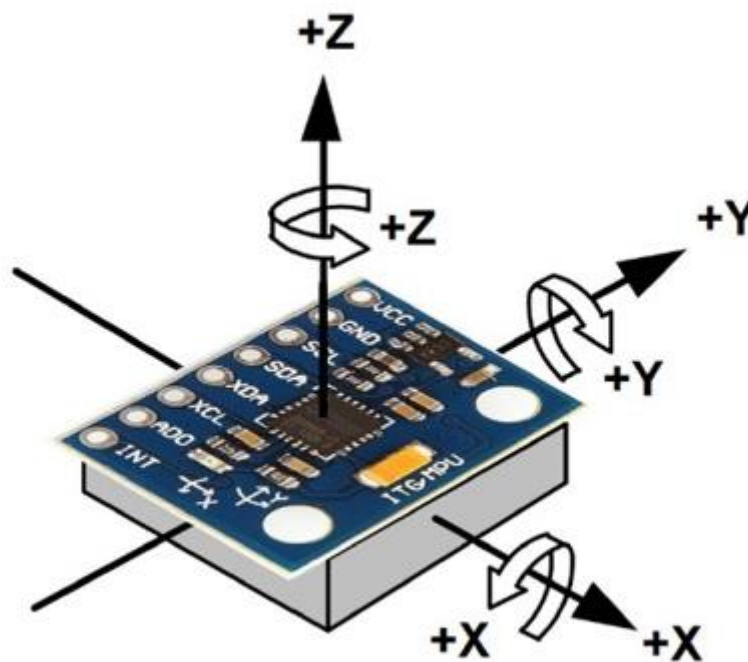


Figura 6. Ejes de trabajo del sensor IMU MPU6050

Este sensor puede comunicarse tanto a través de SPI como de bus I2C, lo que facilita la integración con otros dispositivos y microcontroladores. Además, su bajo requerimiento de voltaje de alimentación lo hace compatible con una variedad de sistemas electrónicos.

La disponibilidad de módulos preintegrados, como el GY-521, que incluyen la electrónica necesaria para conectar el MPU-6050 fácilmente a plataformas como Arduino, simplifica aún más su uso en proyectos de electrónica y robótica. Estos módulos suelen incorporar un regulador de voltaje que permite alimentar el sensor directamente con 5V, aumentando su compatibilidad con sistemas existentes.

Otro factor importante es la precisión y configurabilidad del MPU-6050. Con convertidores analógicos digitales (ADC) de 16 bits y la capacidad de ajustar el rango del acelerómetro y el giroscopio, este sensor ofrece una gran flexibilidad para adaptarse a diferentes aplicaciones y entornos de trabajo. Además, su procesador interno (DMP, Digital Motion Processor) ejecuta complejos algoritmos de MotionFusion, combinando las mediciones de los sensores internos y simplificando el procesamiento de datos en aplicaciones básicas.

Por último, el MPU-6050 destaca por su relación calidad-precio. Con un costo aproximado de 1.10€ en vendedores internacionales [11], este sensor ofrece un excelente rendimiento a un precio accesible, lo que lo convierte en una opción atractiva y ampliamente utilizada en proyectos de electrónica y robótica.

Trabajos previos

- Proyecto fin de carrera: D. Segovia, Universidad Carlos III de Madrid

En el año 2018, Daniel Segovia realizó su trabajo de fin de grado en la universidad Carlos III de Madrid, titulado “Adquisición de datos desde un sensor inercial” [12]. En él, se hace uso de un sensor inercial MPU6050 para medir las vibraciones de un vehículo ferroviario con el fin de averiguar el estado de los raíles por los que rueda.

En este proyecto, se hace uso del mismo sensor inercial, entre otros sensores utilizados. Aunque el uso del sensor inercial es distinto, el principio de funcionamiento es el mismo y ha servido como base para empezar a entender y analizar el sensor.

- Artículo: A Prototype of Flex Sensor Based Data Gloves to Track the Movements of Fingers

En 2019, la revista “Smart Media Journal” de Gwangju en Corea del Sur publicó este artículo que propone un guante de datos basado en sensores flexibles para realizar un seguimiento de los dedos humanos para la educación de realidad virtual, financiado por el ministerio de educación de Corea del Sur [13].

A pesar de que el fin de este artículo es de carácter educativo, el fundamento del guante con los sensores es el mismo que en el presente proyecto, lo que nos servirá más adelante como precedente para la implementación del guante.

DISEÑO DE HARDWARE Y SOFTWARE

Organización de diseño

El diseño de este proyecto se compone de sensores, etapas acondicionadoras, un microcontrolador y un ordenador que muestra en pantalla una aplicación final.

El sensor MPU6050 proporciona, a través de su acelerómetro, los datos de inclinación de la mano que se pondrá el guante de datos. Dichos datos necesitan ser procesados para dar una respuesta más precisa. La forma de hacer este procesado es a través de filtros internos del propio dispositivo y mediante la programación en el microcontrolador. La conexión del sensor se hará mediante el bus de comunicaciones I2C, protocolo de comunicación serial que define la trama de datos y las conexiones físicas para transferir datos entre dos dispositivos digitales.

Por otro lado, el sensor Flex necesitará una etapa acondicionadora para ampliar su rango de datos y así obtener la mayor precisión posible. A pesar de que más tarde se filtrarán estos datos a través de programación en el microcontrolador, esta etapa de acondicionamiento se hace necesaria. La conexión de este sensor y su etapa de acondicionamiento se hace mediante una entrada analógica del microcontrolador.

Una vez conectados ambos sensores al microcontrolador Arduino UNO, éste realizará el procesado comentado para cada sensor correspondiente y dará una respuesta en forma de tres datos, inclinación del sensor MPU6050 en dos ejes (X e Y) y flexión del sensor Flex. Desde este punto, se conecta con el ordenador vía USB.

Por último, el ordenador presentará una aplicación en LabVIEW, entorno de programación gráfica que, en este caso, se encarga de mostrar por pantalla los datos de los sensores ya procesados y en forma de gráficas para hacer visual y más entendibles los valores obtenidos.

Todo este diseño se muestra en forma de diagrama de bloques en la *Figura 7*.

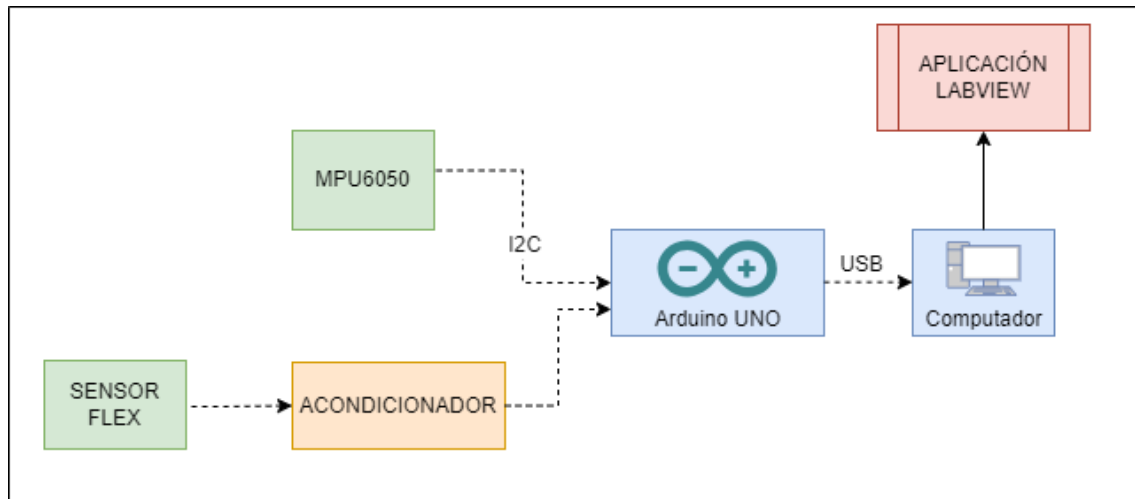


Figura 7. Diagrama de bloques del diseño

Elección del microcontrolador

La elección del microcontrolador no abordó gran parte del tiempo, ya que desde el primer momento se pudo ver la capacidad de la placa Arduino UNO y las facilidades que ofrece.

El microcontrolador destaca por varias características que lo hacen una opción sólida, especialmente porque la alimentación requerida para el proyecto es de 5 V y si la funcionalidad que ofrece es suficiente para las necesidades del diseño. Una de sus ventajas principales es su amplia compatibilidad. Como uno de los modelos más populares de la familia Arduino, el UNO cuenta con una gran cantidad de documentación, tutoriales y soporte comunitario disponibles en línea. Esta abundante información puede facilitar significativamente el desarrollo y la resolución de problemas durante el proceso de diseño.

Además de su popularidad, el Arduino UNO se distingue por su facilidad de uso. La programación se realiza en un entorno de desarrollo integrado (IDE) intuitivo y fácil de utilizar, lo que agiliza el proceso de desarrollo y reduce la curva de aprendizaje para los usuarios menos experimentados.

Otra ventaja del Arduino UNO es su disponibilidad de pines GPIO. Con un número suficiente de pines de entrada/salida general, esta placa facilita la conexión de sensores necesarios para el proyecto. Esto permite una mayor flexibilidad en la implementación de diversas funcionalidades, sin necesidad de utilizar adaptadores adicionales.

Además de sus características técnicas, el Arduino UNO ofrece una ventaja adicional en términos de costos. En comparación con placas microcontroladoras más avanzadas como el ESP32, el UNO suele ser más económico, lo que lo convierte en una opción atractiva para proyectos con presupuestos ajustados. Si bien el Arduino NANO también es una opción económica, el UNO puede ser preferible debido a su robustez y estabilidad, especialmente en proyectos donde estas cualidades son prioritarias.

En resumen, el Arduino UNO es una opción sólida si se requiere una alimentación de 5V y si la funcionalidad ofrecida por esta placa es adecuada para el proyecto. Su amplia compatibilidad, facilidad de uso, disponibilidad de pines GPIO, compatibilidad con 5V y costo accesible lo convierten en una opción confiable para esta aplicación de diseño.

En la *Figura 8* se puede ver una vista superior de una placa Arduino UNO en la que se pueden apreciar las entradas, salidas y demás componentes de la misma [14].

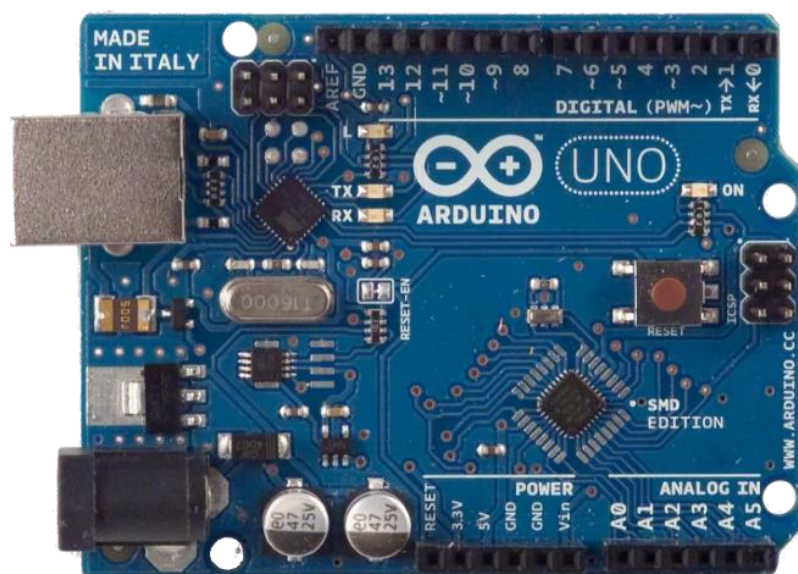


Figura 8. Placa Arduino UNO

Esquema hardware de la aplicación

El esquema hardware del proyecto se ha diseñado para capturar de manera precisa la flexión de los dedos y la inclinación de la mano en dos ejes relevantes mediante el uso de dos sensores principales: un sensor Flex y un sensor MPU6050.

