



COMUNICACIONES

SECRETARÍA DE INFRAESTRUCTURA, COMUNICACIONES Y TRANSPORTES



Desarrollo de unidad de medición inercial para pruebas de dinámica vehicular

Mauricio Eliseo Cruz Acevedo

Marco Antonio Hernández Nochebuena

David Vázquez Vega

Oscar Flores Centeno

José Ricardo Hernández Jiménez

Publicación Técnica No. 815
Querétaro, México
2024

ISSN 0188-7297

Esta investigación fue realizada en la Coordinación de Ingeniería Vehicular e Integridad Estructural (CIVIE) del Instituto Mexicano del Transporte, por MTA. Mauricio Eliseo Cruz Acevedo, MC. David Vázquez Vega, MC. José Ricardo Hernández Jiménez y el MC. Marco Antonio Hernández Nochebuena.

Esta investigación es el producto final del proyecto de investigación interna EI 23/23 “Implementación de unidad inercial para mediciones de variables dinámicas en vehículos de carretera”.

Se agradece la colaboración del Dr. Francisco Javier Carrión Viramontes coordinador de la CIVIE en la revisión de la presente publicación, así como a la colaboración del C. Bryan Alexis Jiménez Aguilar, estudiante de la Universidad Politécnica de Chiapas.

Las opiniones expresadas en esta publicación son de los autores (as) y no necesariamente reflejan los puntos de vista del Instituto Mexicano del Transporte.

Tabla de Contenido

	Página
Sinopsis.....	v
Abstract.....	vii
Introducción.....	1
1. Antecedentes.....	3
1.1 Aceleración y velocidad angular.....	5
1.1.1 Aceleración longitudinal.....	6
1.1.2 Aceleración lateral.....	7
1.1.3 Aceleración vertical.....	8
1.2 Sistemas de medición de aceleración.....	9
1.2.1 Velocidad angular.....	12
1.3 Sistemas de medición de velocidad angular.....	14
2. Marco teórico.....	17
2.1 Módulo ESP32.....	17
2.2 Sensor MPU-6050.....	19
2.3 Módulo lector de memoria microSD.....	21
2.4 Comunicación SPI.....	22
2.5 Interfaz I2C.....	23
3. Metodología.....	25
3.1 Prototipo IMU.....	26
3.1.1 Definición de requerimientos.....	26
3.1.2 Materiales utilizados.....	26

3.1.3	Herramientas y material de construcción	27
3.1.4	Construcción de IMU	27
3.2	Validación del prototipo	28
3.2.1	Prueba modular.....	28
3.2.2	Prueba de integración.....	32
3.2.3	Prueba de aceptación.....	35
4.	Resultados.....	37
4.1.1	Resultados de aceleración en eje X.....	38
4.1.2	Resultados de aceleración en eje Y.....	38
4.1.3	Resultados de aceleración en eje Z.....	39
4.1.4	Resultados de velocidades angulares.....	39
	Conclusiones y trabajo futuro.....	43
	Bibliografía	45
	Anexo 1 Códigos de pruebas modulares.....	47

Sinopsis

En las industrias automotriz, aeroespacial y marítima, la medición precisa de las variables cinemáticas del movimiento, como la aceleración y la velocidad, es esencial para evaluar el comportamiento de los vehículos y comprender cómo responden ante las fuerzas aplicadas, como son la aceleración, el frenado y los cambios de dirección; esto es crucial para mejorar la seguridad y la eficiencia. El presente estudio aborda el desarrollo de un prototipo de unidad inercial que mida y registre tanto las velocidades angulares como las aceleraciones lineales, con el fin de identificar el comportamiento dinámico y estabilidad de vehículos de carretera al estar en movimiento.

El prototipo de unidad inercial desarrollado en este trabajo se realizó con un sensor MPU-6050 y una placa de adquisición ESP32, el dispositivo mostró un correcto funcionamiento para medir las variables de aceleración lineal y velocidad angular. Los errores obtenidos para la aceleración están por debajo del 10%; para el caso de la velocidad angular, se obtuvieron mediciones y registros que demostraron la capacidad del dispositivo para adquirir estas variables, aunque no fue posible evaluar su grado de precisión.

Abstract

In the automotive, aerospace and maritime industries, the accurate measurement of kinematic motion variables, such as acceleration and velocity, is essential to evaluate the behavior of vehicles and understand how they respond to applied forces such as acceleration, braking and directional changes; this is crucial to improve safety and efficiency. This study addresses the development of an inertial measurement unit prototype that measures and records both angular velocities and linear accelerations to identify the dynamic behavior and stability of road vehicles while in motion.

The inertial unit prototype developed in this work was based on a MPU-6050 sensor and an ESP32 acquisition board, the device showed a correct operation to measure the linear acceleration and angular velocity variables. The errors obtained for acceleration are below 10%; for the case of angular velocity, measurements and records were obtained that demonstrated the capability of the device to acquire these variables, although it was not possible to assess its degree of accuracy.

Introducción

La medición de la aceleración es esencial en numerosos campos y aplicaciones, desde la ingeniería y la medicina hasta la investigación científica, en donde es necesario contar con información crítica para comprender el movimiento y, de ahí, mejorar el diseño de los sistemas y de los dispositivos. En términos de los vehículos de carretera, el contar con la capacidad de medir la aceleración y la velocidad en tiempo real, nos permite monitorear el desempeño de los vehículos y/o conductas de los operadores y así, poder identificar situaciones potenciales de riesgo, como son el exceso de velocidad, el frenado brusco o la aceleración rápida, lo que puede ayudar a prevenir accidentes y mejorar la seguridad vial.

Las unidades de medición inercial (IMU, por sus siglas en inglés) son capaces de proporcionar mediciones precisas de aceleración lineal, aceleración angular y orientación espacial del vehículo en tiempo real. Esto permite una comprensión detallada del movimiento del vehículo, incluyendo su aceleración, deceleración, cambios de dirección y estabilidad dinámica. Con datos precisos de las IMU, los ingenieros pueden realizar un análisis exhaustivo de la dinámica del vehículo, incluyendo la evaluación de su comportamiento en curvas, su respuesta a maniobras de emergencia y su estabilidad en diferentes condiciones de conducción. Los datos de las IMU también son utilizados por organismos reguladores y fabricantes de vehículos para evaluar la resistencia estructural, la estabilidad y la capacidad de maniobra del vehículo en diferentes escenarios de conducción.

Las IMU son fundamentales para el desarrollo y la validación de sistemas avanzados de asistencia al conductor, como el control de estabilidad, el control de tracción, el control de crucero adaptativo y los sistemas de frenado automático de emergencia. Estos sistemas utilizan datos de las IMU para detectar situaciones de riesgo y tomar medidas correctivas para mejorar la seguridad y la estabilidad del vehículo. Las IMU también desempeñan un papel fundamental en la investigación y el desarrollo de vehículos autónomos, proporcionando datos precisos sobre la posición, la velocidad y la orientación del vehículo en tiempo real. Estos datos son utilizados por los sistemas de navegación y control para tomar decisiones en tiempo real y garantizar una conducción segura y eficiente en entornos complejos y variables.

El presente trabajo es una propuesta de un prototipo de IMU, el cual se utilizará como complemento de instrumentación para pruebas de evaluación de comportamiento dinámico en vehículos de carretera. Con este trabajo se busca reducir la dependencia tecnológica, ya que los instrumentos suministrados por otros fabricantes extranjeros resultan costosos, no solo por la adquisición de los instrumentos, sino también, por el soporte técnico requerido y el mantenimiento que se pudiera requerir para conservarlos en buen estado de operación. Para fundamentar y comprender mejor el aporte de este trabajo, en el capítulo 1 se describen los antecedentes de los sistemas de medición de aceleración lineal y velocidad angular; en el capítulo 2 se describe la parte técnica de los componentes principales utilizados para la construcción de la unidad inercial, así como el protocolo de comunicación utilizado. Posteriormente, en el capítulo 3 se describe la metodología utilizada para el desarrollo de la IMU y el protocolo de pruebas sugerido para su evaluación; en el capítulo 4 se presentan los resultados de la evaluación del prototipo, concluyendo que este cumple con los requerimientos iniciales de diseño, correspondientes a su capacidad de medición y registro tanto de la aceleración lineal como de la velocidad angular. Finalmente, se exponen las conclusiones y se sugieren propuestas de mejora para el desarrollo tecnológico.

1. Antecedentes

El concepto de aceleración tiene antecedentes históricos que se remontan a la antigüedad, aunque su formalización matemática se desarrolló principalmente durante la Revolución Científica en los siglos XVI y XVII.

En la antigua Grecia, Aristóteles planteó ideas sobre el movimiento de los objetos. Propuso que los objetos en movimiento tendían a detenerse porque, según él, necesitaban una fuerza para mantenerse en movimiento. Sin embargo, su explicación del movimiento era limitada y no incluía el concepto moderno de aceleración. Por otro lado, Galileo realizó numerosos experimentos sobre el movimiento, incluidos los estudios sobre la caída libre de los cuerpos. Descubrió que todos los objetos, independientemente de su masa, caen hacia la Tierra con la misma aceleración constante, bajo la influencia exclusiva de la gravedad. Sus experimentos y observaciones sentaron las bases para el concepto moderno de aceleración. Fue hasta Newton cuando se formalizó las leyes del movimiento en su obra "*Philosophiæ Naturalis Principia Mathematica*" (Principia), publicada en 1687. En estas leyes, describió la relación entre la fuerza aplicada a un objeto y su cambio resultante en velocidad. La Segunda Ley de Newton establece que la aceleración de un objeto es directamente proporcional a la sumatoria de todas las fuerzas que actúa sobre él e inversamente proporcional a su masa ($a = \sum F/m$). Esta formulación matemática es fundamental para comprender el concepto moderno de aceleración [1, 2, 3].

La evolución de los sistemas para medir la aceleración ha experimentado un desarrollo significativo a lo largo del tiempo, desde métodos simples hasta tecnologías avanzadas y precisas, como se describe a continuación:

- Observación directa: En los primeros tiempos, la medición de la aceleración se realizaba principalmente mediante la observación directa del movimiento de los objetos. Galileo, por ejemplo, utilizaba un simple cronómetro de agua para medir el tiempo que tardaba un objeto en caer cierta distancia debido a la aceleración de la gravedad.
- Pendulares: Se desarrollaron dispositivos como el péndulo para medir la aceleración debido a la gravedad. La longitud del péndulo

- y el período de oscilación se relacionan con la aceleración gravitacional, lo que permitió mediciones más precisas.
- Aparatos mecánicos: Durante el siglo XIX, se diseñaron y fabricaron dispositivos mecánicos más sofisticados para medir la aceleración. Estos incluían acelerómetros basados en resortes y mecanismos de péndulos, que se utilizaban en aplicaciones científicas y técnicas.
 - Acelerómetros electrónicos: Con el avance de la tecnología electrónica en el siglo XX, surgieron los acelerómetros electrónicos. Estos dispositivos utilizan sensores piezoeléctricos, capacitivos o basados en MEMS (Sistemas Microelectromecánicos) para medir la aceleración. Son compactos, precisos y pueden proporcionar mediciones en tiempo real.
 - Acelerómetros láser y de fibra óptica: Estas tecnologías más recientes utilizan principios ópticos para medir la aceleración. Los acelerómetros láser miden los cambios en la interferencia de la luz láser para determinar la aceleración, mientras que los acelerómetros de fibra óptica utilizan variaciones en la longitud de onda de la luz que viaja a través de una fibra óptica para medir la aceleración [4].

Hoy en día, los acelerómetros se integran en una amplia gama de sistemas y dispositivos, desde teléfonos inteligentes hasta vehículos espaciales. Se utilizan en aplicaciones que van desde la navegación inercial y la monitorización de la salud hasta la detección de terremotos y la ingeniería estructural.

Las IMU tienen sus antecedentes en los sistemas de navegación inercial desarrollados principalmente para aplicaciones aeroespaciales y militares durante la Segunda Guerra Mundial y la Guerra Fría. Inicialmente se desarrollaron los sistemas de navegación inercial para ayudar a los aviones y misiles a mantener su rumbo y posición en ausencia de señales externas como las provenientes de sistemas de GPS (*Global Positioning System*). Estos sistemas utilizaban giroscopios y acelerómetros mecánicos para medir la velocidad y la orientación del vehículo en el espacio. Con el tiempo, los giroscopios y acelerómetros mecánicos fueron reemplazados por versiones más avanzadas basadas en tecnologías electrónicas y microelectromecánicas (MEMS, *microelectromechanical systems*). Estos dispositivos fueron más pequeños, livianos y precisos, lo que los hizo ideales para una gran variedad de aplicaciones, incluida la navegación inercial en vehículos terrestres, aéreos y marítimos [5]. En la Figura 1.1 se puede apreciar el esquema de un sistema de navegación inercial.

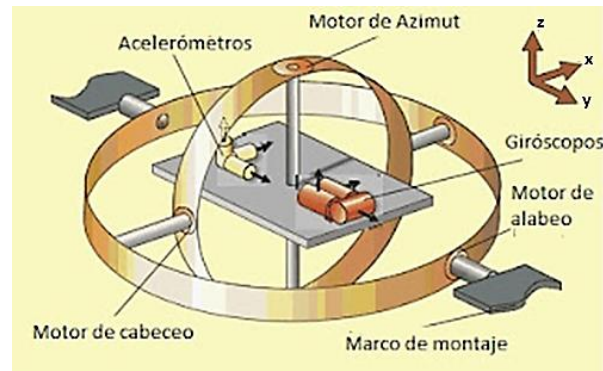


Figura 1.1 Esquema simple de sistema de navegación inercial

Fuente: aviación para todos (2020).

En las últimas décadas, se han producido avances significativos en la miniaturización, la precisión y la robustez de las IMU. Estos dispositivos ahora pueden integrarse en una sola unidad compacta que incluye giroscopios de estado sólido, acelerómetros y a menudo magnetómetros para medir la orientación absoluta en relación con el campo magnético terrestre [6].

Las IMU modernas se utilizan en una amplia gama de aplicaciones, desde sistemas de navegación y control de vehículos autónomos hasta sistemas de estabilización y realidad virtual. Su capacidad para proporcionar mediciones precisas de movimiento en tiempo real las hace indispensables en una variedad de campos donde se requiere la monitorización y el control del movimiento.