El sensor Flex se encuentra conectado a tierra. Su salida se dirige hacia una etapa de acondicionamiento, la cual amplía y ajusta la señal para obtener una respuesta óptima del sensor. Los detalles específicos de esta etapa de acondicionamiento se abordarán en otro apartado dedicado.

Una vez procesada la señal del sensor Flex mediante la etapa de acondicionamiento, se conecta al microcontrolador Arduino mediante el pin A0. Esta conexión permite que el Arduino procese y utilice los datos proporcionados por el sensor Flex para aplicaciones futuras.

El sensor MPU6050 se enlaza con el Arduino mediante tres conexiones clave: INT, SCL y SDA. La conexión INT, destinada a la gestión de interrupciones, se conecta a una entrada digital del Arduino, permitiendo al microcontrolador detectar eventos específicos generados por el sensor. Por otro lado, las conexiones SCL y SDA, que implementan el protocolo de comunicación I2C, se vinculan a los pines A5 y A4 del Arduino, respectivamente. Estas conexiones facilitan la transmisión bidireccional de datos entre el Arduino y el sensor MPU6050, lo que permite la adquisición de datos de aceleración y velocidad angular con una alta precisión y eficiencia.

Es importante destacar que, si bien la conexión INT está presente en el esquema de conexión, no se utilizará en el presente trabajo. Sin embargo, es relevante mencionarla para futuras líneas de trabajo, donde podría ser aprovechada para gestionar interrupciones y eventos específicos del sensor. Además, aunque el protocolo de comunicación I2C permite una transmisión bidireccional de datos, en el contexto de este proyecto, la comunicación se realiza de manera unidireccional, desde el sensor MPU6050 hacia el Arduino. Esta configuración simplifica la implementación y el procesamiento de los datos, asegurando una adquisición eficiente y precisa de la información de inclinación proporcionada por el sensor.

Tanto el sensor MPU6050 como la etapa de acondicionamiento se alimentan conectándolos a masa y a V_{cc} de la propia placa Arduino, siendo los pines GND y 5 V respectivamente, garantizando así un suministro eléctrico adecuado para su funcionamiento correcto.

Es crucial destacar que la selección del amplificador operacional conlleva una importancia equiparable a la del sensor mismo, dado que su rango de salida determina el dimensionado de las resistencias asociadas. En este sentido, se optó por el LM324 debido a su disponibilidad de cuatro amplificadores integrados. Además, este componente satisface criterios esenciales, como una ganancia elevada y un voltaje de entrada de 0 a 5 voltios, factores determinantes para el adecuado funcionamiento del sistema.

En conjunto, este esquema hardware, el cual se puede ver en la *Figura 9*, proporciona la base necesaria para la captura precisa de la flexión de los dedos e inclinación de la mano, permitiendo así el desarrollo de aplicaciones avanzadas basadas en los datos recopilados por estos sensores.

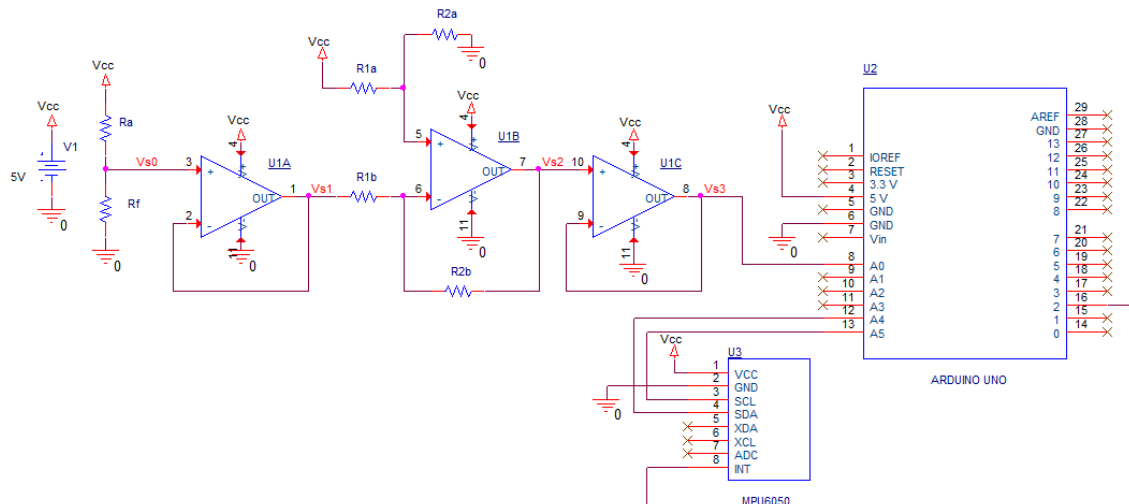


Figura 9. Esquema hardware completo

Acondicionamiento de señal del sensor Flex

La etapa de acondicionamiento del sensor Flex se ha diseñado con el propósito de expandir el rango de tensión disponible en el nodo del divisor de tensión. Originalmente, el rango de voltaje proporcionado por el sensor Flex se encuentra dentro de límites estrechos, de entre 1.87 V y 2.99 V con una alimentación de 0-5 V, lo que limita el aprovechamiento del rango completo de entrada (FSR) del convertidor AD y la resolución del mismo. Por ello, se ha implementado un circuito acondicionador que amplía el rango de voltaje disponible, permitiendo así una mayor resolución en la detección de la flexión.

El circuito se divide en cuatro etapas diferenciadas: un divisor de tensión en primer lugar, un buffer, un amplificador en modo sumador-restador y un segundo buffer. Para abordar una mejor comprensión del circuito, cada parte se explicará por separado.

Divisor de tensión

A pesar de que el acondicionamiento del sensor Flex viene a continuación de esta etapa, es necesario comentar que ésta primera se trata de un divisor de tensión entre una resistencia de pull-up (R_a) y la resistencia del propio sensor Flex (R_f).

Un divisor de tensión es un circuito eléctrico compuesto entre dos resistencias conectadas en serie, en este caso R_a y R_f , entre una fuente de tensión y tierra. La tensión entre los dos puntos de conexión de las resistencias

se divide de manera proporcional a las resistencias individuales. La tensión en el punto intermedio del divisor, conocido como nodo, es una fracción de la tensión total de la fuente y se puede calcular usando la ley de Ohm.

Al ser R_f una resistencia variable, la tensión en el nodo variará en función a la flexión del sensor, lo que permite medir la misma mediante la lectura de tensión con el microcontrolador.

Primer buffer

El primer buffer desempeña un papel crucial al desacoplar las etapas del sensor y del amplificador. Su función principal es permitir el diseño de la resistencia de pull-up (R_a) sin considerar la resistencia de entrada del amplificador. Esto asegura una mayor flexibilidad en el diseño y optimización de la etapa de acondicionamiento.

Buffer de salida

Se ha incorporado una etapa seguidora de tensión en la salida del circuito acondicionador para garantizar una impedancia de salida inferior a $10\text{k}\Omega$, conforme a las especificaciones del fabricante de la placa Arduino UNO. Esta etapa, además de reducir la impedancia de salida, también actúa como driver del condensador de entrada del convertidor analógico-digital (ADC), asegurando una correcta conversión analógico-digital. La configuración de esta etapa se muestra en la *Figura 10*, que presenta el modelo de entrada del ADC del microcontrolador ATmega328P [15].

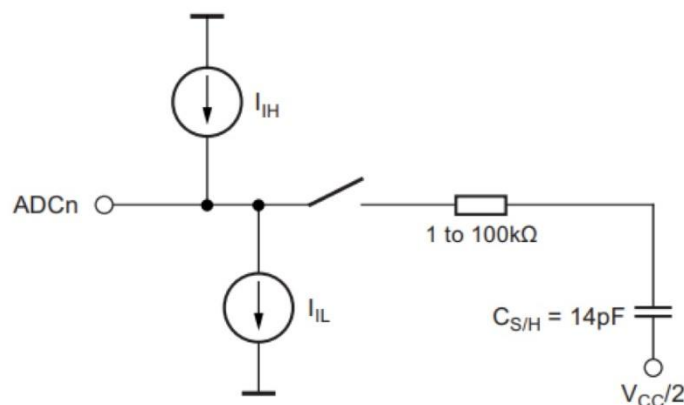


Figura 10. Conversor ADC del microcontrolador ATmega328P. Fuente: Microchip.

Sumador-restador

El amplificador operacional configurado como sumador-restador es esencial para adaptar la señal del sensor al rango de entrada del ADC. Sin embargo, debido a la alimentación empleada, el rango máximo de la señal de

salida del amplificador operacional es ligeramente inferior al rango dinámico óptimo del ADC. A pesar de esta limitación, se logra un rango de señal mejorado en comparación con la salida directa del sensor, y muy cercano al rango óptimo del ADC.

Se considera una alternativa para aumentar el rango de salida del amplificador hasta abarcar todo el rango del ADC, es decir, $[0, 5]$ V, incluyendo una fuente de alimentación bipolar. Sin embargo, por razones de simplicidad y conveniencia, se ha optado por utilizar la fuente de alimentación proporcionada por el Arduino, lo que simplifica el diseño y la implementación del circuito de acondicionamiento.

Elección de resistencia de pull-up (R_a)

Para maximizar la sensibilidad del sensor se debe seleccionar R_a procurando maximizar la variación de la V_{s0} ($\Delta V_{s0} = V_{s0\max} - V_{s0\min}$) para todo el rango de flexión del sensor. Para ello analizó la curva de la dependencia de esa magnitud respecto de R_a como se presenta en la *Figura 11* para el sensor 1. Aunque el valor óptimo de resistencia es de $44\text{k}\Omega$, se escoge una resistencia de $47\text{k}\Omega$, por ser un valor normalizado, que consigue también valores muy altos en el rango de V_{s0} .

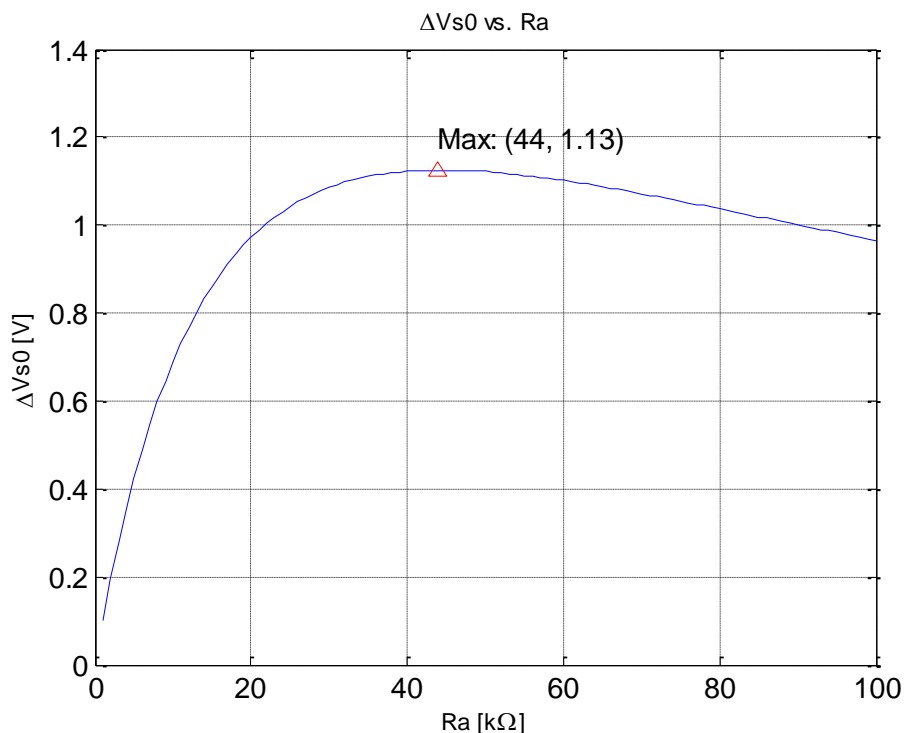


Figura 11. Curva de valores de R_a frente a la variación de V_{s0}

Los resultados para los tres sensores se resumen en la Tabla 1, que muestra, para cada sensor Flex, los valores máximos y mínimos de resistencia

del sensor, el valor óptimo de R_a para cada sensor, y el rango de variación de la tensión de salida (V_{s1}) con una alimentación de 5V para cada sensor.

	RESISTENCIA MIN-MAX [k Ω]	Ra ÓPTIMA (k Ω)	VARIACIÓN MÁXIMA DE LA TENSIÓN DE SALIDA ΔV_{s0} (V)
SENSOR 1	28-70	44	1.13
SENSOR 2	28-55	39	0.84
SENSOR 3	25-40	33	0.44

Tabla 1. Ra óptima para diferentes sensores

Es importante señalar que existe una notable disparidad en este intervalo entre los diferentes sensores, lo cual dificulta utilizar el mismo diseño acondicionador para todos ellos. Esta información proporcionará una guía para la selección adecuada de la resistencia R_a . Dada la alta dispersión que presentan los diferentes sensores se recomienda elegir la resistencia R_a para uno de los sensores, en nuestro caso se ha utilizado el sensor 1.

Diseño de la etapa amplificadora

La etapa amplificadora se basa en una configuración sumador-restador. El objetivo de esta etapa es mapear el rango de salida del voltaje V_{s0} a un rango más adecuado para aprovechar al máximo el rango de entrada del conversor AD (0 a 5V).

En este circuito, las resistencias deben calcularse para ajustar las ganancias de modo que se aproveche el rango de salida de los elementos más restrictivos. En este caso son los propios amplificadores operacionales debido a que tienen alimentación unipolar de 5 V, proporcionada por la placa Arduino UNO y alcanzan la saturación a un rango de salida medido de [0.023, 3.812] V. Esta solución no aprovecha al completo el rango de entrada del AD, igual a [0, 5] V, aunque es una solución simple al no necesitar una fuente de alimentación extra, y que mejora significativamente el rango de la señal de entrada. Como se puede ver en la Figura 12 y apreciar sus valores más significativos en la Tabla 2.

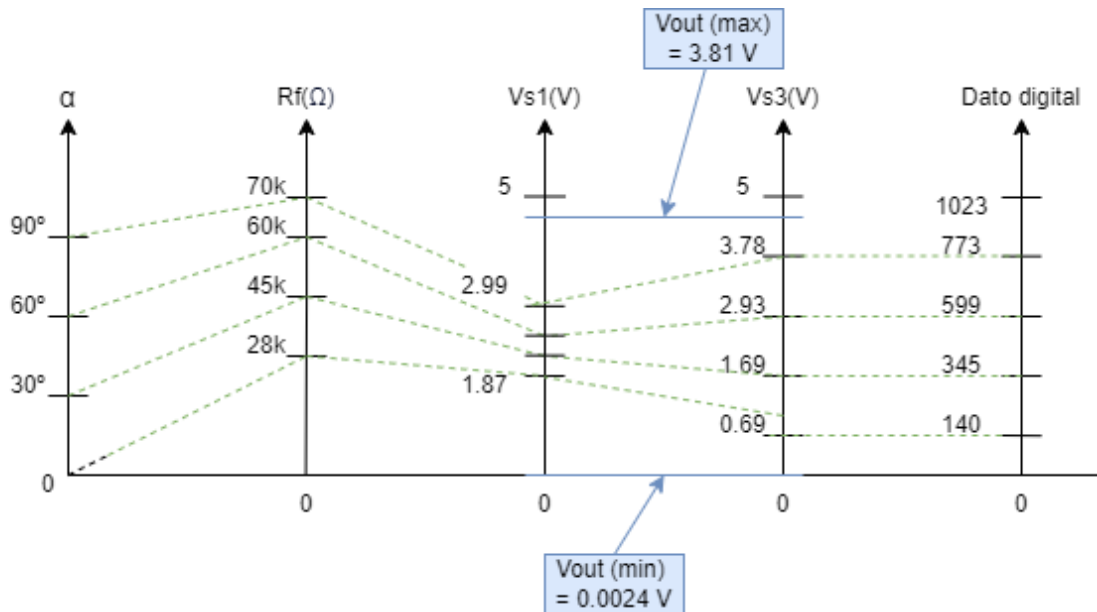


Figura 12. Gráfica de resistencia y voltaje en función de la flexión del sensor

	Valor máximo	Valor mínimo
α	0°	90°
R _f [kΩ]	28k	70k
V _{s1} [V]	1.87	2.99
V _{s3} [V]	0.69	3.78
Dato digital	140	773

Tabla 2. Rangos de magnitudes empleadas para el cálculo de resistencias

Se han realizado los cálculos de las resistencias y tensiones teniendo como modelo un sensor Flex 1 de la Tabla 1. Además, se ha utilizado una resistencia de pull-up R_a de 47 kΩ, como se ha mencionado anteriormente.

En primer lugar, se han calculado los valores máximo y mínimo de V_{s0} de la siguiente forma:

- Con R_f^{max} = 70kΩ :

$$V_{s0}^{max} = \frac{R_f}{R_a + R_f} \cdot V_{cc} = 2.99 V \tag{1}$$

- Con R_f^{min} = 28kΩ :

$$V_{s0}^{min} = \frac{R_f}{R_a + R_f} \cdot V_{cc} = 1,867 V \tag{2}$$

A continuación, planteamos la fórmula necesaria para calcular el valor de V_{s2} , siendo las ganancias:

$$K_a = \frac{R_{2a}}{R_{1a} + R_{2a}} \quad (3)$$

$$G_b = \frac{R_{2b}}{R_{1b}} \quad (4)$$

Luego:

$$V_{s2} = \left(1 + \frac{R_{2b}}{R_{1b}}\right) \cdot \left(\frac{R_{2a}}{R_{1a} + R_{2a}}\right) \cdot V_{cc} - \left(\frac{R_{2b}}{R_{1b}}\right) \cdot V_{s1} \quad (5)$$

De esta forma, sabiendo que $V_{s0} = V_{s1}$, se pueden calcular los valores tanto de K_a , como de G_b tal que:

$$V_{s2}^{max} - V_{s2}^{min} = (V_{s1}^{max} - V_{s1}^{min}) \cdot G_b \quad (6)$$

$$G_b = \frac{\Delta V_{s2}}{\Delta V_{s1}} = 3.368$$

Para calcular K_a se escoge la ecuación (5) para V_{s2}^{max} , valor que será máximo cuando V_{s1}^{min} :

$$V_{s2}^{max} = (1 + G_b) \cdot V_{cc} \cdot K_a - V_{s1}^{min} \cdot G_b \quad (7)$$

$$K_a = \frac{V_{s2}^{max} + V_{s1}^{min} \cdot G_b}{(1 + G_b) \cdot V_{cc}} = 0.475 \quad (8)$$

Sabiendo que los valores de ganancia deben ser constantes y con los valores de V_{s2}^{max} y V_{s2}^{min} ya obtenidos, se pueden calcular los valores de las resistencias, ajustándolos al máximo a los valores de resistencias normalizadas:

$$K_a = \frac{R_{2a}}{R_{1a} + R_{2a}} \rightarrow R_{2a} = \frac{K_a \cdot R_{1a}}{1 - K_a} \quad (9)$$

Suponiendo $R_{1a} = 56k\Omega \rightarrow R_{2a} = 50.6 k\Omega$. Por lo que se escoge como valor normalizado una resistencia de $51 k\Omega$.

$$G_b = \frac{R_{2b}}{R_{1b}} \rightarrow R_{2b} = G_b \cdot R_{1b} \quad (10)$$

Suponiendo $R_{1b} = 10k\Omega \rightarrow R_{2b} = 27k\Omega$.

Por último, con los valores de las resistencias ya calculados, a través de la ecuación (5) se pueden calcular los valores reales de V_{s2}^{max} y V_{s2}^{min} :

$$V_{s2}^{max} = \left(1 + \frac{R_{2b}}{R_{1b}}\right) \cdot \left(\frac{R_{2a}}{R_{1a} + R_{2a}}\right) \cdot V_{cc} - \left(\frac{R_{2b}}{R_{1b}}\right) \cdot V_{s1}^{min} = 3.78 V \quad (11)$$

$$V_{s2}^{min} = \left(1 + \frac{R_{2b}}{R_{1b}}\right) \cdot \left(\frac{R_{2a}}{R_{1a} + R_{2a}}\right) \cdot V_{cc} - \left(\frac{R_{2b}}{R_{1b}}\right) \cdot V_{s1}^{max} = 0.69 V \quad (12)$$

Finalmente, los valores de todas las resistencias utilizadas en el montaje se muestran en la *Tabla 3*.

Resistencia	Valor serie E-24 [kΩ]
R_a	47
R_f	28 – 70
R_{1a}	56
R_{2a}	51
R_{1b}	10
R_{2b}	27

Tabla 3. Valores de resistencias según la serie E-24.

Por otro lado, en la *Tabla 4* se muestran los valores de tensión correspondientes a los puntos más significativos del montaje.

Señal de voltaje	Medida [V]
V_{s1}	1.867 – 2.99
V_{s3}	0.69 – 3.78

Tabla 4. Valores de tensión para el circuito montado.

Caracterización del sensor Flex

El presente apartado detalla la caracterización del sensor Flex, enfocado en la adaptación de su señal de salida para mejorar su capacidad de discriminación angular. Tras haber realizado un acondicionamiento de señal con el objetivo de ampliar el rango de salida del sensor, optimizando así su capacidad de detección y precisión.

Inicialmente, el rango de salida del sensor era considerablemente estrecho en comparación con el rango máximo de adquisición de datos. Sin embargo, mediante el acondicionamiento de señal, se amplió significativamente el espectro de valores de salida del sensor. Esta expansión representa un avance sustancial en la capacidad del sensor para detectar y procesar variaciones en la flexión. La ampliación del rango de salida permite una mejor discriminación de los diferentes niveles de flexión del sensor, lo que resulta esencial para aplicaciones que requieren una medición precisa de ángulos.

Este proceso de acondicionamiento ha permitido la discriminación de al menos cuatro valores diferenciados de flexión del sensor, correspondientes a ángulos de 0°, 30°, 60° y 90°. La capacidad de detectar estos ángulos con precisión es crucial para aplicaciones donde se requiere una medición exacta.

Aunque tomando como referencia el sensor Flex, se puede decir que la flexión será de 0° a 90°, para la aplicación real de un guante de datos, cuando se refiere a 0° estaremos hablando del dedo estirado por completo y cuando se refiere a 90°, estaría flexionado al máximo que permite la fisionomía humana.

Por último, cabe comentar que la recolección de muestras se llevó a cabo utilizando un montaje experimental diseñado específicamente para este propósito. El sensor fue instalado sobre un soporte rígido que garantizara su estabilidad y minimizara cualquier variación en su flexión durante las mediciones. Es importante destacar que, dado el principio de funcionamiento del sensor, basado en la deformación del sustrato que contiene, se evitó aplicar presión directa sobre el mismo para evitar posibles distorsiones en los resultados. La configuración experimental se ilustra en la *Figura 13*, donde se observa el montaje utilizado para asegurar condiciones óptimas durante el proceso de adquisición de datos.