1.1 Aceleración y velocidad angular

La medición de velocidad angular y aceleración es esencial en una variedad de aplicaciones y campos, desde la navegación y el control de movimiento hasta el diseño de vehículos y la investigación científica. Estos datos son fundamentales para comprender y controlar el movimiento de objetos y sistemas en una variedad de entornos y condiciones. Con respecto a las aceleraciones en un sistema coordinado cartesiano inercial, como se aprecia en la Figura 1.2, la aceleración longitudinal es aquella que se presenta en la misma dirección de avance del vehículo (X), la aceleración lateral o transversal corresponde aquella presentada a lo largo del eje Y, finalmente, la aceleración vertical es aquella que actúa a lo largo del eje Z. En cuanto a las velocidades angulares, estas se pueden identificar como aquellas que se presentan alrededor de cada uno de los ejes del sistema cartesiano inercial (X, Y, Z); conocidas como balanceo (*Roll rate* en inglés), cabeceo (*Pitch rate* en inglés) y coleo (*Yaw rate* en inglés) respectivamente [7].

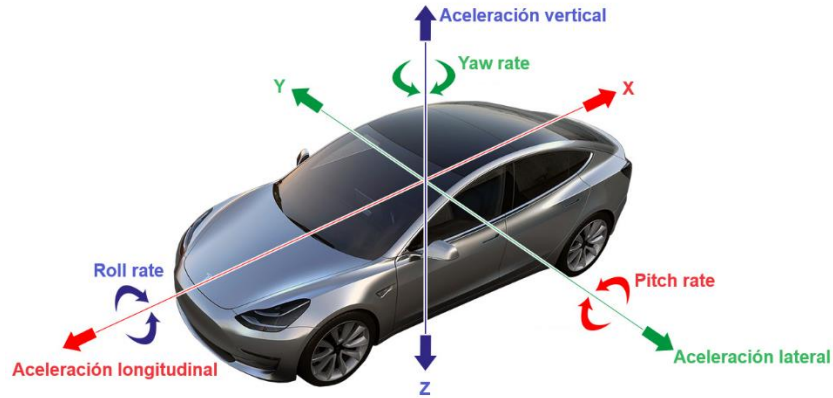


Figura 1.2 Esquema de aceleraciones y velocidades angulares en un vehículo

Fuente: blogmecanicos (2022).

Aunque matemáticamente el parámetro de desplazamiento puede ser obtenido a partir de la velocidad y esta a su vez del parámetro de aceleración, mediante el proceso de integración numérica, un acelerómetro no es el sensor más adecuado para medir la velocidad de un sistema y mucho menos su posición. Podríamos estar tentados a estimar la velocidad, a partir de la aceleración, sin embargo, este proceso acumula tanto los errores de medición como de ruido, lo que conlleva efectos de deriva con respecto al valor real. En todo caso, de no tener alternativa este proceso debería ser el último recurso y nunca tenerlo en consideración más allá de una estimación burda [10].

1.1.1 Aceleración longitudinal

La aceleración longitudinal se refiere a la componente de la aceleración de un objeto que actúa a lo largo de su dirección de movimiento. En otras palabras, es la aceleración experimentada por un objeto en la dirección en la que se está moviendo o acelerando.

Para entender mejor la aceleración longitudinal, consideremos un objeto en movimiento a lo largo de una línea recta. Si este objeto aumenta su velocidad con el tiempo, experimenta una aceleración positiva en la dirección del movimiento, lo que se conoce como aceleración longitudinal positiva. Por otro lado, si el objeto disminuye su velocidad con el tiempo, experimenta una aceleración negativa en la dirección del movimiento, conocida como desaceleración o aceleración longitudinal negativa [2].

La aceleración longitudinal se puede calcular utilizando la siguiente fórmula básica de cinemática:

$$a_x = \frac{\Delta v}{\Delta t}$$

Donde:

a_x es la aceleración.

Δv es el cambio en la velocidad del objeto.

Δt es el cambio en el tiempo.

La aceleración longitudinal es un concepto fundamental en la dinámica y la cinemática, y es crucial en numerosos campos, incluyendo la ingeniería automotriz, la aerodinámica, la física de partículas y muchas otras disciplinas donde el movimiento de objetos y sistemas se estudia y se analiza en relación con su aceleración.

1.1.2 Aceleración lateral

La aceleración lateral en un vehículo puede calcularse utilizando una Unidad de Medición Inercial. Una IMU generalmente contiene acelerómetros y giroscopios que miden la aceleración lineal y la velocidad angular en varias direcciones.

Suponiendo que el eje Y de la IMU está alineado con la dirección lateral del vehículo, la aceleración lateral, a_{lat} , se puede leer directamente de la componente Y del acelerómetro. El cálculo de la aceleración lateral se puede realizar utilizando la siguiente fórmula: [8].

$$a_{lat} = a_y$$

Donde:

a_{lat} es la aceleración lateral.

a_y es la aceleración medida en el eje Y de la IMU.

Si el vehículo está inclinado (por ejemplo, debido al peralte de una curva), se debe considerar la aceleración debida al efecto de la gravedad. Esto puede requerir el uso de las lecturas del giroscopio para determinar el ángulo de inclinación θ .

Si θ es el ángulo de inclinación lateral medido por el giroscopio, la aceleración lateral corregida sería:

$$a_{lat_c} = a_y - g * \sin(\theta)$$

Donde:

a_{lat_c} es la aceleración lateral corregida.

g es la aceleración de la gravedad (9.81 m/s^2)

Este proceso te permite calcular la aceleración lateral de un vehículo utilizando una IMU con precisión, considerando factores como la inclinación del vehículo.

1.1.3 Aceleración vertical

El registro de la aceleración vertical en vehículos se puede realizar utilizando una IMU o un acelerómetro instalado en el vehículo. La aceleración vertical es la componente de la aceleración que actúa en la dirección del eje z del sistema coordenado, es decir, hacia arriba o hacia abajo. Esta aceleración se debe principalmente a la fuerza gravitatoria que actúa sobre el vehículo.

El cálculo de la aceleración vertical se puede realizar de varias maneras, pero una de las formas más comunes es utilizando un acelerómetro, cabe mencionar que existen acelerómetros que permiten medir la aceleración de la gravedad aun cuando el vehículo está en reposo. La aceleración vertical se obtiene a partir de la medición de la aceleración en el eje z , que generalmente está alineado con la dirección vertical [9].

La fórmula básica para calcular la aceleración vertical utilizando el acelerómetro es:

$$a_{vertical} = a_z - g$$

Donde:

$a_{vertical}$ es la aceleración vertical.

a_z es la aceleración medida en el eje z de la IMU.

g es la aceleración debida a la gravedad, que es aproximadamente 9.81 m/s^2 .

Esta fórmula simplemente resta la componente de la aceleración debida a la gravedad (g) de la medición total de aceleración en el eje vertical (a_z), lo que nos da la aceleración vertical neta experimentada por el vehículo cuando está en reposo o se mueve sobre una superficie horizontal.

Es importante recalcar que el cálculo de la aceleración vertical se puede realizar en tiempo real para proporcionar información sobre la dinámica vertical del vehículo, particularmente para el caso de estudios que implican el análisis de la suspensión y el confort que brinda el vehículo.

1.2 Sistemas de medición de aceleración

Los primeros dispositivos para medir la aceleración fueron bastante simples y rudimentarios en comparación con las tecnologías modernas. Uno de los primeros métodos para medir la aceleración fue el uso de péndulos o plomadas. Estos dispositivos aprovechaban la propiedad de la gravedad para medir la aceleración experimentada por un objeto en caída libre. Registrando el desplazamiento angular del péndulo o la plomada, los científicos podían calcular la aceleración experimentada por el objeto [12]. En la Figura 1.3 se puede observar un diagrama de cuerpo libre de un péndulo simple.

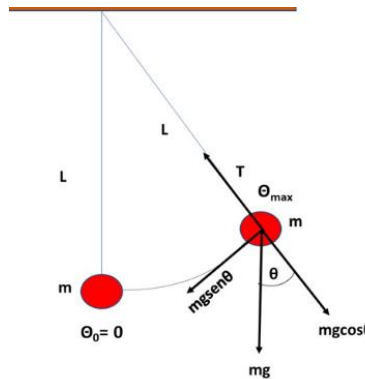


Figura 1.3 Diagrama de cuerpo libre en el péndulo simple

Fuente: HIVE BLOG (2022).

Pasando a dispositivos más especializados. Los acelerómetros de masa móvil son dispositivos que utilizan la inercia de una masa suspendida dentro de una estructura para medir la aceleración. Cuando el sistema se acelera, la masa interna se desplaza en relación con la estructura externa, lo que produce una fuerza proporcional a la aceleración. Esta fuerza se puede medir y convertir en una lectura de aceleración.

Los acelerómetros piezoeléctricos emplean cristales piezoeléctricos que generan una carga eléctrica cuando se deforman bajo fuerzas mecánicas, como la aceleración. Estos cristales están polarizados eléctricamente. Cuando se aplica una fuerza, los cristales se deforman, lo que genera una diferencia de potencial eléctrico proporcional a la fuerza aplicada. Esta señal eléctrica se amplifica y procesa para medir la aceleración. Un circuito integrado convierte la carga generada en una señal eléctrica utilizable. La magnitud y dirección de la aceleración se determinan según la intensidad y polaridad de la señal generada. Estos datos se envían a dispositivos externos para su procesamiento y aplicación, como la detección de impactos en dispositivos móviles, sistemas de navegación inercial o seguridad en vehículos. En resumen, los acelerómetros piezoeléctricos

transforman la energía mecánica en señales eléctricas gracias a la propiedad piezoeléctrica de los cristales, proporcionando una medida precisa de la aceleración en diversas aplicaciones tecnológicas [13]. En la Figura 1.4 se puede observar un diagrama de un acelerómetro piezoeléctrico.

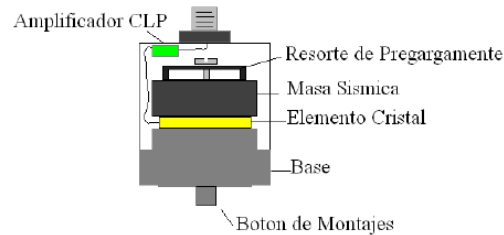


Figura 1.4 Diagrama de un acelerómetro piezoeléctrico

Fuente: Mediciones industriales (2008).

Un acelerómetro piezorresistivo detecta la aceleración mediante materiales que cambian su resistencia eléctrica bajo fuerzas mecánicas. Estos materiales están dispuestos en un puente de Wheatstone. Cuando el acelerómetro se mueve, los materiales piezorresistivos se deforman, alterando la resistencia eléctrica del puente. Un circuito electrónico detecta estos cambios y los convierte en señales eléctricas proporcionales a la aceleración. Estas señales se amplifican y procesan para calcular la magnitud y dirección de la aceleración. La información resultante se envía a dispositivos externos para su uso en diversas aplicaciones, como navegación inercial o control de movimiento. En resumen, los acelerómetros piezorresistivos utilizan el efecto piezorresistivo para medir cambios en la resistencia eléctrica, ofreciendo una forma eficaz de detectar y medir la aceleración en una variedad de aplicaciones tecnológicas [13]. En la Figura 1.5 se puede observar un diagrama de un acelerómetro piezorresistivo.

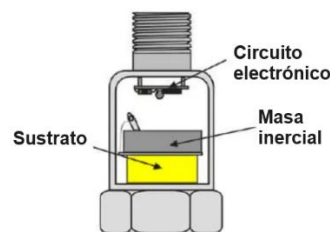


Figura 1.5 Diagrama de un acelerómetro piezorresistivo

Fuente: HIVE BLOG (2022).

Un acelerómetro capacitivo detecta cambios en la capacitancia entre placas paralelas cuando se mueve o experimenta aceleración. La distancia entre estas placas cambia con el movimiento, alterando la capacitancia.

Un circuito electrónico dentro del sensor monitorea estos cambios y los convierte en una señal eléctrica proporcional a la aceleración. Esta señal se procesa para calcular la magnitud y dirección de la aceleración. La información resultante se envía a dispositivos. En resumen, los acelerómetros capacitivos son dispositivos que convierten cambios en la capacitancia en señales eléctricas utilizadas para medir la aceleración y el movimiento en diversas aplicaciones tecnológicas [14]. En la Figura 1.6 se puede observar el diagrama de un acelerómetro capacitivo.

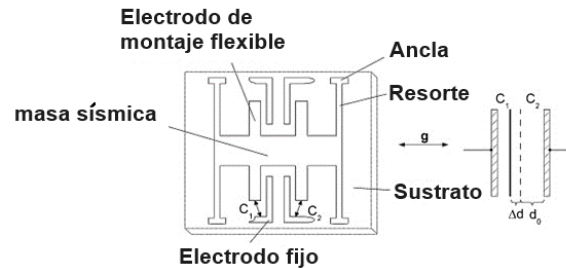


Figura 1.6 Diagrama de un acelerómetro capacitivo

Fuente: Engenharia e sistemas ibéricos Lda (2023).

Estos son solo algunos ejemplos de los primeros sistemas utilizados para medir la aceleración. Con el avance de la tecnología, se han desarrollado métodos más precisos y sofisticados, incluidos los acelerómetros basados en MEMS (sistemas microelectromecánicos), que son ampliamente utilizados en la actualidad en una variedad de aplicaciones, desde teléfonos inteligentes hasta vehículos espaciales. En la Figura 1.7 se puede observar un ejemplo de sistema MEMS.

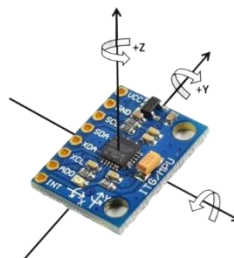


Figura 1.7 Ejemplo de sistema microelectromecánico

Fuente: github – MPU6050 (2023).

Los acelerómetros MEMS son dispositivos miniaturizados para detectar y medir la aceleración que emplean tecnología similar a la utilizada para la fabricación de semiconductores. La mayoría de estos sistemas son del tipo capacitivo que consisten en pequeñas estructuras mecánicas suspendidas sobre un sustrato de silicio conformando sistemas

mecánicos y electrónicos que eventualmente se conectan a circuitos integrados. Su desarrollo ha permitido la producción en masa de sensores de bajo costo y tamaño reducido, lo que los hace ideales para una variedad de aplicaciones, incluidos los vehículos.