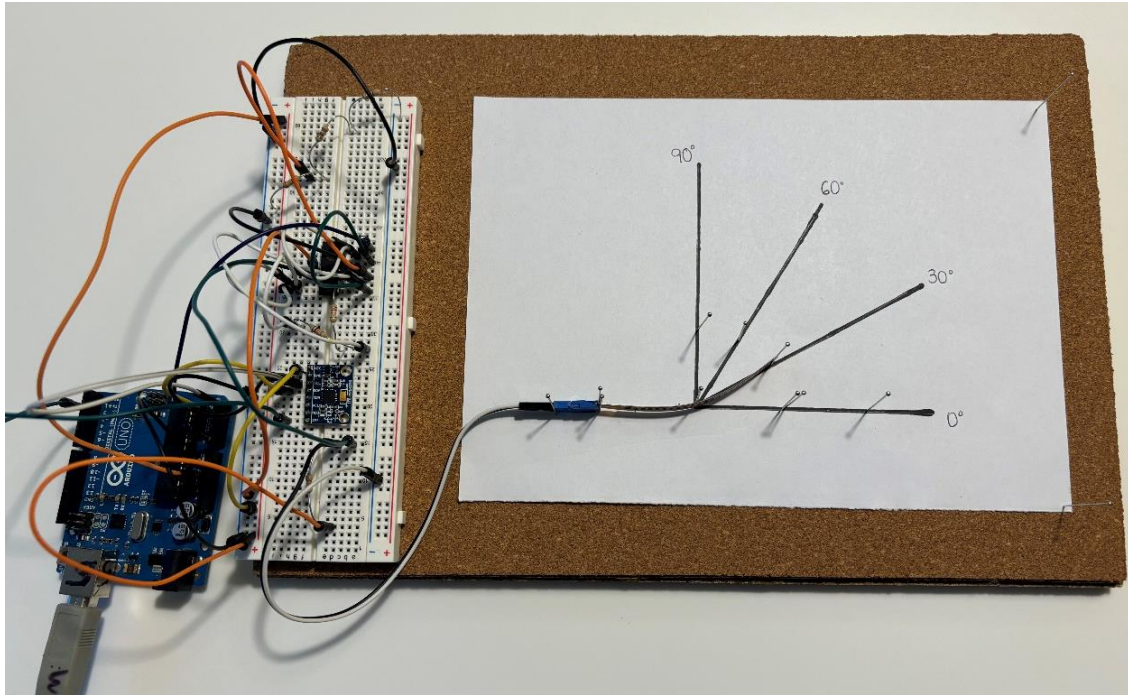


Figura 13. Montaje para toma de datos de cada valor de flexión

Caracterización

La caracterización se ha realizado mediante el análisis de histogramas para observar la dispersión de las medidas y cuantificar el error, asegurando la repetitividad de las medidas.

A continuación, se presentan los histogramas correspondientes a los cuatro ángulos de flexión del sensor: 0°, 30°, 60° y 90° respectivamente. Cada histograma se ha generado a partir de la recopilación de datos mediante mediciones realizadas a intervalos regulares de 20 milisegundos, con un total de 200 muestras por ángulo de flexión.

Al observar el histograma correspondiente al sensor en su estado estirado al máximo, se observa una distribución significativa de valores, donde la mayoría de las 200 muestras recopiladas se concentran en torno a un valor específico. Este fenómeno indica una respuesta característica del sensor en condiciones de máxima extensión, evidenciando una tendencia hacia la centralización de las mediciones alrededor de dicho valor. Además, se nota una ausencia de dispersión significativa en los datos, como se ilustra en la *Figura 14*.

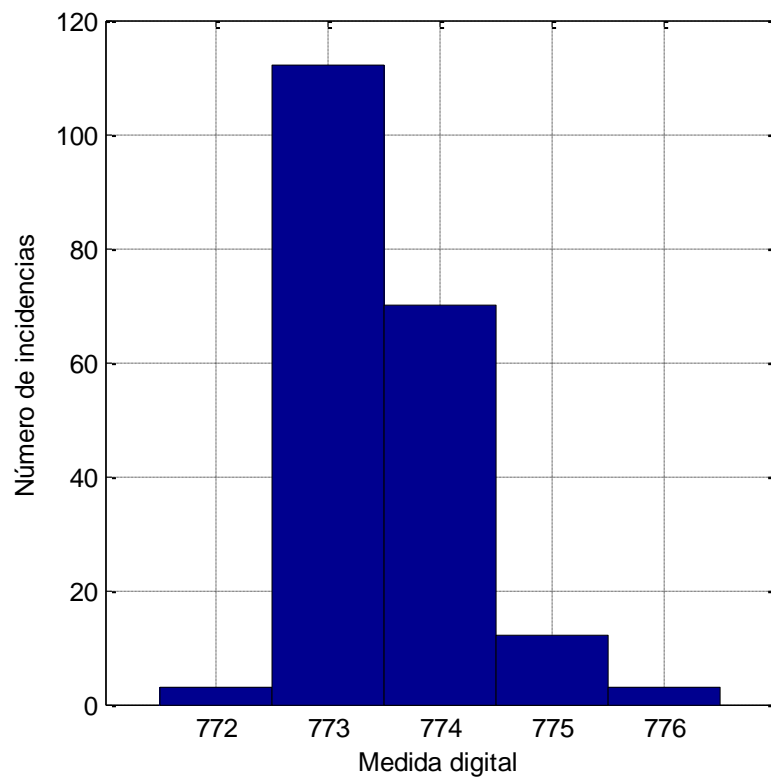


Figura 14. Histograma asociado al ángulo de flexión de 0°

En cuanto al ángulo de flexión de 30°, el histograma muestra una distribución similar, aunque con la particularidad de que dos valores distintos muestran una frecuencia significativa de repeticiones. A pesar de esta diferencia, la respuesta del sensor sigue siendo consistente y en línea con las expectativas, mostrando una adecuada capacidad de discriminación angular. Se destaca también una reducida cantidad de mediciones en los extremos del rango de valores, como se observa en la *Figura 15*.

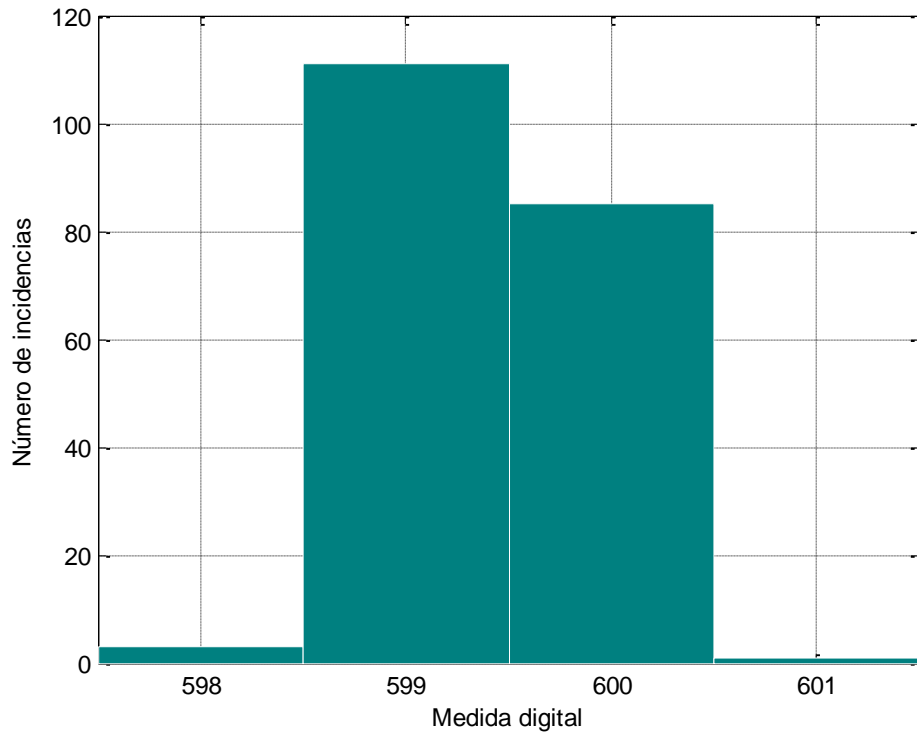


Figura 15. Histograma asociado al ángulo de flexión de 30°

Para el ángulo de flexión de 60°, el histograma exhibe una distribución más compacta, con una marcada predominancia de un solo valor digital. A pesar de esta concentración, la distribución general del histograma sigue siendo coherente con los patrones observados en ángulos de flexión anteriores. Esta consistencia en la respuesta del sensor se refleja en el histograma correspondiente, que se presenta en la *Figura 16*.

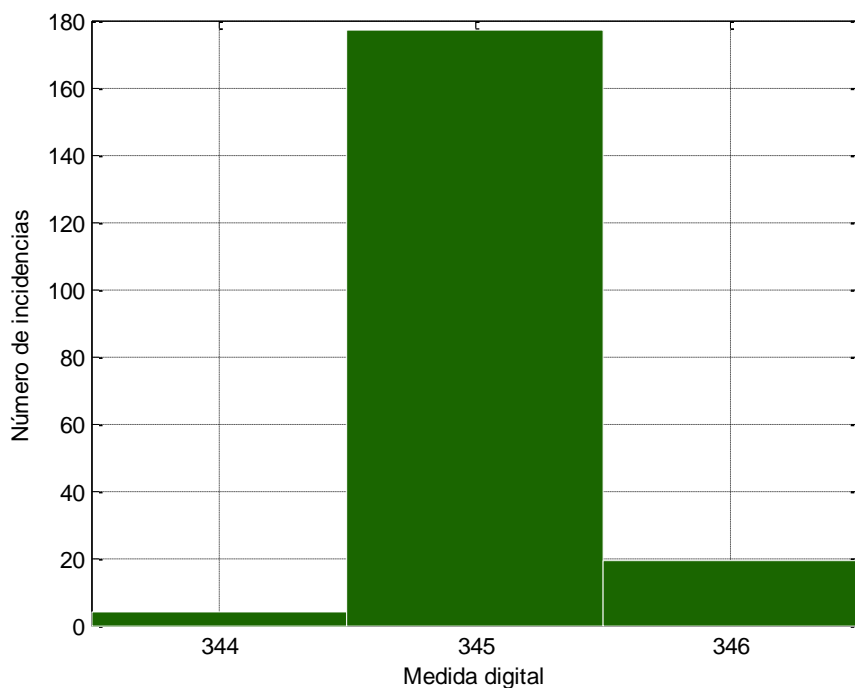


Figura 16. Histograma asociado al ángulo de flexión de 60°

Finalmente, en el histograma asociado al ángulo de flexión de 90° , se identifica una distribución con una marcada concentración de valores digitales en el extremo inferior del rango. Esta respuesta característica del sensor ante una flexión máxima se traduce en una mayor concentración de mediciones en valores digitales cercanos al límite inferior del rango de detección. Aunque se observa una tendencia hacia uno o dos valores específicos con mayor incidencia, se aprecia una mayor dispersión de los valores en comparación con los histogramas de ángulos anteriores, como se puede observar en la *Figura 17*.

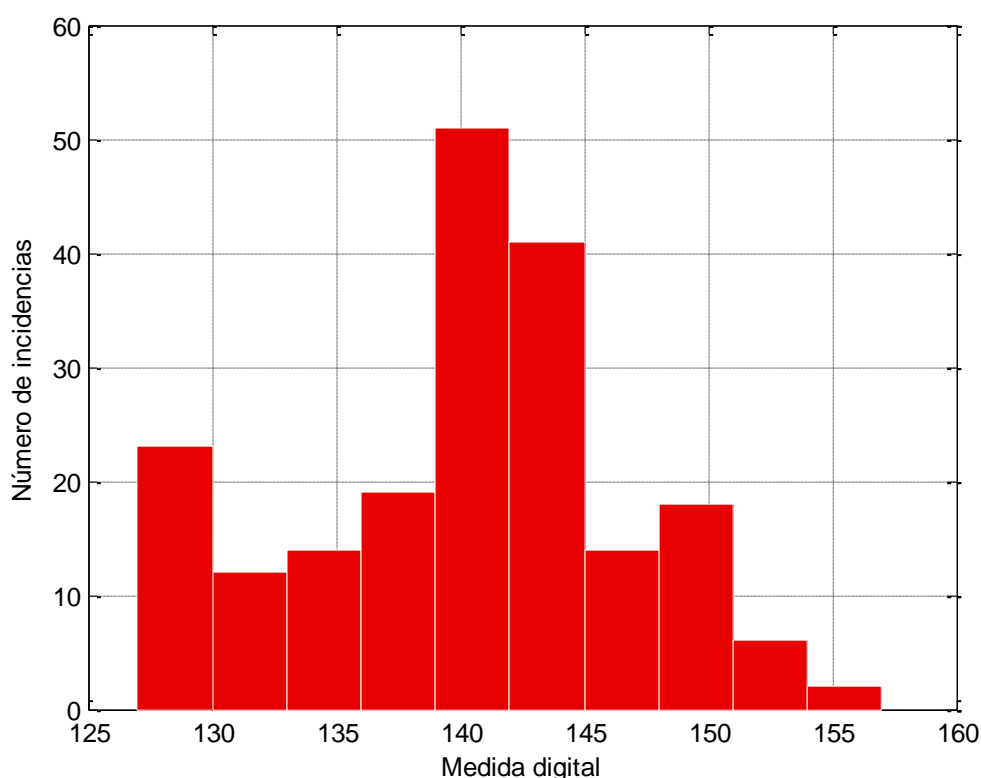


Figura 17. Histograma asociado al ángulo de flexión de 90°

En resumen, el análisis de los cuatro histogramas revela un comportamiento coherente y consistente del sensor Flex en respuesta a diferentes ángulos de flexión. Se observa una distribución significativa de valores en todos los casos, con una tendencia hacia la centralización de las mediciones en torno a uno o dos valores específicos. Aunque se evidencia una ligera variación en la distribución de los valores entre los diferentes ángulos de flexión, la capacidad del sensor para discriminar y detectar con precisión las distintas posiciones angulares se mantiene sólida en todas las condiciones evaluadas. Estos resultados respaldan la aplicación en un guante que solo necesite discriminar cuatro ángulos de flexión, los resultados obtenidos demuestran que el sensor proporciona una capacidad de detección más que adecuada.

Análisis de Errores de Medida

Es importante tener en cuenta que, a pesar de la consistencia observada en las mediciones, es necesario considerar el error asociado a las mismas. Con el fin de evaluar la precisión de las mediciones realizadas, se calculará la desviación estándar de los datos obtenidos y se determinará el error correspondiente.

En primer lugar, se calcula la desviación típica para cada grado de flexión, con los valores obtenidos. Ésta se calcula como:

$$\sigma = \sqrt{\frac{\sum(\bar{x} - x_i)^2}{N}} \quad (133)$$

Siendo N el número de muestras, \bar{x} la media aritmética y x_i cada valor.

A continuación, se calcula el error de medida, que se determina como el doble de la desviación típica, representando así un intervalo dentro del cual se espera que caigan la gran mayoría de las mediciones, con una confianza del 95%.

$$\varepsilon = \pm 2 \cdot \sigma \quad (144)$$

	0°	30°	60°	90°
σ	0.702	0.534	0.332	6.82
ε	1.4	1.07	0.66	13.64

Tabla 5. Valores de desviación típica y error para cada grado de flexión

Para una flexión de 0°, el valor de error indica que las mediciones pueden variar en hasta 1.4 unidades digitales alrededor del valor medio. Es importante considerar esta incertidumbre al interpretar los resultados experimentales.

Para la flexión de 30°, la menor desviación típica en comparación con 0° sugiere una mayor precisión en las mediciones realizadas a este ángulo de flexión, al igual que ocurre con la flexión de 60°.

Por último, para la máxima flexión, aunque el error es mayor en comparación con los ángulos más bajos, sigue siendo un valor significativo que debe tenerse en cuenta en el análisis de los datos obtenidos para 90°.

Estos valores de error proporcionan información crucial sobre la confiabilidad y precisión de las mediciones realizadas con el sensor de flexión, siendo fundamentales para la interpretación de los resultados experimentales y el diseño de sistemas que empleen este sensor.

Procesado de datos del sensor Flex

Al obtener medidas del sensor Flex, nos enfrentamos a la presencia de un considerable nivel de ruido en las mediciones realizadas. Este ruido introduce un margen de error mayor del deseado en los datos obtenidos, lo que afecta la precisión de las mediciones. Por ello, resulta fundamental aplicar un proceso de filtrado que reduzca este error y proporcione datos más exactos y confiables.

El procesamiento de los datos del sensor Flex se lleva a cabo mediante la aplicación de un filtro que reduce el ruido presente en las mediciones. Este filtro se implementa en código de programación en Arduino, y se basa en la siguiente fórmula:

$$\begin{aligned} X_{new} &= \alpha \cdot X_{actual} + (1 - \alpha) \cdot X_{old} \\ X_{old} &= X_{new} \end{aligned} \quad (15)$$

Donde:

- X_{new} es el nuevo valor procesado
- X_{actual} es el valor actual obtenido del sensor
- X_{old} es el valor del procesado anterior
- α es una constante de suavizado, con un valor de entre 0 y 1

En este caso, hemos determinado que un valor de $\alpha=0.9$ proporciona los mejores resultados después de experimentar con diferentes valores. Este coeficiente de suavizado equilibra la influencia del valor actual y el valor previo en la estimación del nuevo valor, ayudando así a reducir el ruido y mejorar la precisión de las mediciones.

Este proceso de filtrado es esencial para obtener mediciones más fiables y precisas del sensor Flex, permitiendo así una interpretación más exacta de los datos y una mejor capacidad para detectar variaciones significativas en la flexión del sensor.

Procesado de datos del sensor MPU6050

El sensor MPU6050, como se ha comentado en apartados anteriores, integra un acelerómetro y un giroscopio en un solo dispositivo, permitiendo la medición de la aceleración lineal y la velocidad angular en varios ejes. Aunque para este proyecto se ha seleccionado para medir la inclinación de la mano en dos ejes relevantes para su uso en un guante de datos.

El MPU6050 se presenta en un encapsulado compacto y fácil de integrar en dispositivos electrónicos, como se puede ver en la *Figura 18*. Para su comunicación con el microcontrolador, utiliza el protocolo de comunicación I2C, lo que facilita su conexión con una amplia gama de microcontroladores y dispositivos periféricos.



Figura 18. Sensor MPU6050

Según las especificaciones del fabricante, el sensor MPU6050 tiene un rango de medidas definido por el rango de operación del acelerómetro y el giroscopio. El acelerómetro puede medir aceleraciones en el rango de $\pm 2g$, $\pm 4g$, $\pm 8g$ o $\pm 16g$, dependiendo de la configuración seleccionada [16].

El principal inconveniente de obtener medidas tanto del sensor inercial como del sensor Flex es que en las mediciones realizadas se obtiene mucho ruido. De esta forma los datos obtenidos tienen un margen de error más grande de lo que debería ser. Por esto es necesario aplicar un filtro que reduzca el error y proporcione datos más exactos.

Existen varios filtros posibles para reducir el error en los datos siendo el más famoso el filtro de Kalman, desarrollado en 1960 por Rudolf E. Kalman. Es considerado como uno de los grandes descubrimientos del siglo XX por sus implicaciones en el proceso de filtrado de sensores (no solo en IMU's) y es uno de los artífices de la carrera espacial.

A grandes rasgos el filtro de Kalman realiza una estimación del valor futuro de la medición, y después compara el valor real mediante un análisis estadístico para compensar el error en futuras mediciones.

Sin embargo, el filtro de Kalman en su versión general implica la realización de cálculos complejos que suponen una implementación y tiempo de cálculo excesivo para Arduino. Por este motivo utilizaremos un filtro más sencillo denominado filtro complementario. En realidad, el filtro complementario puede considerarse una simplificación del filtro de Kalman que prescinde por completo del análisis estadístico.

El filtro complementario combina el ángulo calculado por el giroscopio y el ángulo calculado por el acelerómetro. La necesidad de combinar ambas lecturas surge debido a que, si trabajamos con el acelerómetro, este es susceptible a las aceleraciones producto del movimiento del MPU o a fuerzas externas, pero en tiempos largos el ángulo no acumula errores. A diferencia que si trabajamos solo con el giroscopio si bien este no es susceptible a fuerzas externas, con el tiempo el drift es muy grande y nos sirve solo para mediciones de tiempos cortos.

El filtro complementario se comporta como un filtro de paso alto para la medición del giroscopio y un filtro de paso bajo para la señal del acelerómetro. Es decir, la señal del giroscopio manda a corto plazo, y la del acelerómetro a medio y largo plazo, que es exactamente lo que queremos para compensar sus ventajas y defectos.

Existen varias formulaciones para un filtro complementario. En su expresión más sencilla, el filtro complementario puede expresarse como:

$$\theta = A \cdot (\theta_{\text{prev}} + \theta_{\text{gyro}}) + B \cdot \theta_{\text{accel}} \quad (16)$$

Donde A y B son dos constantes que, inicialmente, se toman como 0.8 y 0.2, respectivamente. Tras realizar pruebas y ajustar estas constantes para calibrar el filtro. Se ha determinado que los valores óptimos son 0.95 para A y 0.05 para B. Es importante señalar que, para una calibración adecuada, los valores de A y B deben sumar 1.

En la *Figura 19* se muestra la diferencia entre las mediciones obtenidas sin aplicar el filtro (izquierda) y con el filtro complementario (derecha). La señal medida presenta significativamente menos ruido, mostrando una gráfica más estabilizada y con error reducido.

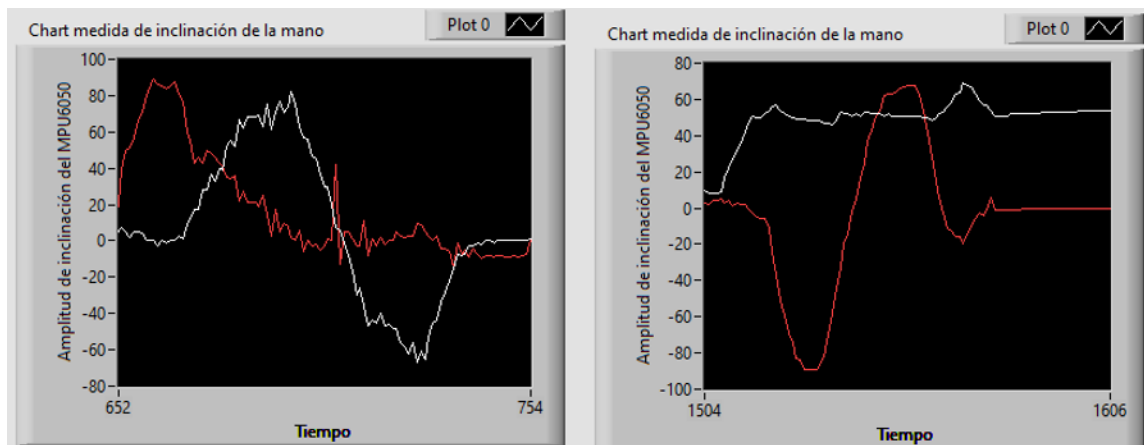


Figura 19. Gráfica de inclinación sin filtro frente a con filtro complementario

Por otro lado, se puede hablar del procesador DMP (Digital Monitor Processor) que permite calcular a tiempo real los ángulos del dispositivo. El acceso a los registros del DMP es algo complejo, debido a que el fabricante no ha abierto toda la información al respecto. Lo interesante del DMP es que implementa filtros que, de otro modo, tendrían que ser implementados en el software de nuestro Arduino, incluyendo los que dan como resultado los ángulos propios del dispositivo. Hay que destacar que en el presente trabajo solo se ha utilizado dicho filtro, pero no se ha configurado por el motivo de complejidad mencionado.

En definitiva, además de los filtros internos del DMP, el filtro complementario es una alternativa al filtro de Kalman lo suficientemente precisa como para que nos proporcione valores con bajo error, por lo que será el elegido sin duda alguna.

Desarrollo Software

Lectura de los valores de los sensores

Para entender cómo se ha realizado la lectura de datos, es necesario saber algunos conceptos básicos sobre la programación en Arduino [17].

Un proyecto en Arduino se denomina *sketch* y tiene una extensión *.ino*. Este sketch debe tener, como mínimo, dos funciones principales:

- **void setup():** esta función se ejecuta una sola vez cuando se inicia el programa. En ella se realizan configuraciones como la inicialización de librerías o la configuración de un pin, de manera que se quede inicializado así para el resto del programa.
- **void loop():** es la función que se llamará continuamente y es donde se realizará el funcionamiento del programa.