En los vehículos actuales, los acelerómetros MEMS se utilizan ampliamente para una variedad de funciones críticas, como la detección de impactos para desplegar las bolsas de aire (*airbags*) en caso de colisión. También se emplean en sistemas de control de estabilidad, control de tracción, antibloqueo de frenos (ABS) y sistemas de navegación inercial. Su tamaño compacto y bajo consumo de energía los hacen ideales para integrarse en sistemas de control de vehículos.

El impacto de los acelerómetros MEMS en los vehículos actuales es significativo, ya que han mejorado la seguridad, la eficiencia y la comodidad del conductor y los pasajeros. Permiten una respuesta más rápida y precisa a los cambios en la aceleración, lo que resulta en una conducción más segura y suave. Además, su bajo costo ha facilitado su adopción generalizada en la industria automotriz, lo que ha llevado a mejoras significativas en la tecnología de vehículos y una mayor accesibilidad para los consumidores. En resumen, los acelerómetros MEMS han revolucionado la industria automotriz al proporcionar una tecnología precisa, confiable y rentable para una variedad de aplicaciones clave en los vehículos modernos [15].

1.2.1 Velocidad angular

Para calcular las velocidades angulares en vehículos, se utilizan unidades de medición inercial (IMU) o sistemas de navegación inercial que incluyen giroscopios y/o magnetómetros para medir las velocidades de rotación alrededor de los ejes espaciales. Los giroscopios miden la velocidad angular, es decir, la velocidad de rotación de un objeto alrededor de un eje específico. Se utilizan giroscopios de tres ejes para medir las velocidades angulares en los ejes de *roll* (balanceo), *pitch* (cabeceo) y *yaw* (coleo) del vehículo. Estas mediciones se realizan en unidades de radianes por segundo (rad/s) o grados por segundo (°/s) [8]. En la Figura 1.8 se puede apreciar un esquema general de las partes que integran un giroscopio.

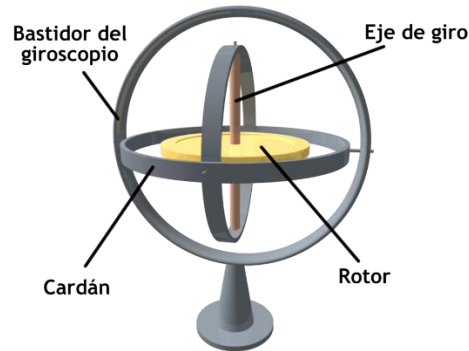


Figura 1.8 Esquema general de un giroscopio

Fuente: Wikipedia-Giróscopo (2023).

Para calcular las orientaciones angulares del vehículo a partir de las velocidades angulares medidas por los giroscopios, se utiliza la integración numérica. Sin embargo, similar al caso descrito para la obtención de velocidad a partir de la aceleración, este método acumula errores con el tiempo debido a las imprecisiones inherentes en las mediciones de los giroscopios y pueden verse afectados por errores de deriva y ruido, lo que puede afectar la precisión de las mediciones de las velocidades angulares. Se aplican técnicas de compensación de errores para corregir estos problemas, como la calibración inicial de los sensores, la estimación y corrección de la deriva por medio del uso de algoritmos de filtrado adaptativos [11].

La velocidad angular (ω) se define como la tasa de cambio del ángulo a través del cual un objeto gira en un período de tiempo dado. Se puede calcular utilizando la siguiente fórmula básica [11]:

$$\omega = \frac{\Delta\theta}{\Delta t}$$

Donde:

ω es la velocidad angular (rad/s).

$\Delta\theta$ es el cambio en el ángulo (rad).

Δt es el cambio en el tiempo (s).

Si se trata de un movimiento angular uniformemente acelerado o desacelerado, donde la velocidad angular cambia constantemente, se puede utilizar la siguiente fórmula para calcular la velocidad angular en cualquier instante:

$$\omega_f = \omega_i + \alpha \cdot t$$

Donde:

ω_f es la velocidad angular final (rad/s).

ω_i es la velocidad angular inicial (rad/s).

α es la aceleración angular (rad/s²).

t es el tiempo transcurrido desde el inicio (s).

Esta fórmula es análoga a la ecuación de movimiento lineal bajo aceleración constante ($v_f = v_i + a \cdot t$), donde v_f y v_i son las velocidades final e inicial, respectivamente, y a es la aceleración. Es importante tener en cuenta que la velocidad angular puede ser positiva si el objeto gira en sentido antihorario y negativa si gira en sentido horario, dependiendo de la convención de signos utilizada.

1.3 Sistemas de medición de velocidad angular

Los primeros dispositivos para medir la velocidad angular se remontan a dispositivos mecánicos simples como los giroscopios y los giróscopos. Estos dispositivos fueron desarrollados en el siglo XIX y principios del siglo XX y se utilizaron principalmente en aplicaciones de navegación y control de orientación. A continuación, se mencionan 3 ejemplos de sistemas para medición de velocidad angular utilizados a lo largo de la historia.

- **Giroscopio mecánico:** El giroscopio mecánico es uno de los primeros dispositivos utilizados para medir la velocidad angular. Consiste en un disco giratorio montado en un soporte que le permite mantener su orientación espacial a pesar de los movimientos del vehículo. La velocidad angular se puede determinar midiendo el ángulo a través del cual el giroscopio se desplaza en un período de tiempo dado.
- **Giróscopo de suspensión libre:** Este tipo de giróscopo se basa en el principio de la conservación del momento angular. Consiste en un rotor giratorio suspendido en un marco con articulaciones flexibles. La velocidad angular se puede calcular midiendo la desviación del eje del rotor con respecto a la dirección de referencia.
- **Giróscopo láser:** Los giróscopos láser, también conocidos como giróscopos de fibra óptica, son una forma más moderna de medir la velocidad angular. Utilizan el efecto Sagnac, que se basa en la interferencia de la luz que viaja en direcciones opuestas alrededor de un circuito cerrado. El cambio en el patrón de interferencia, obtenido de las dos trayectorias de luz, se utiliza para determinar la velocidad angular [16, 17].

Estos dispositivos fueron fundamentales en aplicaciones de navegación marítima, aeronáutica y espacial, así como en la industria militar. Aunque han sido reemplazados en gran medida por tecnologías más avanzadas como los acelerómetros MEMS y los sistemas de navegación inercial, estos primeros dispositivos sentaron las bases para el desarrollo de tecnologías modernas de medición de velocidad angular.

2. Marco teórico

El concepto básico de diseño para el sistema inercial de este proyecto se basó en diferentes componentes electrónicos, en los que resaltan un sistema en chip (SoC, por sus siglas en inglés, "*System on a Chip*") ESP32 como circuito de control y adquisición, el sensor inercial MPU-6050 que permite obtener la información de aceleración y velocidades angulares en los 3 ejes, un módulo de tarjeta micro-SD que permite almacenar la información y el desarrollo de circuitos de acondicionamiento de voltaje de alimentación, interfaz gráfica a través de una pequeña pantalla, botones para la navegación por la interfaz y una batería interna. A continuación, se describen las características técnicas de los principales componentes, así como los protocolos de comunicación utilizados durante el desarrollo y operación.

2.1 Módulo ESP32

El módulo ESP32 representa una solución completa y certificada para conectividad Wi-Fi/Bluetooth. No solo incluye la radio inalámbrica, sino también un potente procesador integrado con múltiples interfaces para conectar una variedad de periféricos. Destacando su versatilidad, el procesador cuenta con dos núcleos de procesamiento independientes, lo que permite un control flexible de las frecuencias operativas, que van desde 80 megahercios (MHz) hasta 240 MHz. Esta capacidad de ajuste brinda a los desarrolladores la posibilidad de optimizar el rendimiento según las necesidades específicas de sus aplicaciones [18]. Los periféricos del procesador facilitan la conexión a una variedad de interfaces externas como:

Interfaz periférica serial (SPI)

- I2C
- Transmisor receptor asíncrono universal (UART)
- I2S
- Ethernet
- Tarjetas SD
- Interfaces táctiles y capacitivas

El ESP32 ha ganado una popularidad considerable, lo que ha generado una variedad de opciones para desarrollar y programar el dispositivo. Entre las herramientas de desarrollo más destacadas se encuentran:

- El marco de desarrollo de IoT (ESP-IDF) de *Espressif Systems* [18], que ofrece un conjunto completo de herramientas y bibliotecas para aprovechar al máximo las capacidades del ESP32, ideal para proyectos más avanzados que requieren un control preciso del hardware.
- El entorno completo de desarrollo integrado (IDE) de Arduino, una opción muy popular entre los principiantes y los aficionados que buscan una curva de aprendizaje suave y una amplia comunidad de soporte.
- *MicroPython*, una implementación de Python 3 diseñada específicamente para sistemas embebidos, que brinda una experiencia de programación sencilla y accesible para aquellos familiarizados con el lenguaje Python.

Estas opciones ofrecen flexibilidad para adaptarse a las preferencias y necesidades de los desarrolladores, desde aquellos que requieren un control total sobre el hardware hasta aquellos que buscan una solución rápida y fácil de implementar. Finalmente, para los desarrolladores que estén interesados en desarrollar su aplicación en Python, el ESP32 es compatible con el núcleo de código abierto MicroPython. Los desarrolladores pueden cargar MicroPython en el ESP32 y luego desarrollar scripts de Python para su aplicación. Esto puede hacer que sea muy fácil actualizar la aplicación sobre la marcha en un entorno industrial y eliminar las capas de experiencia requeridas que normalmente vienen con el desarrollo integrado [18]. En la Figura 2.1 se puede observar una placa de desarrollo ESP32.



Figura 2.1 ESP32-WROOM-32U

Fuente: Digikey (2020).

2.2 Sensor MPU-6050

La unidad de procesamiento de movimiento MPU-6050 representa un hito en la industria al ser la primera solución en ofrecer fusión integrada de sensores de 9 ejes, 6 internos correspondientes a los 6 grados de libertad de aceleraciones lineales y velocidades angulares, y 3 externos correspondientes a los grados de libertad de un magnetómetro. Su tecnología incluye algoritmos de *firmware MotionFusion™*, probados y utilizados en diversos dispositivos de consumo como teléfonos móviles, tabletas, controladores de juegos, entre otros. Esta fusión de sensores proporciona una experiencia de movimiento fluida y precisa en una variedad de aplicaciones, desde mandos a distancia hasta dispositivos de realidad virtual.

El sensor MPU-6050 combina un giroscopio MEMS de 3 ejes con un acelerómetro MEMS de 3 ejes, todo en un solo chip de silicio, junto con un procesador de movimiento digital (DMP™). Esto permite que el sensor procese algoritmos complejos de movimientos de 9 ejes, gracias a su capacidad para fusionar los datos de los sensores de forma precisa y eficiente. Desde el seguimiento de la orientación en dispositivos portátiles hasta la estabilización de imágenes en drones, el MPU-6050 es una pieza clave en numerosas aplicaciones que requieren detección de movimiento de alta calidad.

El MPU-6050 ofrece una solución efectiva para eliminar los problemas de alineación del eje transversal que pueden surgir en distintas etapas, permite incluso la conexión de magnetómetros externos u otros sensores a través de un bus I2C auxiliar maestro. Esto posibilita la integración de un conjunto completo de dispositivos sensores de datos sin requerir la intervención del procesador del sistema principal.

Este sensor, también conocido como IMU de seis ejes (6 DOF), proporciona seis valores de salida que representan la orientación y movimiento en el espacio tridimensional. Su procesador digital de movimiento (DMP) integrado se encarga de calcular los algoritmos de procesamiento de movimiento, liberando así al procesador principal de dicha tarea. Los datos resultantes pueden ser accedidos a través de los registros de la DMP o almacenados en un FIFO (*First Input-First Output*), mientras que el DMP puede generar interrupciones a través de uno de los pines externos del sensor.

La precisión del MPU-6050 es notable, gracias a su conversión de 16 bits de A/D por cada canal, lo que permite una digitalización precisa de las

salidas del acelerómetro en los ejes x, y, y z simultáneamente. Conectado mediante el bus I2C, este sensor es compatible con Arduino, aunque algunas versiones cuentan con un regulador que permite su uso con alimentación de 5V. Es importante tener en cuenta el rango de voltajes de operación, que va desde 2.37V a 3.45V, para un rendimiento óptimo y seguro [19]. En la Figura 2.2 se puede observar una tarjeta MPU-6050.



Figura 2.2 Módulo MPU-6050

Fuente: Digikey (2020).

La hoja de datos del MPU6050, ofrece los números utilizados para la configuración de este. Por ejemplo, para activar el modo de configuración del sensor y modificar sus parámetros internos, se utiliza el registro 6B. En este mismo registro el usuario puede configurar el modo de alimentación y la fuente del reloj, también se describe como restablecer todo el dispositivo o como desactivar el sensor de temperatura.

Para la configuración del giroscopio se utiliza el registro 1B. Este registro desempeña un papel crucial en la configuración del giroscopio. Se utiliza para activar la autocomprobación del giroscopio y ajustar el rango de escala completa de los giroscopios, lo que permite adaptar la sensibilidad del sensor a las necesidades específicas de la aplicación.

Continuando con los parámetros de configuración para el acelerómetro se accede al registro. Similar al registro 1B, pero para el acelerómetro, este registro activa la auto prueba del acelerómetro y configura su rango de escala completa. Además, también controla el filtro de paso alto digital (DHPF), lo que permite filtrar el ruido no deseado y mejorar la precisión de las mediciones.

Por último, hay que considerar la configuración del registro 1A para el filtro pasa bajas. Este registro se encarga de configurar el filtro pasa bajas de la salida, controlando el muestreo de clavijas de Sincronización de Fotogramas Externo (FSYNC) y el Filtro de Paso Bajo Digital (DLPF) tanto para los giroscopios como para los acelerómetros. Es importante tener en

cuenta que el filtrado de la salida introduce un retardo en las lecturas del sensor, que puede ser de hasta 20ms, por lo que se debe considerar este retardo al configurar el filtro para garantizar una respuesta adecuada del sistema [20].

2.3 Módulo lector de memoria microSD

El módulo lector de memoria microSD (ver Figura 2.3) ha sido específicamente diseñado para facilitar el acceso a la memoria microSD en modo SPI, lo que se refleja en la clara etiquetación de las señales de control conforme al bus de comunicaciones SPI. Este módulo es compatible con tarjetas microSD y microSDHC, ofreciendo una amplia gama de opciones de almacenamiento.

Una característica destacada de este módulo es su capacidad de adaptarse a diferentes niveles de voltaje, gracias al chip MC74VHCT125A utilizado como circuito de conversión de voltaje lógico. Esto permite la comunicación tanto a 3.3V como a 5V, lo que lo hace compatible con una variedad de dispositivos y microcontroladores.

Además, el módulo cuenta con un regulador de voltaje integrado en placa que permite alimentarlo hasta con 5V, lo que brinda flexibilidad en cuanto a las fuentes de alimentación disponibles. Gracias a su compatibilidad con Arduino y otros microcontroladores, así como con una amplia gama de tarjetas de desarrollo, este módulo lector de memoria MicroSD es una opción versátil y práctica para proyectos que requieren acceso a almacenamiento externo de manera sencilla y eficiente [21].

Algunas características de este módulo son:

- Voltaje de Operación: 3.3V-5V
- Interfaz: SPI
- Cuenta con todos los pines SPI de la tarjeta SD: MOSI, MISO, SCK, CS
- Chip conversor lógico: MC74VHCT125A
- Dimensiones: 42*24 mm



Figura 2.3 módulo lector de memoria microSD

Fuente: aelectronic's (2020).

2.4 Comunicación SPI

La Interfaz Periférica Serial (SPI, por sus siglas en inglés) fue desarrollada por Motorola (ahora parte de NXP Semiconductors) alrededor de 1985. Esta interfaz serial síncrona fue diseñada para permitir la comunicación entre dispositivos a corta distancia. Desde entonces, se ha convertido en un estándar de facto ampliamente adoptado por numerosos fabricantes de semiconductores, especialmente en microprocesadores y microcontroladores.

La popularidad de SPI se debe a varias ventajas significativas. En primer lugar, es una interfaz direccionada por hardware simple que ofrece una flexibilidad completa en cuanto a la cantidad de bits transferidos. Utiliza un modelo maestro-esclavo con un solo maestro, lo que le permite manejar varios dispositivos esclavos mediante comunicaciones dúplex a velocidades de reloj de hasta 50 MHz. Además, no utiliza un protocolo estándar y transfiere únicamente paquetes de datos, lo que la hace ideal para la transferencia de flujos de datos largos.

La interfaz SPI utiliza un máximo de cuatro líneas de señal. El dispositivo maestro, típicamente un procesador o controlador, suministra y controla el reloj (SCK) y las líneas de selección de chip (CS). La comunicación bidireccional se realiza a través de las líneas de datos *Master Out Slave In* (MOSI) y *Master In Slave Out* (MISO). En configuraciones con un único maestro y dispositivos secundarios individuales, la línea de selección de chip puede ser eliminada y se puede forzar la entrada de CS al estado lógico habilitado. Además, si el dispositivo secundario solo puede enviar datos (comunicación semidúplex), la línea MOSI también puede ser eliminada para reducir aún más el número de señales requeridas. La transferencia de datos en SPI se realiza de manera sincronizada con el reloj, lo que se asemeja a un registro de desplazamiento con un bit transmitido o recibido en cada pulso del reloj [22].

En el protocolo SPI, el dispositivo maestro controla y genera el reloj que sincroniza la comunicación con los dispositivos secundarios. Dos atributos fundamentales del reloj en SPI son la polaridad del reloj (CPOL) y la fase del reloj (CPHA). Estos atributos determinan el comportamiento del reloj y cómo se sincronizan los datos entre el maestro y los dispositivos secundarios.

La polaridad del reloj (CPOL) especifica el nivel de voltaje en el que se encuentra el reloj cuando no está activo. Cuando CPOL = 0, el reloj se encuentra en nivel bajo (lógica 0) cuando está inactivo, mientras que CPOL = 1 indica que el reloj está en nivel alto (lógica 1) en su estado inactivo.

Por otro lado, la fase del reloj (CPHA) determina en qué borde del reloj se capturan o se establecen los datos. Cuando CPHA = 0, los datos se capturan o se establecen en el borde principal del reloj, es decir, cuando el reloj cambia de inactivo a activo (flanco ascendente). En contraste, con CPHA = 1, los datos se capturan o se establecen en el borde de rastreo del reloj, es decir, cuando el reloj cambia de activo a inactivo (flanco descendente).

Estos atributos CPOL y CPHA permiten configurar la sincronización precisa de los datos entre el maestro y los dispositivos secundarios, lo que proporciona flexibilidad en la comunicación SPI y la adaptabilidad a las necesidades específicas de cada aplicación [22].

2.5 Interfaz I2C

I2C, o *Inter-Integrated Circuit*, es un puerto y protocolo de comunicación serial que establece la estructura de los datos y las conexiones físicas necesarias para transferir bits entre dos dispositivos digitales. Este puerto se compone de dos cables de comunicación principales: SDA (Serial Data Line) y SCL (*Serial Clock Line*). El protocolo permite conectar hasta 127 dispositivos esclavos a través de estas dos líneas, con velocidades de transmisión que pueden alcanzar los 100, 400 y 1000 kbits/s. I2C también es conocido como IIC (Inter-IC) o TWI (*Two Wire Interface*).

Una de las principales ventajas del protocolo I2C es su capacidad para conectar múltiples dispositivos al mismo bus de comunicación, lo que lo hace especialmente útil en sistemas con numerosos periféricos digitales. En comparación con otros protocolos seriales, como Serial TTL, I2C ofrece una trama de comunicación más completa y detallada, lo que permite el envío de mensajes más complejos y específicos.

Cuando se envían mensajes a través de un puerto I2C, cada trama de datos incluye un byte de información junto con direcciones tanto del registro como del sensor. Además, el protocolo I2C proporciona una confirmación de recepción por parte del dispositivo receptor para cada dato enviado, lo que garantiza la integridad de la comunicación. Esta capacidad de confirmación en la misma trama de datos es una de las ventajas clave de I2C en comparación con otros protocolos seriales, como el puerto Serial estándar, donde la confirmación puede requerir tramas de datos adicionales. Diferenciar los diferentes elementos involucrados en la comunicación I2C es fundamental para garantizar una comunicación eficiente y sin errores [23].

Cuando se trata del MPU-6050, este dispositivo siempre funciona como un esclavo cuando se comunica con el procesador del sistema, que actúa

como maestro en la comunicación I2C. Las líneas SDA y SCL suelen requerir resistencias *pull-up* para garantizar niveles de señal adecuados. La velocidad máxima de transmisión del bus I2C es de 400 kHz.

La dirección esclava del MPU-6050 está establecida en b110100X, con una longitud de 7 bits. El bit menos significativo (LSB) de la dirección de 7 bits está determinado por el nivel lógico en el pin AD0. Esta configuración permite conectar dos MPU-6050 al mismo bus I2C. En esta configuración, la dirección de uno de los dispositivos debe ser b1101000 (cuando el pin AD0 está lógicamente bajo) y la dirección del otro debe ser b1101001 (cuando el pin AD0 está lógicamente alto) [23].

3. Metodología

Para el desarrollo del dispositivo se empleó una metodología en “V” la cual ha sido ampliamente utilizada para desarrollos tecnológicos. Para este desarrollo primeramente se realizó una lista para definir los requerimientos mínimos que deberán observarse en el dispositivo tanto a nivel componentes como de material utilizado para su implementación, a fin de que sea funcional para la aplicación que se requiere, más adelante se detallarán los puntos considerados. Por otro lado, una vez construido se requiere de una etapa de evaluación, tanto a nivel modular como integrado, finalizando con pruebas de aceptación. En la Figura 3.1 muestra los pasos seguidos en la metodología.

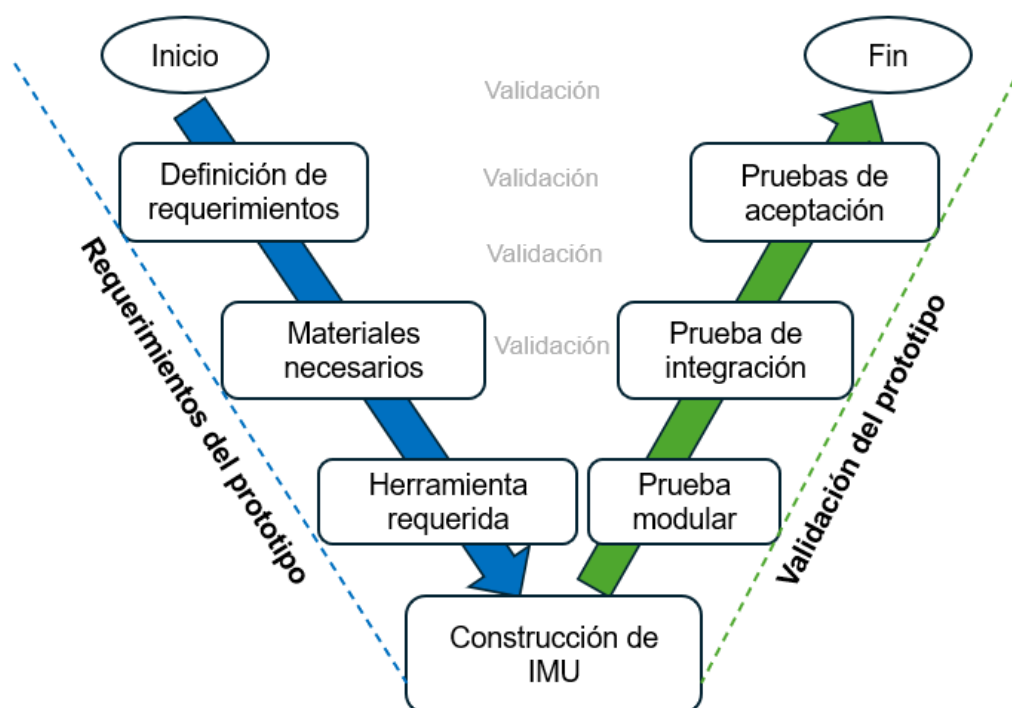


Figura 3.1 Metodología “V” para el desarrollo de IMU

3.1 Prototipo IMU

Para el desarrollo del prototipo se consideró como punto de partida dispositivos comerciales similares, donde una gran mayoría de dispositivos necesitan estar conectados a una tarjeta de adquisición de datos externa, por lo cual para el desarrollo de este prototipo se busca que tenga la capacidad de ser un todo en uno, lo cual lo vuelve un sistema completo para medición de aceleraciones y velocidades angulares. Los requerimientos del prototipo consideran 3 etapas, las cuales consisten en primeramente definir los requerimientos mínimos con los que debe contar el desarrollo. Como segunda etapa abarca la lista de materiales necesarios para la construcción del prototipo, en donde se aborda la lista de materiales adicionales para realizar la conexión y la identificación de características de dichos componentes. Por último, se describe un poco sobre el material y herramienta para la construcción integral del prototipo.

3.1.1 Definición de requerimientos

Los requerimientos establecidos para esta etapa del proyecto pretenden apegarse a:

- Medir aceleración longitudinal
- Medir aceleración lateral
- Medir aceleración vertical
- Medir velocidad angular de cabeceo (*Pitch rate*)
- Medir velocidad angular de balanceo (*Roll rate*)
- Medir velocidad angular de coleo (*Yaw rate*)
- Contar con una batería de alimentación
- Contar con una pantalla para visualización.
- Disponer de botones de navegación y encendido
- Contar con una estructura modular
- Contar con sistema de almacenamiento interno
- Contar con la capacidad de comunicación inalámbrica

Con estos requerimientos se busca construir un sistema integral que permita adquirir aceleraciones lineales y velocidades angulares en pruebas de evaluación de comportamiento dinámico en vehículos.

3.1.2 Materiales utilizados

A continuación, se coloca la lista de componentes utilizados para la construcción del IMU, abarcando las tarjetas y módulos principales, así como los materiales adicionales para la interconexión de todo.

- Módulo ESP-32 con salida para antena
- Antena SMA para ESP-32
- Interruptor tipo botón
- Módulo MPU-6050
- Módulo lector microSD
- Pantalla Oled Azul I2C
- Batería LiPo 3.7 VDC 1000 mAh
- Conectores tipo C macho y hembra
- Módulo Boost Step Up cargador de baterías Li-Ion/LiPo
- Módulo Step Down
- Cables de diferentes longitudes
- Conectores SHI de 2 y 4 pines macho y hembra
- Plástico PETG para impresión 3D
- Placas de cobre para circuitos electrónicos
- Tornillería M3

3.1.3 Herramientas y material de construcción

Para la construcción también se recurrió a diferente herramienta para la manufactura de la estructura contenedora del dispositivo, así como la necesaria para la conexión de módulos y circuitos electrónicos. A continuación, se presenta una pequeña lista de herramienta para manufactura del dispositivo.

- Impresora 3D de plástico
- Estación de soldadura
- Pinza pelacables
- Ponchadora de conectores SHI
- Pinzas de corte para electrónica

3.1.4 Construcción de IMU

Para la fabricación de la estructura se utilizó una máquina de impresión 3D de plásticos, el material utilizado fue El PETG (tereftalato de polietileno glicol) es un material termoplástico ampliamente utilizado en la impresión 3D debido a sus propiedades mecánicas, químicas y de impresión favorables. Es una variante del PET (tereftalato de polietileno) que ha sido modificado para mejorar su capacidad de impresión y su resistencia.

El PETG es conocido por su durabilidad, resistencia al impacto, transparencia y resistencia química. También tiene una buena resistencia al calor, lo que lo hace adecuado para aplicaciones que requieren cierto grado de resistencia térmica. Además, es un material fácil de imprimir en comparación con otros filamentos, ya que tiene una temperatura de

impresión relativamente baja y es menos propenso a deformarse o encogerse durante el proceso de impresión.

Antes de realizar la impresión 3D se realizaron diferentes diseños en software especializado, donde a través del dibujo asistido por computadora se realizaron las modificaciones necesarias adecuadas a la aplicación que se buscaba desarrollar. En la Figura 3.2 se puede observar una impresora 3D modelo CR-10 Smart Pro utilizada para la fabricación de la estructura contenedora del dispositivo.