Ambas funciones deben existir en el sketch para el correcto funcionamiento del programa.

El código implementado comienza con la inicialización de las variables y librerías necesarias. A continuación, se define una función denominada `updateFiltered()`, encargada de calcular la inclinación a partir de los datos proporcionados por el acelerómetro y el giroscopio del sensor MPU6050.

Después, en la función `setup()`, se realizan las configuraciones iniciales, como la inicialización del puerto serie que se utilizará para la comunicación de datos.

Finalmente, en la función `loop()`, se encuentra la mayor parte del código. En este punto, se adquieren los datos de los sensores a través de los pines correspondientes, se procesan los datos del sensor Flex según lo discutido anteriormente y se llama a la función `updateFiltered()`. Una vez obtenidos los tres datos requeridos, se envían por el puerto serie utilizando los comandos `Serial.print()` y `Serial.println()`, lo que permite que sean recibidos y procesados por LabVIEW en etapas posteriores del proyecto.

Interfaz de LabVIEW

El software desempeña un papel crucial en el funcionamiento del proyecto, ya que proporciona la interfaz de usuario necesaria para interactuar con los dispositivos hardware y visualizar los datos capturados. En este apartado, se describirá la implementación de la interfaz de usuario utilizando LabVIEW, un entorno de programación gráfica ampliamente utilizado en aplicaciones de control, adquisición de datos y sistemas embebidos.

LabVIEW, desarrollado por National Instruments, ofrece una interfaz intuitiva y visual que permite crear aplicaciones complejas sin necesidad de tener experiencia en programación convencional. Utilizando un enfoque de programación gráfica basado en bloques, se pueden diseñar sistemas completos mediante la conexión de elementos gráficos llamados "iconos" o "VI" (Virtual Instruments), que representan funciones y operaciones.

Además de su facilidad de uso, LabVIEW ofrece una amplia variedad de herramientas y funciones para el análisis de datos, la visualización de resultados y la comunicación con dispositivos hardware. Estas características hacen de LabVIEW una opción ideal para el desarrollo de esta aplicación.

En este apartado, se proporcionará una visión general del diseño y la implementación de la interfaz de usuario en LabVIEW, destacando las características clave del entorno de programación y su contribución al éxito del proyecto.

Descripción de la interfaz de usuario

Aunque la funcionalidad del interfaz de usuario se encuentra descrita con detalle en el anexo correspondiente de este documento (Operativa de usuario), en este apartado se muestra una breve descripción de su funcionalidad para contextualizar los tipos de terminales usados y las acciones que sobre ellos hace el programa para facilitar el uso de la aplicación.

En la *Figura 20* se puede ver la interfaz de usuario o, lo que es lo mismo, lo que en LabVIEW se denomina como panel frontal.

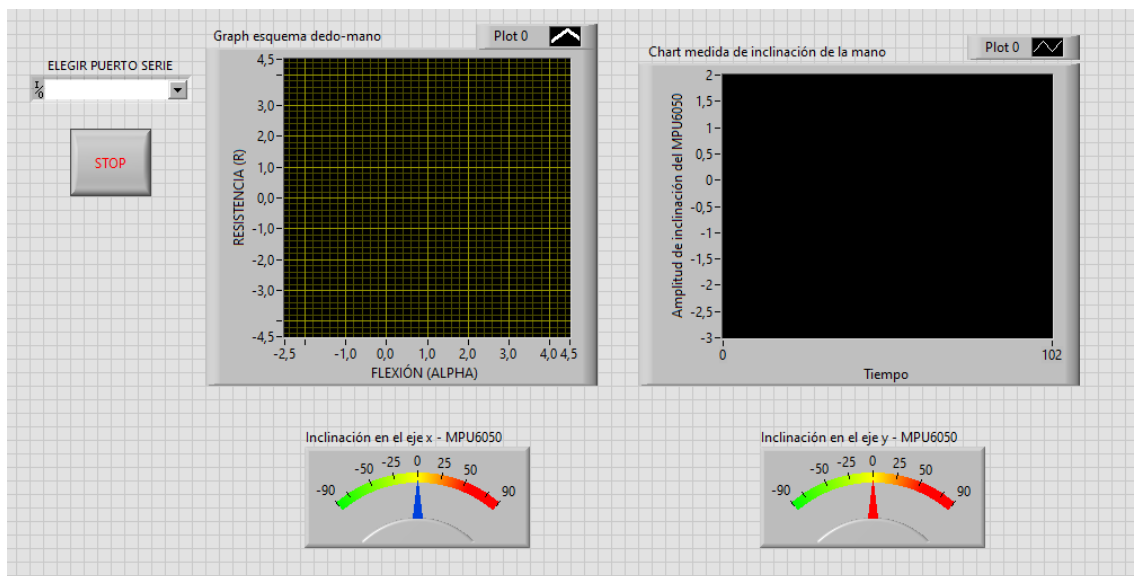


Figura 20. Interfaz de usuario (Panel frontal)

La interfaz de usuario en LabVIEW cuenta con varios elementos clave para la interacción y visualización de datos:

- Controlador de Puerto Serie (COM):

Este componente, representado como un desplegable, permite al usuario seleccionar el puerto COM al que está conectado el Arduino UNO. Por ejemplo, cuando el Arduino está conectado a través de USB, el puerto COM asignado, como COM3, se mostrará en la lista desplegable. Es esencial seleccionar el puerto adecuado antes de inicializar el programa para garantizar la comunicación correcta con los sensores.

- Botón de Stop:

Este botón permite al usuario detener el programa en cualquier momento. Al hacer clic en él, el programa dejará de recibir y mostrar datos, pausando su ejecución en el punto actual.

- Indicadores de inclinación:

Estos indicadores muestran los datos recibidos por los sensores. Los dos datos mostrados son la inclinación en los ejes X e Y, proporcionada por el sensor MPU6050. Cada indicador tiene una escala que se ajusta en función del rango de valores máximos y mínimos que pueden proporcionar los sensores.

- Gráfica "Graph XY":

Esta gráfica representa un esquema dedo-mano en el que se muestran dos segmentos unidos. La flexión del sensor Flex o la inclinación del MPU6050 determinan la posición de los segmentos en tiempo real, simulando el movimiento de una mano flexionando un dedo o inclinándose.

- Gráfica "Waveform Chart":

Esta gráfica es distinta ya que tiene un buffer que almacena los datos recibidos, lo que permite mostrar las dos inclinaciones a lo largo del tiempo, permitiendo ver el recorrido realizado por los datos. Es especialmente útil para visualizar cambios en las inclinaciones de manera dinámica y continua.

Estos elementos dotan a la interfaz de una experiencia interactiva y visualmente informativa, optimizada para el monitoreo y análisis de los datos provenientes de los sensores. Todos estos componentes se ilustran en la Figura 21.

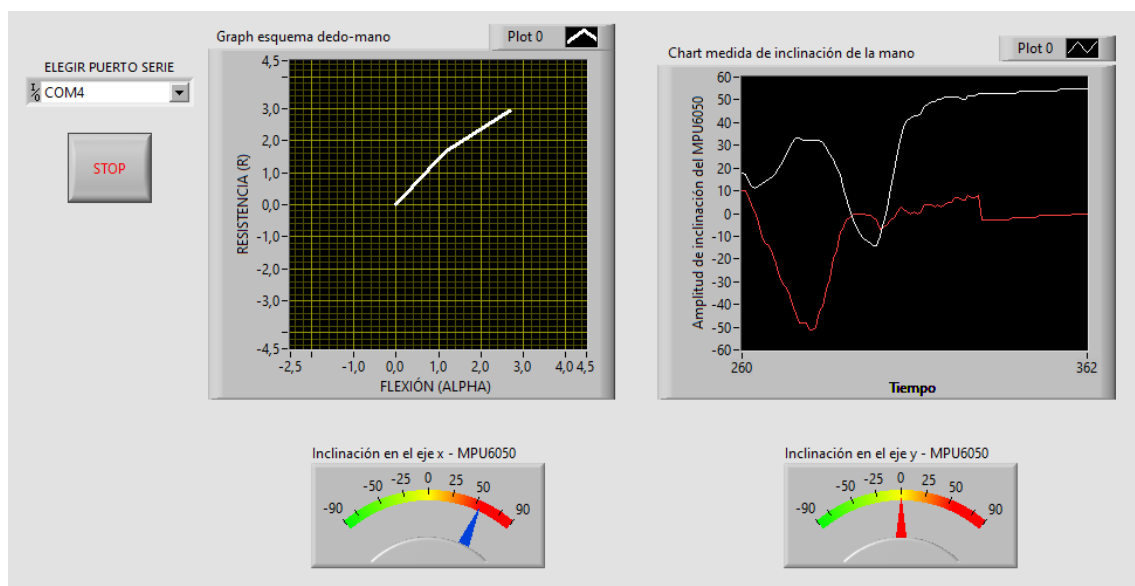


Figura 21. Interfaz de usuario en funcionamiento

CONCLUSIONES Y LÍNEAS FUTURAS

Conclusiones

Una vez finalizada la aplicación, podemos afirmar que hemos cumplido con los objetivos propuestos al inicio de nuestro desarrollo. Se ha llevado a cabo una selección y caracterización exhaustiva de los sensores Flex y MPU6050, mediante un proceso de filtrado y caracterización, para su potencial aplicación en un guante de captura de datos.

Dado que nuestra aplicación tiene un enfoque docente, la mayoría de los requisitos del interfaz de usuario fueron dictados por los tutores. La verificación del cumplimiento de todos los requisitos mínimos planteados se realizó mediante un detallado plan de pruebas.

La elección de LabVIEW como entorno de desarrollo fue motivada principalmente por el aprendizaje obtenido en la asignatura de Sistemas de Adquisición de Datos. Los resultados de este proyecto destacan la capacidad del lenguaje de programación gráfica para implementar sistemas inteligentes capaces de adquirir múltiples señales simultáneamente y en tiempo real. Este trabajo ha servido para mejorar e iniciarse en aspectos que serán de gran utilidad en una vida profesional.

Líneas futuras

Durante todo el desarrollo de este proyecto, hemos priorizado la capacidad de ampliación de la aplicación, con el objetivo de facilitar la integración de nuevas funcionalidades y adaptaciones para futuras líneas de trabajo.

Inicialmente, se propuso diseñar una aplicación que pudiera ser implementada en un guante de captura de datos. Los sensores seleccionados fueron cuidadosamente evaluados en función de su capacidad para discriminar diferentes grados de inclinación, tanto en los dedos como en la mano. Hemos

logrado discriminar al menos cuatro posiciones distintas de los dedos, lo que representa un avance significativo en la utilidad potencial de un guante de datos.

Además, el trabajo se ha diseñado con la flexibilidad necesaria para adaptarse a configuraciones más complejas. Por ejemplo, el sistema puede soportar la implementación de un guante equipado con cinco sensores Flex y un MPU6050. Gracias a la capacidad de recibir datos de manera simultánea y en tiempo real, y a la amplia disponibilidad de entradas digitales en el microcontrolador, esta configuración es completamente factible. Además, una posible elección de una placa Arduino Nano ofrece una mayor movilidad a la mano ya que se trata de un microcontrolador de menor tamaño, lo que amplía aún más las posibilidades de aplicación.

En cuanto a la alimentación, aunque en este caso se opta por alimentar el sistema a través de un ordenador conectado a la placa Arduino UNO, esta solución puede ser fácilmente adaptada para utilizar baterías pequeñas. Con la memoria disponible en la placa, no sería necesario mantener una conexión constante con un ordenador, lo que resultaría en un sistema más compacto y portátil.

Una línea futura de desarrollo que consideramos altamente prometedora es la implementación de conectividad Bluetooth. Esto permitiría una mayor movilidad que, junto con la alimentación a través de baterías, elimina la necesidad de cables conectados al ordenador, lo que abriría nuevas posibilidades de uso en entornos móviles o de campo.