Figura 3.2 Metodología “V” para el desarrollo de IMU

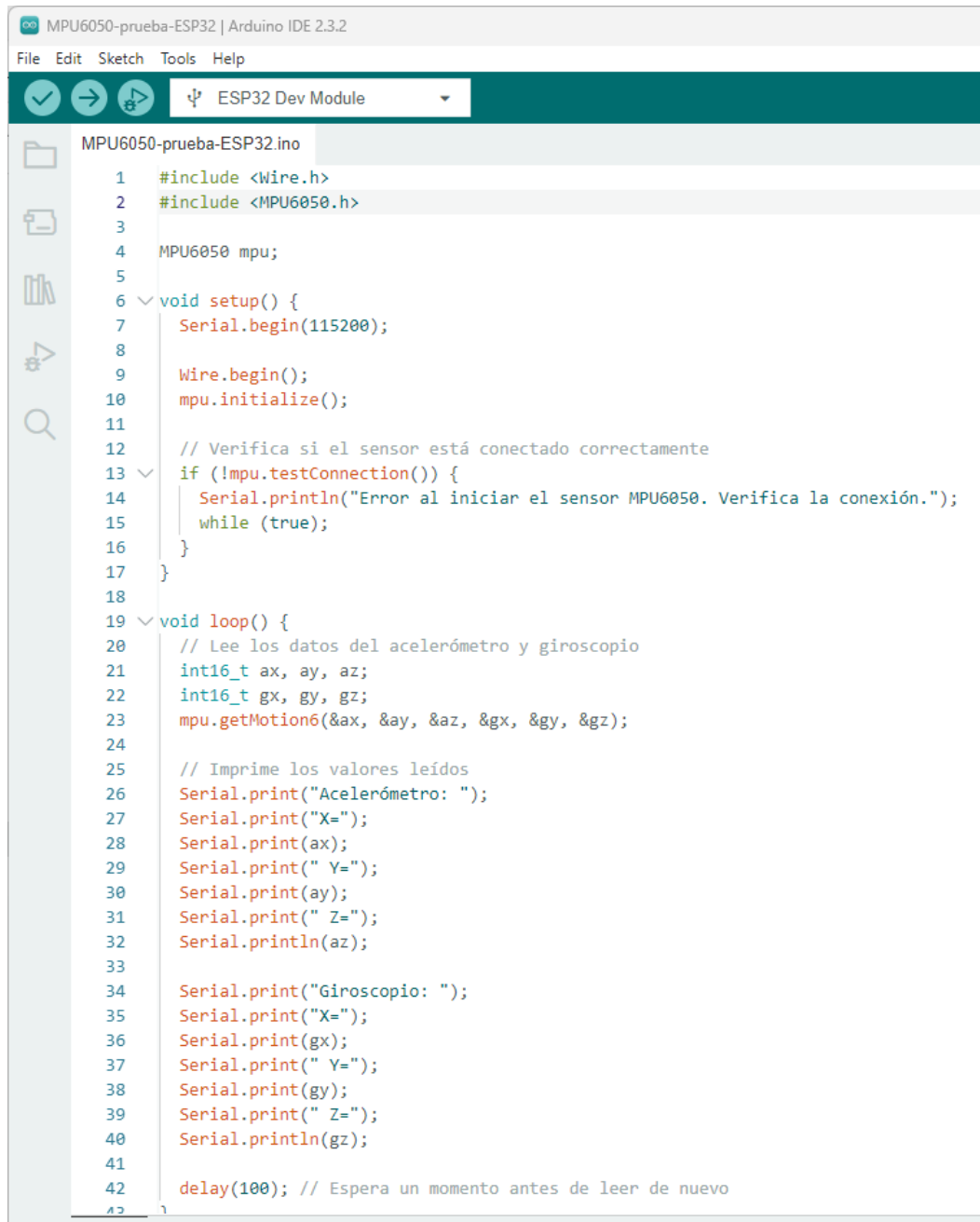
Fuente: Creality (2024).

3.2 Validación del prototipo

La validación del prototipo consta de 3 etapas, la primera es una prueba modular, esto con el fin de probar de forma independiente cada módulo que se conectara al ESP-32. En la segunda etapa se realiza una validación integral, se revisa el correcto funcionamiento de los módulos integrados al ESP-32. Por último, se realiza una prueba de aceptación, la cual consiste en evaluar la respuesta del prototipo ante aceleraciones conocidas, para lo cual se utilizará un sistema generador de vibraciones que cuenta con una caracterización realizada por el Centro Nacional de Metrología (CENAM).

3.2.1 Prueba modular

Para llevar a cabo la validación del prototipo de IMU, primeramente, se realizó una prueba modular, la cual consistió en probar únicamente el circuito MPU-6050 conectada a la tarjeta de adquisición ESP-32, a la cual se programó a través del software de Arduino IDE, donde el programa simplemente consistió en adquirir las señales provenientes del circuito MPU-6050. A continuación, en la Figura 3.3 se presenta el código utilizado para dicha prueba.



```

MPU6050-prueba-ESP32 | Arduino IDE 2.3.2
File Edit Sketch Tools Help
ESP32 Dev Module
MPU6050-prueba-ESP32.ino
1  #include <Wire.h>
2  #include <MPU6050.h>
3
4  MPU6050 mpu;
5
6  void setup() {
7    Serial.begin(115200);
8
9    Wire.begin();
10   mpu.initialize();
11
12   // Verifica si el sensor está conectado correctamente
13   if (!mpu.testConnection()) {
14     Serial.println("Error al iniciar el sensor MPU6050. Verifica la conexión.");
15     while (true);
16   }
17 }
18
19 void loop() {
20   // Lee los datos del acelerómetro y giroscopio
21   int16_t ax, ay, az;
22   int16_t gx, gy, gz;
23   mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
24
25   // Imprime los valores leídos
26   Serial.print("Acelerómetro: ");
27   Serial.print("X=");
28   Serial.print(ax);
29   Serial.print(" Y=");
30   Serial.print(ay);
31   Serial.print(" Z=");
32   Serial.println(az);
33
34   Serial.print("Giroscopio: ");
35   Serial.print("X=");
36   Serial.print(gx);
37   Serial.print(" Y=");
38   Serial.print(gy);
39   Serial.print(" Z=");
40   Serial.println(gz);
41
42   delay(100); // Espera un momento antes de leer de nuevo
43 }

```

Figura 3.3 Prueba modular ESP-32 con MPU 6050

El código consiste en adquirir y visualizar en el monitor serial las señales de las aceleraciones en las direcciones x , y y z , así como las señales de velocidad angular sobre x (*Pitch rate*), y (*Roll rate*) y z (*Yaw rate*). Este código cumple con la función de adquirir señales del sensor, pero aún falta el complemento para guardar datos en una memoria microSD, transmitir información de forma inalámbrica.

Siguiendo con las pruebas modulares también se implementó un código en el ESP-32 para almacenar datos en un módulo lector microSD. En la

Figura 3.4 se muestra un extracto del código utilizado para probar el módulo microSD. En este código, se inicializa la comunicación con la tarjeta microSD en el pin seleccionado (en este caso, el pin 5). Luego, se crea un archivo llamado "data.txt" en la raíz de la tarjeta microSD y se escribe un encabezado. En el *bucle loop*, se generan datos de ejemplo (en este caso, se lee un valor analógico del pin A0) y se escriben en el archivo "data.txt" en cada iteración (ver figura 3.4).



```
1 #include <SPI.h>
2 #include <SD.h>
3
4 File dataFile;
5 const int chipSelect = 5; // El pin CS de la tarjeta microSD puede ser cualquier pin digital
6
7 void setup() {
8   Serial.begin(115200);
9
10  // Inicializa la comunicación con la tarjeta microSD
11  if (!SD.begin(chipSelect)) {
12    Serial.println("Error al inicializar la tarjeta microSD.");
13    return;
14  }
15
16  Serial.println("Tarjeta microSD inicializada correctamente.");
17
18  // Crea un archivo llamado "data.txt" en la raíz de la tarjeta microSD
19  dataFile = SD.open("data.txt", FILE_WRITE);
20
21  // Si el archivo se abrió correctamente, escribe un encabezado
22  if (dataFile) {
23    dataFile.println("Datos guardados desde ESP32:");
24    dataFile.close();
25    Serial.println("Archivo 'data.txt' creado correctamente.");
26  } else {
27    Serial.println("Error al abrir el archivo 'data.txt'.");
28  }
29 }
30
31 void loop() {
32   // Genera algunos datos de ejemplo
33   int sensorValue = analogRead(A0);
34
35   // Abre el archivo en modo de escritura al final del mismo
36   dataFile = SD.open("data.txt", FILE_WRITE);
37
38   // Si el archivo se abrió correctamente, escribe los datos
39   if (dataFile) {
40     dataFile.print("Valor del sensor: ");
41     dataFile.println(sensorValue);
42     dataFile.close();
43     Serial.println("Datos guardados correctamente.");
44   } else {
45     Serial.println("Error al abrir el archivo 'data.txt'.");
46   }
47 }
```

Figura 3.4 Prueba modular ESP-32 con módulo lector microSD

Continuando con las pruebas modulares, se realizó una prueba para ver las capacidades del ESP-32 para transmitir datos de forma inalámbrica a

través de la red Wi-Fi. En este código, el ESP32 se conecta a una red Wi-Fi utilizando las credenciales proporcionadas en `ssid` y `password`. Luego, inicia un servidor TCP/IP en el puerto 80. Cuando un cliente se conecta al servidor, el ESP32 lee cualquier solicitud que el cliente envíe y responde con un mensaje simple. Hay que asegurar que `"tu_ssid_wifi"` y `"tu_contraseña_wifi"` sea el nombre y contraseña de la red Wi-Fi donde se transmitirá la información. Para probar este código, hay que abrir un navegador web en tu PC y navegar a la dirección IP asignada al ESP32 (que se imprimirá en el Monitor Serie). Aparecerá el mensaje "Hola desde ESP32!" en la página web. En la Figura 3.5 se puede observar un extracto del código utilizado para esta prueba.



```

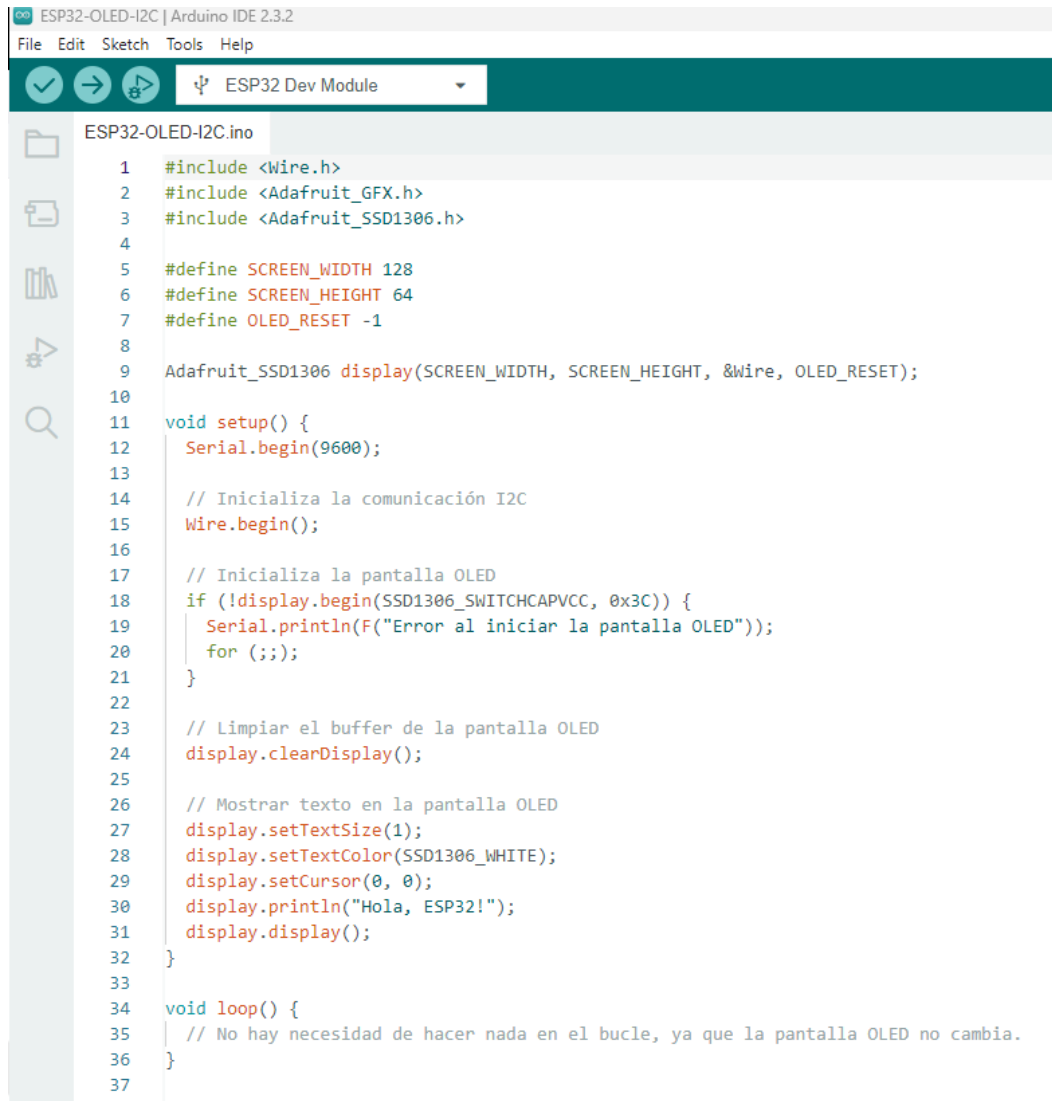
1  #include <WiFi.h>
2
3  const char* ssid = "tu_ssid_wifi";
4  const char* password = "tu_contraseña_wifi";
5
6  WiFiServer server(80);
7
8  void setup() {
9      Serial.begin(115200);
10
11     // Conéctate a la red WiFi
12     Serial.println();
13     Serial.println();
14     Serial.print("Conectando a ");
15     Serial.println(ssid);
16     WiFi.begin(ssid, password);
17
18     while (WiFi.status() != WL_CONNECTED) {
19         delay(500);
20         Serial.print(".");
21     }
22     Serial.println("");
23     Serial.println("WiFi conectado.");
24
25     // Inicia el servidor TCP/IP en el puerto 80
26     server.begin();
27     Serial.println("Servidor iniciado.");
28     Serial.print("Dirección IP: ");
29     Serial.println(WiFi.localIP());
30 }
31
32 void loop() {
33     // Espera a que se conecte un cliente
34     WiFiClient client = server.available();
35     if (!client) {
36         return;

```

Figura 3.5 Prueba de comunicación Wi-Fi

Para conectar una pantalla OLED I2C a un ESP32 hay que descargar la librería "Adafruit SSD1306" e instalarla en Arduino IDE, esta es la librería

más comúnmente utilizada para manejar pantallas OLED I2C con ESP32. Este código inicializa la pantalla OLED con comunicación I2C y muestra un mensaje de "Hola, ESP32!" en la pantalla. Este ejemplo solo es para probar el correcto funcionamiento de la pantalla con el ESP-32. En la Figura 3.6 se puede observar un ejemplo del código utilizado.