En conclusión, este proyecto podría tener aplicaciones muy relevantes en medicina, como la creación de un guante que capture datos de pacientes durante la rehabilitación post-ictus, tal como se mencionaba en la introducción. Además, podría ser utilizado para desarrollar una aplicación que detecte el lenguaje de signos y lo traduzca en texto en pantalla, facilitando así el aprendizaje de este lenguaje a personas que lo necesiten. Estas son solo algunas de las muchas posibilidades que este proyecto ofrece para el futuro.

Bibliografía

- [1] M. K. I. y. A. A. A. Chang Yi Neng, «Flex Force Smart Glove for Therapy Treatment Using Arduino and Raspberry Pi,» de *Lecture Notes in Mechanical Engineering (LNME)*, 2021.
- [2] Escuela Politécnica Superior de Jaén, «Normativa sobre Trabajos Fin de Grado en la Escuela Politécnica Superior de Jaén,» 2017. [En línea]. Available: https://eps.ujaen.es/sites/centro_epsj/files/uploads/documents/normativa/Normativa_TFG_TFM_EPSJ_aprobada%20Junta%20de%20Escuela%2013%20sept%202017.pdf. [Último acceso: 1 mayo 2024].
- [3] Lidia Santos, José L. González, Javier P. Turiel, Juan C. Fraile, Eusebio de la Fuente., «GUANTE DE DATOS SENSORIZADO PARA USO EN CIRUGÍA LAPAROSCÓPICA ASISTIDA POR LA MANO (HALS),» de *s XXXVI Jornadas de Automática*, Bilbao, 2015.
- [4] R. A. Española, «RAE,» Asociación de Academias de la Lengua Española, [En línea]. Available: <https://dle.rae.es/sensor>. [Último acceso: 1 mayo 2024].
- [5] J. M. Infante, «Rambal Automatización y Robótica,» [En línea]. Available: <https://rambal.com/presion-peso-nivel-flex/250-sensor-flex.html>. [Último acceso: 1 mayo 2024].
- [6] «Sparkfun,» Electrónica Sparkfun, [En línea]. Available: https://learn.sparkfun.com/tutorials/flex-sensor-hookup-guide?_ga=2.152086457.186179385.1714584445-1420149787.1670871523. [Último acceso: 2024 mayo 1].
- [7] «5 Hertz Electrónica,» 5 Hertz, [En línea]. Available: https://www.5hertz.com/index.php?route=tutoriales/tutorial&category_id=1&tutorial_id=2#11. [Último acceso: 1 mayo 2024].
- [8] «Universidad de Sevilla,» [En línea]. Available: https://biblus.us.es/bibing/proyectos/abreproy/11638/descargar_fichero/Capitulo+4.pdf. [Último acceso: 1 mayo 2024].
- [9] «5 Hertz Electrónica,» 5 Hertz, [En línea]. Available: https://www.5hertz.com/index.php?route=tutoriales/tutorial&tutorial_id=13. [Último acceso: 1 mayo 2024].
- [10] «SBG Systems,» SBG Systems, [En línea]. Available: <https://www.sbg-systems.com/es/unidad-de-medicion-inercial-sensor-imu/>. [Último acceso: 1 mayo 2024].
- [11] «AliExpress,» PUMUDDSY, [En línea]. Available: https://es.aliexpress.com/item/32944661306.html?spm=a2g0o.productlist.main.9.1f6bwf9Lwf9Lnu&algo_pvid=5cc8b221-23eb-434f-9f77-e6d7f7536a0c&aem_p4p_detail=202405011115278558355687697680014248188&algo_exp_id=5cc8b221-23eb-434f-9f77-e6d7f7536a0c-4&pdp_npi=4%. [Último acceso: 1 mayo 2024].
- [12] D. S. Magaz, Adquisición de datos desde un sensor inercial, Leganés: Escuela Politécnica Superior UC3M, 2018.

- [13] Junseung Bang, Jinho You, Youngho Lee, «A Prototype of Flex Sensor Based Data Gloves to Track the Movements of Fingers,» *Smart Media Journal*, diciembre 2019.
- [14] «Arduino,» Arduino S.r.l., [En línea]. Available: <https://store.arduino.cc/products/arduino-uno-rev3>. [Último acceso: 2 mayo 2024].
- [15] Microchip, «Atmel Corporation,» [En línea]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf. [Último acceso: 2 mayo 2024].
- [16] «Invensense,» [En línea]. Available: <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>. [Último acceso: 2 mayo 2024].
- [17] «Arduino,» MCI Electronics, [En línea]. Available: <https://arduino.cl/programa-arduino/>. [Último acceso: 2 mayo 2024].
- [18] SpectraSymbol, «Sparkfun,» [En línea]. Available: <https://www.sparkfun.com/datasheets/Sensors/Flex/flex22.pdf>. [Último acceso: 2024 mayo 3].
- [19] «Sparkfun,» [En línea]. Available: <https://learn.sparkfun.com/tutorials/flex-sensor-hookup-guide>. [Último acceso: 2024 mayo 3].

ANEXO

Operativa de usuario

La interfaz de usuario proporciona una forma intuitiva de visualizar y monitorizar los datos obtenidos por los sensores conectados al microcontrolador Arduino. Esta guía proporcionará, además de una explicación detallada de cómo utilizar la interfaz para acceder y comprender los datos de manera efectiva, algunos requisitos del hardware.

Requisitos del hardware

Los requisitos del hardware deben ser considerados cuidadosamente antes de utilizar el sensor Flex. Es crucial evitar la flexión excesiva en la base del sensor, limitándola únicamente a su superficie funcional para prevenir posibles daños. Asimismo, es fundamental flexionar el sensor únicamente por su cara funcional, identificable por ser la cara que tiene la tinta conductora. En la Figura 22 se muestra la forma correcta de flexión, proporcionada por el fabricante [18].

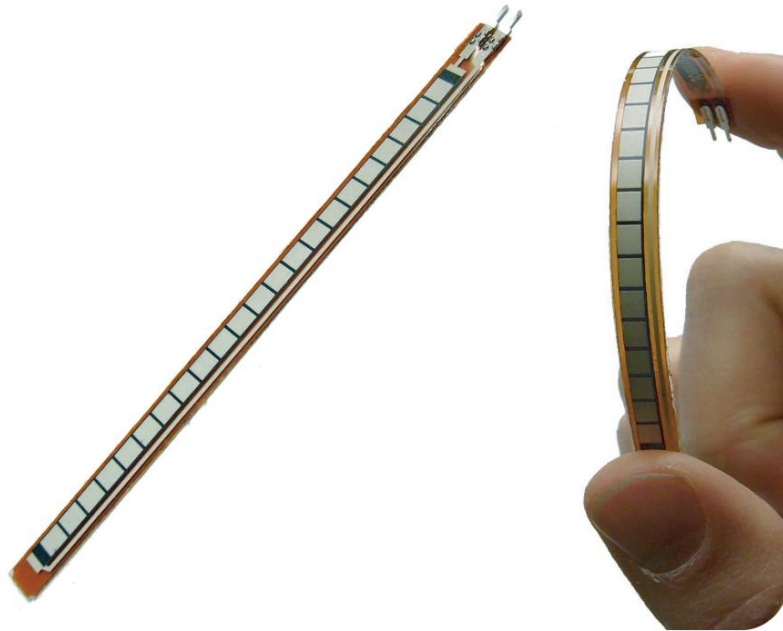


Figura 22. Flexión por la cara correcta del sensor

Si se desea conectar el sensor mediante soldadura de pestañas, es esencial tener en cuenta que tanto el sustrato flexible como el material semiconductor son altamente sensibles al calor. Las resistencias sensibles a la fuerza de Interlink son especialmente susceptibles en comparación con otros sensores flexibles, aumentando el riesgo de dañar el sensor durante el proceso de soldadura. Se recomienda esta técnica únicamente para usuarios avanzados que hayan ajustado su soldador a temperaturas más bajas. Una solución óptima es la utilizada en el presente proyecto, a través del uso de un Conector remachador Amfenol CFI Clincher. A continuación, se muestra la forma correcta de conexión de dicho conector [19].

Requisitos de software

Antes de utilizar la interfaz, hay que asegurarse de tener instalado el software LabVIEW en el sistema. Además, es necesario que el microcontrolador Arduino esté correctamente conectado y enviando datos a través de un puerto serie.

Componentes de la Interfaz

Para utilizar correctamente la interfaz de LabVIEW, es fundamental comprender sus componentes y sus funciones dentro del programa. Estos componentes se dividen en cuatro categorías principales: desplegable de selección del puerto serie, botón de parada, indicadores de datos de inclinación y gráficas de los datos obtenidos.

- Selección del puerto serie:

Se utiliza el menú desplegable para elegir el puerto serie al que está conectado el Arduino. Esto es necesario para asegurar la comunicación adecuada entre la interfaz y el dispositivo y debe ser el primer paso, ANTES de inicializar el programa.

- Botón de parada:

El botón de parada permite detener la recepción de datos en cualquier momento. Al hacer clic en este botón, el programa se detendrá y los últimos datos recibidos se mantendrán visibles en todos los elementos de la interfaz.

- Indicadores de inclinación:

Hay dos indicadores de inclinación en la interfaz, vistos de izquierda a derecha, muestran la inclinación en los ejes X e Y, respectivamente. La escala de los indicadores tiene un rango de -90° a 90° .

- Gráfica del esquema dedo-mano:

Esta gráfica proporciona una representación visual de la inclinación de la mano y la flexión del dedo mediante dos segmentos conectados. El primer segmento, que representa la inclinación de la mano, está anclado en el extremo izquierdo en el punto (0, 0) y tiene una longitud de 2 cm en la gráfica. Su posición se ajusta verticalmente según los valores de inclinación detectados en el eje Y. Por ejemplo, si la mano se inclina verticalmente hacia arriba, el segmento se moverá para formar una línea vertical.

El segundo segmento, que refleja la flexión del dedo, está constantemente conectado al extremo derecho del primer segmento. Inicialmente, este segmento permanece horizontal cuando la flexión es de 0° y se inclina hacia abajo a medida que el ángulo de flexión aumenta. Su inclinación máxima alcanza los 90° , formando una perpendicular con respecto al primer segmento.

Para facilitar la comprensión del funcionamiento y la interpretación de la gráfica, se presentan algunos ejemplos con varios grados de flexión.

Ejemplo 1: cuando el dedo se encuentra flexionado y la mano se encuentra en posición horizontal, la visualización sería como en la Figura 23.

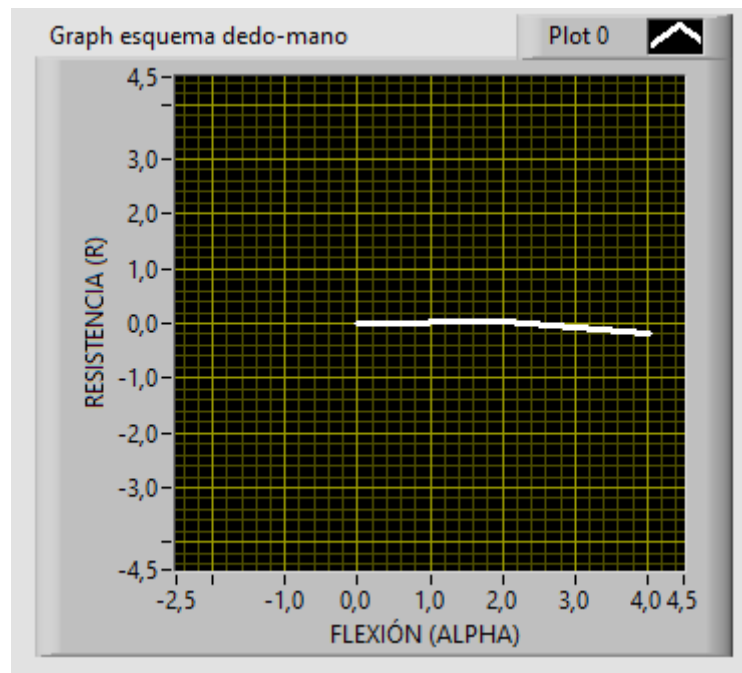


Figura 23. Gráfica de ejemplo 1

Ejemplo 2: cuando el dedo se flexiona a 90° y la mano se encuentra horizontal, se vería como en la Figura 24.