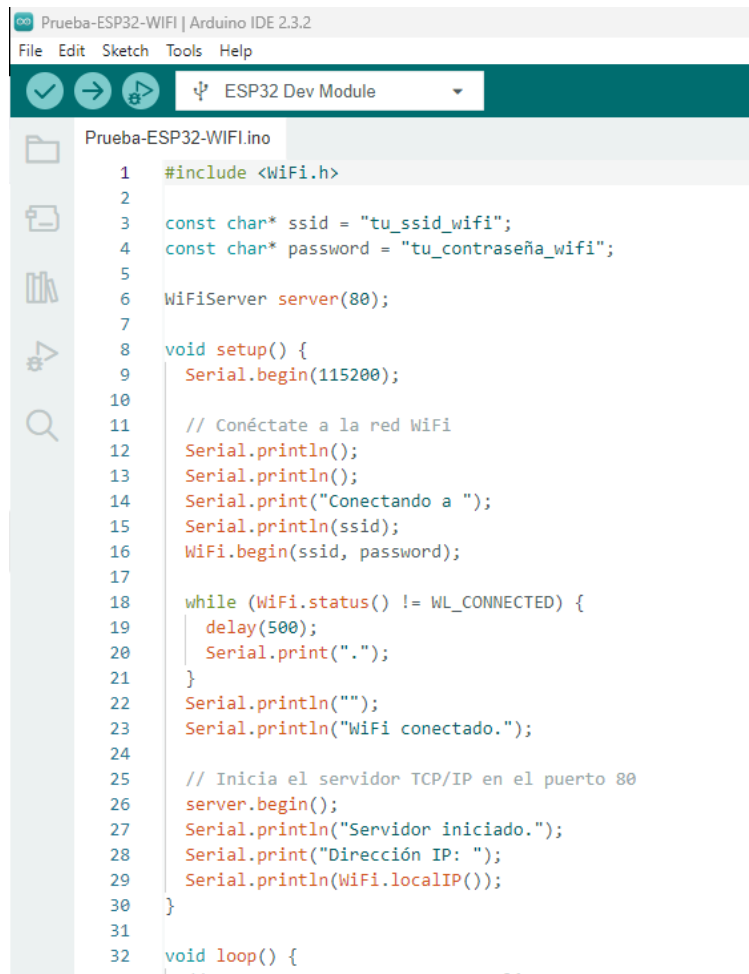
```
1  #include <Wire.h>
2  #include <Adafruit_GFX.h>
3  #include <Adafruit_SSD1306.h>
4
5  #define SCREEN_WIDTH 128
6  #define SCREEN_HEIGHT 64
7  #define OLED_RESET -1
8
9  Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
10
11 void setup() {
12   Serial.begin(9600);
13
14   // Inicializa la comunicación I2C
15   Wire.begin();
16
17   // Inicializa la pantalla OLED
18   if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
19     Serial.println(F("Error al iniciar la pantalla OLED"));
20     for (;;);
21   }
22
23   // Limpiar el buffer de la pantalla OLED
24   display.clearDisplay();
25
26   // Mostrar texto en la pantalla OLED
27   display.setTextSize(1);
28   display.setTextColor(SSD1306_WHITE);
29   display.setCursor(0, 0);
30   display.println("Hola, ESP32!");
31   display.display();
32 }
33
34 void loop() {
35   // No hay necesidad de hacer nada en el bucle, ya que la pantalla OLED no cambia.
36 }
37
```

Figura 3.6 Prueba de comunicación con pantalla OLED

3.2.2 Prueba de integración

Para la integración de los módulos primeramente se partió de un código que uniera la comunicación inalámbrica, el almacenamiento en tarjeta microSD y el sensor MPU-6050. En la Figura 3.7 se puede observar un extracto del código utilizado, donde primero se inicializa la comunicación con el sensor MPU-6050 y el módulo lector de tarjetas microSD, después se conecta a la red Wi-Fi para iniciar un servidor TCP/IP en el ESP-32. En el

bucle principal se leen los datos del acelerómetro y el giroscopio del MPU6050. Para posteriormente guardar los datos en un archivo llamado "datos.txt" en la tarjeta microSD. Se espera la conexión de un cliente a través de Wi-Fi. Cuando un cliente se conecta, lee cualquier solicitud que envíe y posteriormente reenvía los datos del MPU6050 de vuelta al cliente.



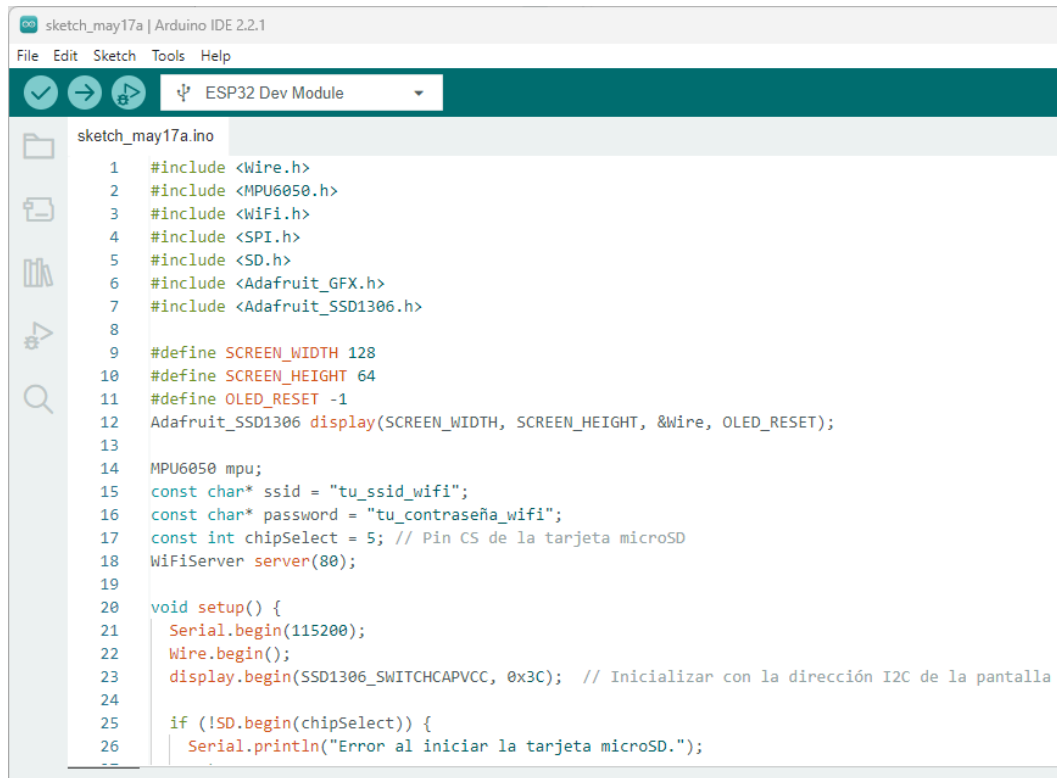
```

1  #include <WiFi.h>
2
3  const char* ssid = "tu_ssid_wifi";
4  const char* password = "tu_contraseña_wifi";
5
6  WiFiServer server(80);
7
8  void setup() {
9      Serial.begin(115200);
10
11     // Conéctate a la red WiFi
12     Serial.println();
13     Serial.println();
14     Serial.print("Conectando a ");
15     Serial.println(ssid);
16     WiFi.begin(ssid, password);
17
18     while (WiFi.status() != WL_CONNECTED) {
19         delay(500);
20         Serial.print(".");
21     }
22     Serial.println("");
23     Serial.println("WiFi conectado.");
24
25     // Inicia el servidor TCP/IP en el puerto 80
26     server.begin();
27     Serial.println("Servidor iniciado.");
28     Serial.print("Dirección IP: ");
29     Serial.println(WiFi.localIP());
30 }
31
32 void loop() {

```

Figura 3.7 Prueba de integración ESP-32, MPU-6050, módulo microSD y Wi-Fi

La segunda prueba de integración requiere de una pantalla para desplegar la información proporcionada por el sensor MPU-6050, la cual solo modifica ligeramente el código para enviar la información de aceleración y velocidad angular como una referencia de que el dispositivo está adquiriendo. Inicialmente la pantalla estaba considerada para configurar parámetros dentro del dispositivo, pero se acotó solo a visualizar datos, ya que se usó la configuración inicial del sensor y los módulos adicionales. En la Figura 3.8 se puede observar un extracto del código desarrollado. Los códigos desarrollados se muestran en el Anexo 1 de este documento.



```
sketch_may17a.ino
1  #include <Wire.h>
2  #include <MPU6050.h>
3  #include <WiFi.h>
4  #include <SPI.h>
5  #include <SD.h>
6  #include <Adafruit_GFX.h>
7  #include <Adafruit_SSD1306.h>
8
9  #define SCREEN_WIDTH 128
10 #define SCREEN_HEIGHT 64
11 #define OLED_RESET -1
12 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
13
14 MPU6050 mpu;
15 const char* ssid = "tu_ssid_wifi";
16 const char* password = "tu_contraseña_wifi";
17 const int chipSelect = 5; // Pin CS de la tarjeta microSD
18 WiFiServer server(80);
19
20 void setup() {
21   Serial.begin(115200);
22   Wire.begin();
23   display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // Inicializar con la dirección I2C de la pantalla
24
25   if (!SD.begin(chipSelect)) {
26     Serial.println("Error al iniciar la tarjeta microSD.");
27   }
28 }
```

Figura 3.8 Prueba de integración ESP-32, MPU-6050, módulo microSD, Wi-Fi y OLED

Finalmente, una vez que ha sido validada la comunicación entre los diferentes módulos se procedió al diseño y la fabricación de un chasis que pueda albergar todos los dispositivos, para lo cual se utilizó la impresión 3D y el diseño asistido por computadora. En la Figura 3.9 se puede observar la integración del prototipo de unidad inercial.

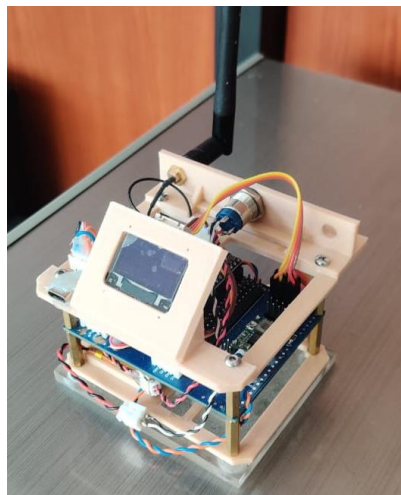


Figura 3.9 chasis con todos los módulos y circuitos electrónicos utilizados

3.2.3 Prueba de aceptación

El criterio de aceptación consiste en que el prototipo satisfaga los requerimientos establecidos anteriormente. En este sentido, una vez desarrollado y construido el prototipo se procedió a llevar a cabo determinadas pruebas de validación. Para ello se utilizó un sistema generador de vibraciones de la marca *Ling Dynamics System* (LDS) que cuenta con una base plana para colocar el espécimen de prueba que será sometido a vibración en una dirección vertical. En estas pruebas, el prototipo fue sometido a aceleraciones en los ejes X, Y y Z, seleccionando tres amplitudes de aceleración distintas: 10 m/s^2 , 15 m/s^2 y 20 m/s^2 . Dado que el vibrador aplica aceleración únicamente en una dirección, fue necesario rotar y fijar el dispositivo para evaluar cada eje de forma independiente.

Dado que la aceleración de la gravedad está siempre presente en la dirección vertical y a fin de obtener la aceleración neta vertical producida por la mesa de vibraciones, se restó el valor correspondiente en el post-procesamiento de los datos. Esto permitió aislar la aceleración aplicada específicamente por la mesa y obtener las mediciones para cada uno de los ejes.

Estas pruebas de vibración son fundamentales para garantizar la robustez y la fiabilidad del prototipo bajo condiciones de aceleración variadas, asegurando que el dispositivo funcione correctamente en diversas aplicaciones y entornos. En la Figura 3.10 se puede observar el dispositivo montado en el sistema generador de vibraciones.



Figura 3.10 Pruebas de aceptación en mesa de vibraciones LDS

4. Resultados

Derivado de las pruebas de laboratorio, donde se evaluó el dispositivo a diferentes amplitudes de aceleración. De acuerdo con las pruebas realizadas en la mesa de vibraciones se obtuvieron resultados de la aceleración medida en cada eje del dispositivo. En este capítulo se presentan algunos de los resultados obtenidos, así como el análisis de estos y el porcentaje de error con respecto a las aceleraciones de referencia. En cuanto a las velocidades angulares al final se detalla algunos resultados, por el alcance de este trabajo no se realizaron pruebas de comparación contra un dispositivo de referencia, por lo cual solo se presenta la evidencia de que el dispositivo es capaz de captar velocidades angulares.

Como sensor de referencia se utilizó un acelerómetro patrón calibrado Brüel & Kjær tipo 4382 con una capacidad de medir aceleración de hasta 2,000 g pico con sensibilidad de 31.19 pC/g. Para las pruebas de validación de aceleración se realizaron 3 mediciones por cada eje. A los datos obtenidos se le aplicó un filtraje por promedio móvil central de 10 datos. Posterior al filtraje se aplicó un promedio a los picos positivos y negativos más altos, esto con el fin de identificar la aceleración positiva y negativa promedio. Por último, se calculó el error absoluto y relativo con respecto al valor de referencia.

Para entender mejor los errores relativos, se pueden calcular usando la siguiente fórmula:

$$Error\ relativo\ (\%) = \left(\frac{|Acel_{prom} - Acel_{ref}|}{Acel_{ref}} \right) * 100$$

Donde:

$Acel_{ref}$ es la aceleración de referencia.

$Acel_{prom}$ es la aceleración promedio de los picos positivos y negativos de aceleración medida por el IMU.

A continuación, se presentan los resultados de las aceleraciones lineales y de las velocidades angulares medidas con el prototipo de IMU.

4.1.1 Resultados de aceleración en eje X

En la Tabla 4.1 se observa un resumen de las aceleraciones resultantes de evaluar el prototipo de unidad inercial en una mesa de vibraciones, donde se puede observar que para los valores de referencia negativos (-10, -15 y -20 m/s²) el error relativo es mayor que para las aceleraciones de referencia positivas. El error más pequeño se puede observar en la aceleración de 20 m/s² tanto para los picos positivos como los negativos, esto puede deberse a que el sensor estaba configurado en su máxima resolución de +/- 16 g lo que permitió lograr mayor precisión en mediciones cercanas a este rango.

Tabla 4.1 Resultados de errores relativos de aceleración en el eje X

Aceleración de referencia, [m/s ²]	Aceleración promedio, [m/s ²]		Error relativo, [%]	
	Positiva	Negativa	Positivo	Negativo
+/-10	9.51	-9.18	4.90	8.20
+/-15	14.37	-14.24	4.20	5.07
+/-20	19.46	-19.33	2.70	3.35

Los datos indican que, para aceleraciones de referencia de 10, 15 y 20 m/s², los errores relativos en las mediciones de aceleración promedio negativa son mayores que los errores relativos en las mediciones de aceleración promedio positiva. Se puede observar que el valor de 2.70% de error relativo se concentra en la aceleración más alta de 20 m/s².

4.1.2 Resultados de aceleración en eje Y

En la tabla 4.2 se presenta los resultados de las aceleraciones medidas en el eje Y con el prototipo de unidad inercial. En la cual se puede observar un comportamiento similar al del eje X, donde la mayor precisión se logró analizando los picos positivos de aceleración, mientras que para los picos negativos se mantiene un error relativo mayor, de igual forma se observa que a mayor aceleración la lectura del sensor se aproxima al valor de referencia. El error más pequeño para este eje fue de 2.90% en el valor de referencia -20 m/s², donde a diferencia del eje X este se encuentra del lado negativo, aunque también el error más grande de 8.80 % se encuentra en los picos negativos, para este caso el error más pequeño se mantiene por debajo del 3.5 % en los dos valores de referencia positivo y negativo de -20 m/s². Al igual que en el eje X donde el error por debajo de 3.5 % únicamente se encuentra en el valor de referencia más alto de +/-20 m/s². De igual forma la tendencia del error relativo más grande se encuentra en el valor de referencia más pequeño.

Tabla 4.2 Resultados de errores relativos de aceleración en el eje Y

Aceleración de referencia, [m/s ²]	Aceleración promedio, [m/s ²]		Error relativo, [%]	
	Positiva	Negativa	Positivo	Negativo
+/-10	9.29	-9.12	7.10	8.80
+/-15	14.41	-14.37	3.93	4.20
+/-20	19.38	-19.42	3.10	2.90

4.1.3 Resultados de aceleración en eje Z

En la Tabla 4.3 Se observan los resultados de las aceleraciones medidas en el eje Z, donde los errores relativos se mantiene por debajo del 10% para las aceleraciones positivas y negativas, donde el menor error se presenta para las amplitudes más altas en aceleración, siendo la aceleración positiva la de menor porcentaje de error, con un error relativo de 3.40% para la aceleración de 20 m/s² y de 3.60 % para la parte negativa. Nuevamente se observa una tendencia con respecto a los otros ejes evaluados, donde la precisión del dispositivo mejora a amplitudes de aceleración altas, lo cual puede ser por la misma configuración utilizada para adquirir la señal del sensor, faltaría realizar pruebas con las configuraciones de +/- 8 g, +/- 4 g y +/- 2 g, para ver si la precisión mejora cambiando la configuración de amplitud de aceleración en el sensor.

Tabla 4.3 Resultados de errores relativos de aceleración en el eje Z

Aceleración de referencia, [m/s ²]	Aceleración promedio, [m/s ²]		Error relativo, [%]	
	Positiva	Negativa	Positivo	Negativo
+/-10	9.17	-9.05	8.30	9.50
+/-15	14.11	-14.31	5.93	4.60
+/-20	19.32	-19.28	3.40	3.60

4.1.4 Resultados de velocidades angulares

Para las velocidades angulares se realizó una pequeña prueba de evasión de obstáculos para realizar una maniobra en zigzag, esto con el fin de ver si el sensor respondía al estímulo. Con esta maniobra se logró obtener información de la velocidad angular del vehículo de prueba. Para la evaluación de velocidades angulares únicamente se adquirió información proveniente del dispositivo, ya que para este caso no se contó con un dispositivo que permitiera comparar la velocidad angular del sensor. Para el procesamiento de la velocidad angular se utilizó un promedio móvil central de 13 datos, ya que la señal original se encontraba muy ruidosa. Por

lo tanto, se selecciona 13 datos para ser promediados y así ir limpiando la señal. Cabe señalar que la selección de los 13 datos fue a criterio propio ya que para una mejor visualización de las señales de velocidad angular se optó por este procedimiento, ya que para casos de comparación se tendría que tomar en cuenta un filtro que no tiene en gran medida los picos de velocidad angular o aplicar el mismo filtro, tanto a la señal de referencia como a la evaluada.

En la Figura 4.1 se puede observar la gráfica de la velocidad angular de cabeceo, en la cual se puede observar unos pequeños picos derivados del frenado del vehículo para dar una vuelta en “U” donde se registraron valores cercanos a 10 °/s. En esta señal no se alcanza a percibir una variación significativa debido a la naturaleza de la prueba, ya que esta señal reaccionaría más a movimiento de cabeceo del vehículo, que son más comunes en frenados bruscos.

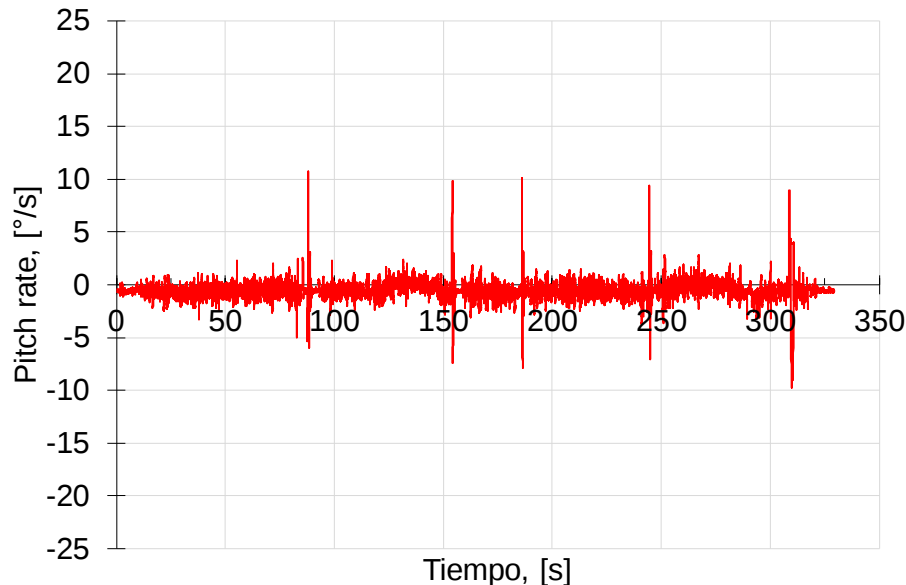


Figura 4.1 Velocidad angular cabeceo obtenida del prototipo de IMU

En la Figura 4.2 se observa la gráfica con los valores obtenidos de la velocidad angular de balanceo, en esta gráfica se observa que solo hubo 3 pequeños eventos de roll donde se midieron picos entre 5 y 10 °/s, lo que nos indica que, si hubo balanceo o inclinación lateral del vehículo sin llegar al vuelco, ya que la velocidad angular fue muy pequeña y de corta duración. Por la naturaleza de la prueba tanto la Figura 4.1 y 4.2 no muestran gran variación en amplitud, pero si dan información de los posibles movimientos que pudieron provocar estos picos de velocidad angular.

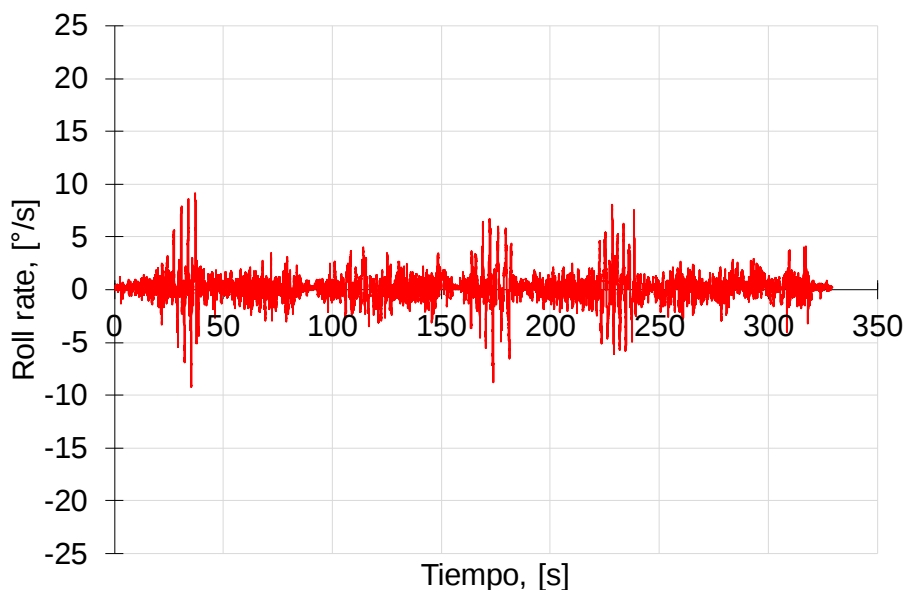


Figura 4.2 Velocidad angular de balanceo obtenida del prototipo de IMU

En la Figura 4.3 se presenta la velocidad angular de coleo obtenida, donde a diferencia de las 2 anteriores puede notarse más variación, esto debido al tipo de maniobra que se realizó. Es posible observar el coleo generado en las pruebas de evasión de obstáculos, donde se observa velocidades angulares por arriba de 10 $^{\circ}/s$ en la primera parte y una vuelta brusca al final con velocidad angular de 25 $^{\circ}/s$.

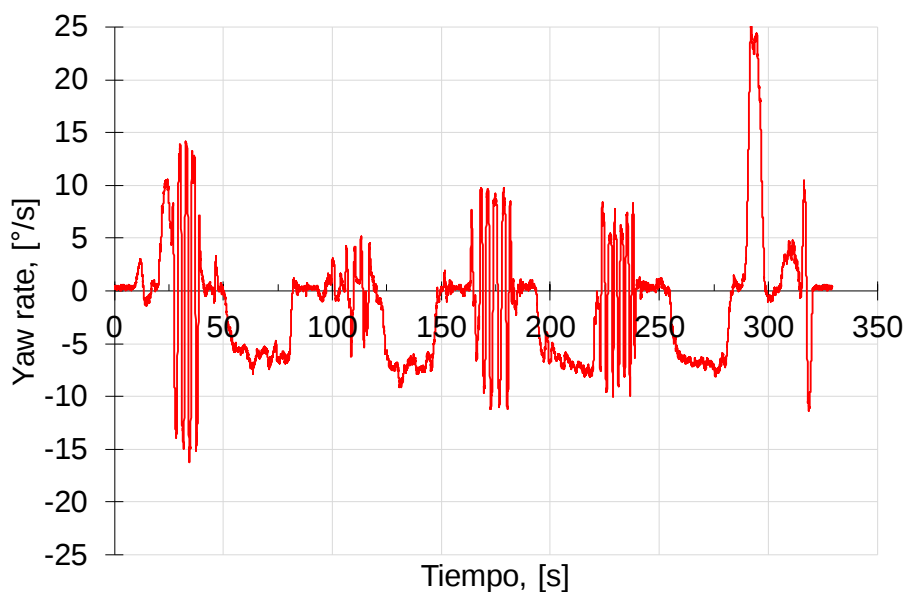


Figura 4.3 Velocidad angular de coleo obtenida del prototipo de IMU

De las 3 gráficas es donde podemos observar mayor variación, lo es importante para complementar el análisis dinámico del vehículo, a simple vista sin conocer los antecedentes de la prueba, se puede determinar solo

viendo el grafico que existe un movimiento de coleo continuo, lo cual permite intuir que se trataría de una evasión de obstáculos en zigzag y que al final de cada evasión el vehículo dio una vuelta y volvió a hacer la misma maniobra, claro no se conoce si fue una vuelta en “U” o solo una evasión más, pero por el sensor se sabe que tuvo una velocidad angular de coleo y esta se produce comúnmente cuando das una vuelta o realizas una maniobra de evasión.

Conclusiones y trabajo futuro

El desarrollo de prototipos de Unidades de Medición Inercial (IMU) representa un paso significativo hacia la innovación tecnológica y la competitividad global de México. Estos dispositivos son esenciales para una variedad de aplicaciones en sectores clave como la industria automotriz, la aeronáutica, la robótica y la electrónica de consumo, entre otras. Derivado del trabajo realizado y los resultados obtenidos se concluye lo siguiente:

- El dispositivo fue capaz de medir aceleración en X , Y y Z con un porcentaje de error relativo por debajo del 10%, lo cual es bastante bueno para ser un sensor con materiales realmente económicos. Por otro lado, para pruebas de dinámica vehicular si se requiere una precisión mayor con un error relativo inferior al 2%, ya que para la aplicación de evaluar la seguridad de un vehículo una gran parte de sensores comerciales presenta una alta precisión en los parámetros de aceleración, lo cual va de mano con su costo de fabricación ya que se incrementa por encima de los 500 dólares americanos, mientras que el prototipo creado en el laboratorio ronda entre 50 a 60 dólares.
- El dispositivo tiene la capacidad de medir velocidad angular en 3 direcciones (pitch, roll y yaw).
- La frecuencia de muestreo llegó hasta 100 Hz con un porcentaje de error de ± 10 milisegundos esto debido a retrasos cortos por el tiempo de escritural a la memoria interna.
- La tarjeta de adquisición ESP32, demostró ser útil para transferir datos de forma inalámbrica a través de una red Wi-Fi, lo facilita la visualización remota de la información adquirida.
- La autonomía del dispositivo se determinó de 6 horas, lo cual es suficiente para pruebas de evaluación dinámica en vehículos, además de contar con la posibilidad de desmontar la batería y sustituirla por otra cargada, lo cual reduce los tiempos de espera para utilizar el dispositivo.