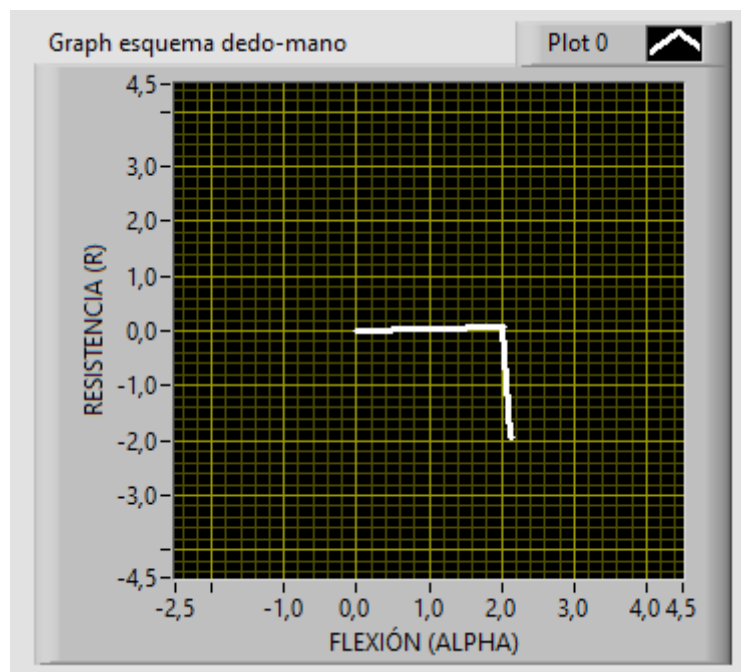


Figura 24. Gráfica de ejemplo 2

Ejemplo 3: cuando la mano se encuentra inclinada al máximo hacia arriba y el dedo se flexiona unos 45° , se observa en la Figura 25.

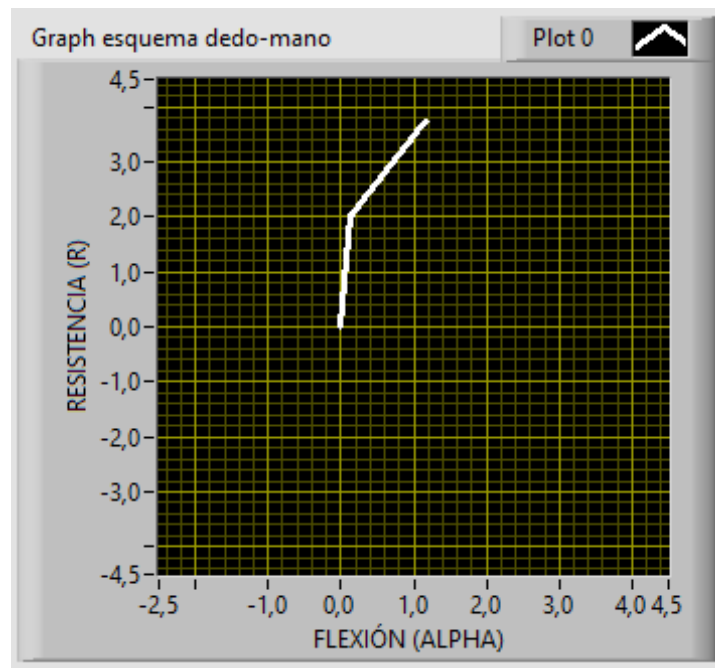


Figura 25. Gráfica de ejemplo 3

- Gráfica de medida de inclinación de la mano en dos ejes:

Muestra los datos de inclinación en función del tiempo para ambos ejes del sensor inercial (X en blanco e Y en rojo). Los datos se van almacenando en un buffer interno de este tipo de gráfica y se superponen en la gráfica para visualizar tanto los datos más recientes como los anteriores. Los ejes de la gráfica se ajustan automáticamente para adaptarse a los valores máximos y mínimos de los datos mostrados. Sin embargo, si se prefiere, los ejes también pueden fijarse manualmente para una mejor visualización. En la Figura 26 se muestra la gráfica con distintos movimientos de la mano en ambas direcciones.

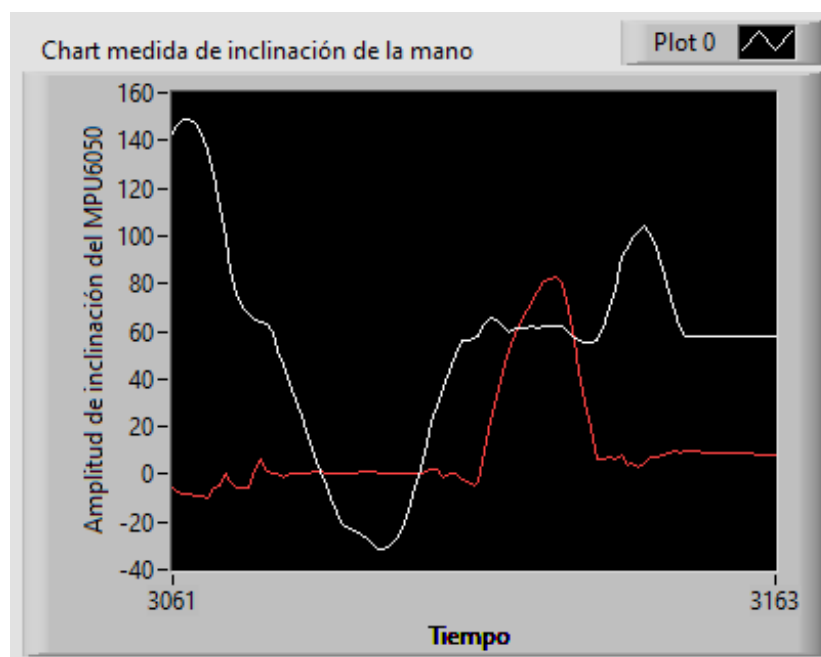


Figura 26. Gráfica de valores de inclinación de la mano

Diagrama de bloques

Para crear la interfaz y visualizarla en el Front Panel, se ha desarrollado previamente un diagrama utilizando programación gráfica o en bloques. Debido a su amplitud, este diagrama se ha dividido en dos imágenes separadas para una mejor visualización.

En la primera parte, representada en la Figura 27, se lleva a cabo la adquisición de datos desde el puerto serie. El diagrama lee todos los datos transmitidos, los cuales están separados por ";", en formato string, y los almacena en un array. Dado que se espera recibir tres datos, se procede a separarlos y mostrar sus valores en los indicadores correspondientes.

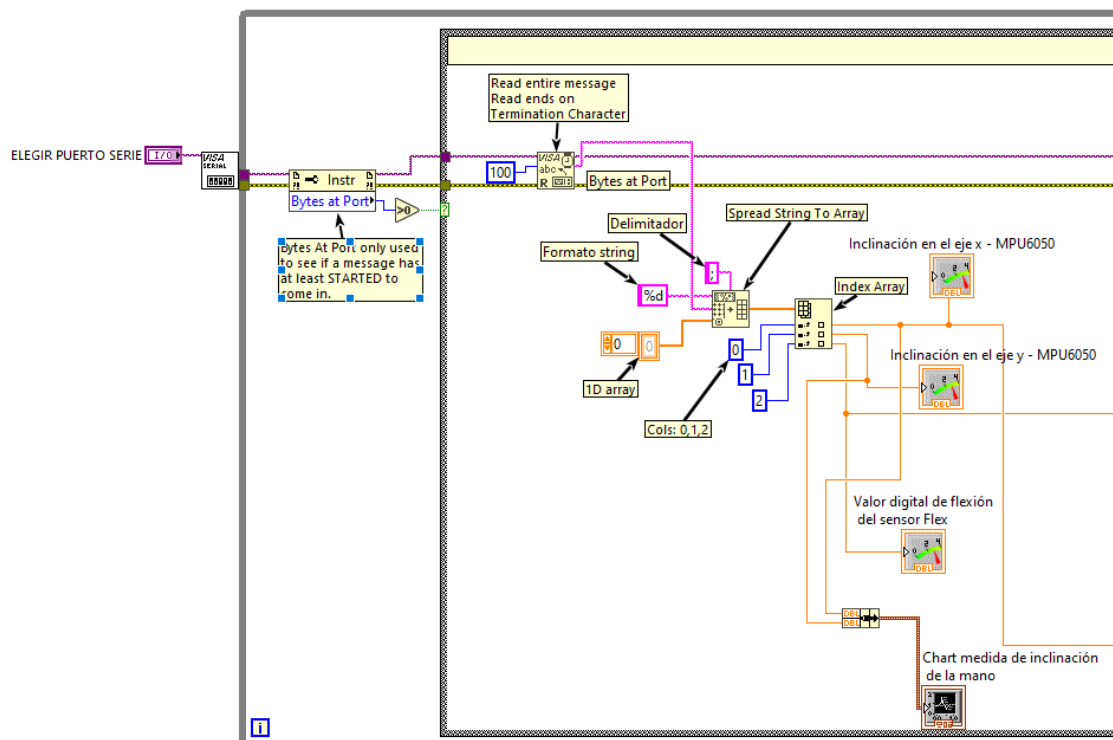


Figura 27. Primera parte del diagrama de bloques

En la segunda parte del diagrama, se realiza el procesamiento de los datos para su visualización en la gráfica del esquema dedo-mano. Esta sección se encarga de interpolar los valores digitales para obtener datos en escala de grados, y luego convierte estos valores de los sensores Flex, junto con las inclinaciones en los ejes X e Y, a radianes. Posteriormente, se realizan las operaciones necesarias para calcular simultáneamente la posición del segmento correspondiente en función de los tres datos recibidos. Finalmente, estos valores se almacenan en dos arrays separados para cada eje de la gráfica y se muestran en la misma. Todo este proceso se puede observar en la Figura 28.

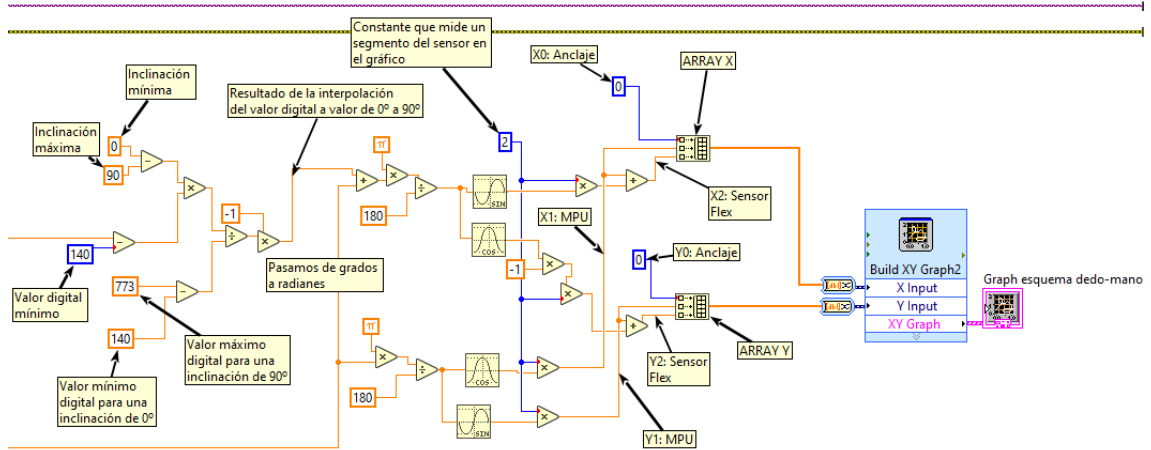


Figura 28. Segunda parte del diagrama de bloques