Como trabajo futuro se plantea una actualización del dispositivo, en cuanto al diseño del chasis y optimización del proceso de filtraje de la tarjeta de adquisición, para esta primera aproximación solo se utilizó la configuración de fábrica para la adquisición de la información, por lo cual

se propone que se implemente, a través de la programación, un sistema de filtro especializado que permita tanto mejorar la precisión del dispositivo como simplificar el análisis de datos.

También es importante señalar los problemas que se tuvieron en la sincronización de la estampa de tiempo con respecto a la escritura de la información en la tarjeta microSD, por lo cual se necesita desarrollar un mejor método de escritura y lectura de la información, por lo que se sugiere verificar si es posible realizar una estructura maestro-esclavo que no afecte los tiempos de adquisición y escritura local de la información, así como el envío remoto de paquetes de datos.

En cuanto a la estructura se recomienda cambiar el plástico PETG utilizado por materiales más rígidos como lo es el aluminio o acero inoxidable, esto debido a que, al ser sometido a aceleraciones o impactos demandantes, la estructura y los conectores pueden ser susceptibles a ruptura o desconexión, respectivamente.

En resumen, el prototipo de IMU fue capaz de medir aceleraciones y velocidades angulares a una frecuencia de adquisición de 100 Hz, el dispositivo es perfectible, pero cumple con los requerimientos establecidos para este primer prototipo. No obstante, quedan desafíos como, mejorar la precisión de las mediciones, así como hacer frente a los retos planteados como trabajo futuro; por lo cual pudiera planificarse una etapa de mejora para este sistema.

Bibliografía

- [1] Tipler, P. (2004). Physics for Scientists and Engineers: Mechanics, Oscillations and Waves, Thermodynamics (5th ed.). ISBN 0-7167-0809-4.
- [2] Serway, R.; Jewett, W. (2004). Physics for Scientists and Engineers (6th ed.). ISBN 0-534-40842-7.
- [3] Caceres, K. (2023). La historia de la aceleración en física. [Consulta en línea], <https://brainly.lat/tarea/70601981>
- [4] Omega engineering. (2023). Acelerómetro. [Consulta en línea], <https://es.omega.com/prodinfo/acelerometro.html>
- [5] Morelle, R. (2014). Tecnología de la Segunda Guerra Mundial por si falla la del siglo XXI - BBC News Mundo. [Consulta en línea], https://www.bbc.com/mundo/noticias/2014/11/141106_tecnologia_bar_cos_gps_eloran_finde_az
- [6] Maloy, G. (2023). ¿Qué es un sistema de navegación inercial? [Consulta en línea], <https://dewesoft.com/es/blog/que-es-un-sistema-de-navegacion-inercial>
- [7] Blog Mecánicos. (2022). ¿Cuál es la función de los sensores de guiñada y de aceleración? [Consulta en línea], <https://www.blogmecanicos.com/2022/10/cual-es-la-funcion-de-los-sensores-de.html>
- [8] Prometec. (2023). Los sistemas de medida inercial. [Consulta en línea], <https://www.prometec.net/imu-mpu6050/>
- [9] Fernandez, J. (2023). Lanzamiento Vertical. [Consulta en línea], <https://www.fisicalab.com/apartado/lanzamiento-vertical>
- [10] Naylamp mechatronics. (2016). MPU6050, Acelerómetro y Giroscopio. [Consulta en línea], https://naylampmechatronics.com/blog/45_tutorial-mpu6050-acelerometro-y-giroscopio.html
- [11] Olmo, M. (2023). Rotational Quantities. [Consulta en línea], <http://hyperphysics.gsu.edu/hbasees/rotq.html>
- [12] Brüel & Kjær. (2023). Guía práctica de acelerómetros piezoeléctricos [Consulta en línea],

- <https://www.bksv.com/es/knowledge/blog/vibration/piezoelectric-accelerometers>
- [13] Sensing. (2023) Acelerómetros piezorresistivos. [Consulta en línea], <https://sensores-de-medida.es/medicion/sensores-y-transductores/acelerometros-sensores-de-aceleracion/acelerometros-piezoresistivos/>
- [14] Aroeira, C. (2023). Tipos de acelerómetros. [Consulta en línea], <https://www.dmc.pt/es/tipos-de-acelerometros/>
- [15] Electronic components. (2023) ¿Cómo funciona y qué hace el acelerómetro? [Consulta en línea], from <https://www.tme.com/mx/es/news/library-articles/page/22568/Como funciona-y-que-hace-el-acelerometro/>
- [16] Hibbeler, R. (2009). Mecánica de la Ingeniería. ISBN 978-0-13-607791-6.
- [17] Tipler, P. (2000). Física para la ciencia y la tecnología (2 volúmenes). ISBN 84-291-4382-3.
- [18] Rivera, J. (2022). ESP32 NODEMCU DEVKIT C V4 PINES Y CARACTERISTICA. [Consulta en línea], <https://pasionelectronica.com/esp32-caracteristicas-y-pines/>
- [19] Altronics. (2023). Acelerómetro y giroscopio MPU-6050. [Consulta en línea], <https://altronics.cl/acelerometro-giroscopio-mpu6050>
- [20] InvenSense. (2013). MPU-6000 and MPU-6050 Product Specification Revision 3.4. [Archivo PDF], https://altronics.cl/uploads/MPU-6050_DataSheet_V3%204.pdf
- [21] NAYLAMP MECHATRONICS. (2023). Módulo lector de memoria Micro SD Card. [Consulta en línea], <https://naylampmechatronics.com/almacenamiento/104-modulo-lector-de-memoria-micro-sd-card-lvc125a.html>
- [22] dIGIKEY. (2019). SPI simplifica la comunicación con dispositivos [Consulta en línea], <https://www.digikey.com.mx/es/articles/why-how-to-use-serial-peripheral-interface-simplify-connections-between-multiple-devices>
- [23] HETPRO. [2022]. I2C - Puerto, Introducción, trama y protocolo. [Consulta en línea], <https://hetpro-store.com/TUTORIALES/i2c/>

Anexo 1 Códigos de pruebas modulares

Código para módulo MPU-6050

```
#include <Wire.h>
#include <MPU6050.h>

MPU6050 mpu;

void setup() {
  Serial.begin(115200);

  Wire.begin();
  mpu.initialize();

  // Verifica si el sensor está conectado correctamente
  if (!mpu.testConnection()) {
    Serial.println("Error al iniciar el sensor MPU6050. Verifica la conexión.");
    while (true);
  }
}

void loop() {
  // Lee los datos del acelerómetro y giroscopio
  int16_t ax, ay, az;
  int16_t gx, gy, gz;
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

  // Imprime los valores leídos
  Serial.print("Acelerómetro: ");
  Serial.print("X=");
  Serial.print(ax);
  Serial.print(" Y=");
  Serial.print(ay);
  Serial.print(" Z=");
  Serial.println(az);

  Serial.print("Giroscopio: ");
  Serial.print("X=");
```

```
Serial.print(gx);
Serial.print(" Y=");
Serial.print(gy);
Serial.print(" Z=");
Serial.println(gz);

delay(100); // Espera un momento antes de leer de nuevo
}
```

Código para módulo MicroSD

```
#include <SPI.h>
#include <SD.h>

File dataFile;
const int chipSelect = 5; // El pin CS de la tarjeta microSD puede ser
cualquier pin digital

void setup() {
  Serial.begin(115200);

  // Inicializa la comunicación con la tarjeta microSD
  if (!SD.begin(chipSelect)) {
    Serial.println("Error al inicializar la tarjeta microSD.");
    return;
  }

  Serial.println("Tarjeta microSD inicializada correctamente.");

  // Crea un archivo llamado "data.txt" en la raíz de la tarjeta microSD
  dataFile = SD.open("data.txt", FILE_WRITE);

  // Si el archivo se abrió correctamente, escribe un encabezado
  if (dataFile) {
    dataFile.println("Datos guardados desde ESP32:");
    dataFile.close();
    Serial.println("Archivo 'data.txt' creado correctamente.");
  } else {
    Serial.println("Error al abrir el archivo 'data.txt'.");
  }
}

void loop() {
```

```
// Genera algunos datos de ejemplo
int sensorValue = analogRead(A0);

// Abre el archivo en modo de escritura al final del mismo
dataFile = SD.open("data.txt", FILE_WRITE);

// Si el archivo se abrió correctamente, escribe los datos
if (dataFile) {
  dataFile.print("Valor del sensor: ");
  dataFile.println(sensorValue);
  dataFile.close();
  Serial.println("Datos guardados correctamente.");
} else {
  Serial.println("Error al abrir el archivo 'data.txt'.");
}

delay(1000); // Espera un momento antes de escribir más datos
}
```

Código para módulo OLED I2C

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);

void setup() {
  Serial.begin(9600);

  // Inicializa la comunicación I2C
  Wire.begin();

  // Inicializa la pantalla OLED
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("Error al iniciar la pantalla OLED"));
    for (;;)
  }
}
```

```
// Limpiar el buffer de la pantalla OLED
display.clearDisplay();

// Mostrar texto en la pantalla OLED
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.setCursor(0, 0);
display.println("Hola, ESP32!");
display.display();
}

void loop() {
  // No hay necesidad de hacer nada en el bucle, ya que la pantalla OLED
  no cambia.
}
```

Código para prueba Wi-Fi con ESP-32

```
#include <WiFi.h>

const char* ssid = "tu_ssid_wifi";
const char* password = "tu_contraseña_wifi";

WiFiServer server(80);

void setup() {
  Serial.begin(115200);

  // Conéctate a la red WiFi
  Serial.println();
  Serial.println();
  Serial.print("Conectando a ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi conectado.");
}
```

```
// Inicia el servidor TCP/IP en el puerto 80
server.begin();
Serial.println("Servidor iniciado.");
Serial.print("Dirección IP: ");
Serial.println(WiFi.localIP());
}

void loop() {
  // Espera a que se conecte un cliente
  WiFiClient client = server.available();
  if (!client) {
    return;
  }

  // Espera a que el cliente envíe datos
  while (!client.available()) {
    delay(1);
  }

  // Lee los datos del cliente y envíalos de vuelta
  String request = client.readStringUntil('\r');
  Serial.println(request);
  client.flush();

  // Envía una respuesta al cliente
  client.println("HTTP/1.1 200 OK");
  client.println("Content-type:text/html");
  client.println();
  client.println("<html>");
  client.println("<head><title>ESP32 Server</title></head>");
  client.println("<body>");
  client.println("<h1>ESP32 Server</h1>");
  client.println("<p>Hola desde ESP32!</p>");
  client.println("</body>");
  client.println("</html>");

  delay(1);
  Serial.println("Cliente desconectado.");
}
```

Código de primera prueba de integración

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

```
#include <MPU6050.h>
#include <WiFi.h>
#include <SPI.h>
#include <SD.h>

const char* ssid = "tu_ssid_wifi";
const char* password = "tu_contraseña_wifi";

const int chipSelect = 5; // Pin CS de la tarjeta microSD

MPU6050 mpu;

WiFiServer server(80);

void setup() {
  Serial.begin(115200);

  // Inicializa la comunicación con el sensor MPU6050
  Wire.begin();
  mpu.initialize();

  // Inicializa la comunicación con la tarjeta microSD
  if (!SD.begin(chipSelect)) {
    Serial.println("Error al inicializar la tarjeta microSD.");
    return;
  }

  // Inicia la conexión WiFi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi conectado.");

  // Inicia el servidor TCP/IP en el puerto 80
  server.begin();
  Serial.println("Servidor iniciado.");
  Serial.print("Dirección IP: ");
  Serial.println(WiFi.localIP());
}

void loop() {
```

```
// Lee los datos del acelerómetro y giroscopio del MPU6050
int16_t ax, ay, az;
int16_t gx, gy, gz;
mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

// Guarda los datos en la tarjeta microSD
File dataFile = SD.open("datos.txt", FILE_WRITE);
if (dataFile) {
  dataFile.print("Acelerómetro: ");
  dataFile.print(ax);
  dataFile.print(" ");
  dataFile.print(ay);
  dataFile.print(" ");
  dataFile.print(az);
  dataFile.print(" Giroscopio: ");
  dataFile.print(gx);
  dataFile.print(" ");
  dataFile.print(gy);
  dataFile.print(" ");
  dataFile.println(gz);
  dataFile.close();
  Serial.println("Datos guardados en la tarjeta microSD.");
} else {
  Serial.println("Error al abrir el archivo 'datos.txt'.");
}

// Espera a que se conecte un cliente
WiFiClient client = server.available();
if (client) {
  // Espera a que el cliente envíe datos
  while (client.connected()) {
    if (client.available()) {
      // Lee los datos del cliente y envíalos de vuelta
      String request = client.readStringUntil('\r');
      Serial.println(request);
      client.flush();

      // Envía los datos del MPU6050 al cliente
      client.println("Acelerómetro: " + String(ax) + " " + String(ay) + " " +
String(az) + " Giroscopio: " + String(gx) + " " + String(gy) + " " + String(gz));
    }
  }
  // Cierra la conexión con el cliente
  client.stop();
}
```

```
    Serial.println("Cliente desconectado.");  
}  
  
delay(1000); // Espera un segundo antes de leer de nuevo  
}
```




COMUNICACIONES

SECRETARÍA DE INFRAESTRUCTURA, COMUNICACIONES Y TRANSPORTES



Km 12+000 Carretera Estatal 431 "El Colorado-Galindo"
San Fandila, Pedro Escobedo
C.P. 76703
Querétaro, México
Tel: +52 442 216 97 77 ext. 2610

publicaciones@imt.mx

<http://www.imt.mx/>