24 de septiembre de 2024

|  |  |  |  |
| --- | --- | --- | --- |
| **Informe inicial BALANCEADOR DE**  **2 RUEDAS** | |  | A3 |
| Facultad de Informática UNLP  Sistemas de Tiempo Real | |  |  |
|  | |  |  |
|  | |  |  |
|  |  |  | |
| * Melina Caciani Toniolo   melicaciani@gmail.com | |  | 02866/1 |
| * Joaquín Chanquía   joaquin.chanquia@alu.ing.unlp.edu.ar | |  | 02887/7 |
| * Mateo Emmanuel Larsen   larsenmateo.ml@gmail.com | |  | 02993/7 |
| * Gabriel Ollier   gabyollier@hotmail.com | |  | 02958/4 |
| * Franco Niderhaus   franconiderhaus@gmail.com   * Bruno Zanetti   bzanetti09@gmail.com | |  | 02976/6  02975/5 |

**ÍNDICE**

[**INTRODUCCIÓN 3**](#_Toc177601057)

[**OBJETIVOS 3**](#_Toc177601058)

[**IDENTIFICACIÓN DE PARTES 4**](#_Toc177601059)

[**Hardware 4**](#_Toc177601060)

[**Análisis Eléctrico 4**](#_Toc177601061)

[**Tecnologías de Software 4**](#_Toc177601062)

[**ESQUEMA GRÁFICO 5**](#_Toc177601063)

[**BIBLIOGRAFÍA 5**](#_Toc177601064)

# INTRODUCCIÓN

Se realizará el proyecto conocido como A3 *Balanceador de 2 ruedas.* Este proyecto busca desarrollar un sistema de balanceo de un vehículo de dos ruedas sobre una plataforma ESP32, logrando mantener el equilibrio de forma autónoma. Se debe trabajar en implementar un sistema de control robusto que permita corregir las desviaciones del vehículo y mantenerlo en posición vertical, a partir de un algoritmo de PID.

Además, se desarrollará una simulación que permita representar cómo funcionaría el balanceador ante determinadas situaciones.

# OBJETIVOS

* **Desarrollar un sistema de balanceo autónomo**: Implementar un sistema de control basado en PID que permita al vehículo de dos ruedas mantener el equilibrio de manera autónoma en tiempo real.
* **Calibración de sensores:** Los sensores deben calibrarse correctamente para obtener lecturas precisas.
* **Control en bucle cerrado**: Desarrollar un sistema de feedback continuo que ajuste la potencia de los motores en función de las lecturas de los sensores, garantizando una respuesta precisa ante perturbaciones externas.
* **Filtrado de ruido y corrección de errores:** Implementar técnicas de filtrado en las señales de los sensores para reducir el impacto del ruido y mejorar la precisión de las mediciones.
* **Evaluar la robustez y el rendimiento en condiciones reales**: Probar y ajustar el sistema bajo diferentes condiciones para asegurar que mantenga el equilibrio de manera estable, evaluando su capacidad para adaptarse a cambios en el entorno o perturbaciones externas.
* **Simulación computacional**: Implementar una simulación del sistema de control PID y del comportamiento del vehículo en un entorno virtual. La simulación debe reflejar el funcionamiento físico del sistema, permitiendo evaluar su desempeño antes de la implementación física.
* **Control mediante aplicación**: Implementar una aplicación web para controlar el manejo del vehículo.
* **Seguridad**: Debe garantizar la seguridad del entorno y las personas alrededor, evitando movimientos bruscos o incontrolados.

# IDENTIFICACIÓN DE PARTES

**Hardware**

La siguiente tabla enumera los elementos necesarios para llevar a cabo este proyecto, obviando elementos como cables, tornillos y materiales que conforman el chasis del robot.

|  |  |
| --- | --- |
| **ELEMENTO** | **COSTO APROXIMADO** |
| Placa ESP32 | $10.000 |
| 2 Motores DC con encoders y ruedas incorporadas | $25.000 c/u |
| Sensor MPU6050 (acelerómetro y giroscopio) | $3800 |
| Driver L298N | $3200 |
| 3 pilas de litio 3,7V | $3250 c/u |

**Análisis Eléctrico**

Se extrajeron datos de las datasheet de los componentes del circuito para analizar su consumo eléctrico.

|  |  |  |
| --- | --- | --- |
| **ELEMENTO** | **VOLTAJE** | **CORRIENTE** |
| Placa ESP32 | 3.3V | 180 Ma |
| Motor DC con encoder (2) | 12V | 2ª |
| Sensor MPU6050 | 3.3V | 3.9Ma |
| L298N | 4.5 V a 7V | 13mA a 70mA |
| **TOTAL** |  |  |

**Tecnologías de Software**

Para el desarrollo de este proyecto se utilizarán varias plataformas y lenguajes. Sus elecciones están basadas en experiencias, recomendaciones y/o conveniencia.

* El código del microcontrolador se realizará en lenguaje *C*.
* La programación de la aplicación web se llevará a cabo con *HTML* y *CSS.*
* Para realizar la simulación, utilizaremos la plataforma *Fritzing*.

**Protocolos de conexión**

* I2C: Se utilizará el protocolo I2C para enviar datos al microcontrolador desde el sensor MPU acelerómetro y giroscopio. I2C es ideal para conectar dispositivos al microcontrolador con solo dos cables (SDA para datos, SCL para reloj). Esto simplifica las conexiones.
* HTTP: será utilizado para la comunicación entre la aplicación web y el ESP32, ya que desde la app se enviarán comandos al robot. El uso de solicitudes HTTP es estándar para la comunicación remota y se puede integrar fácilmente con el ESP32.

# ESQUEMA GRÁFICO

Diagrama

Descripción generada automáticamente

# BIBLIOGRAFÍA

**Videos útiles de proyectos similares**

* <https://www.youtube.com/watch?v=tZynIj1StpM&t=1278s>
* <https://www.youtube.com/watch?v=_tfg8Zx5Gfk&t=6s>

<https://fritzing.org/>

**Costos de los elementos de Hardware:**

* Placa ESP32: <https://www.mercadolibre.com.ar/nodemcu-esp32-wifi-bluetooth-42-iot-wroom-esp32s-38-pines/p/MLA34891403#polycard_client=search-nordic&searchVariation=MLA34891403&position=2&search_layout=grid&type=product&tracking_id=2e27204a-56fb-4622-8421-06dcc6e65aa5&wid=MLA1418554861&sid=search>
* Motor DC con encoder: <https://www.ebay.com/itm/235419100682>
* Sensor MPU6050: <https://www.mercadolibre.com.ar/mpu6050-gy-521-modulo-de-sensor-de-acelerometro-y-giroscopio/p/MLA32941833#polycard_client=search-nordic&searchVariation=MLA32941833&position=1&search_layout=grid&type=product&tracking_id=f9e9e1ff-1ef4-4388-8b44-4fb0b4623559&wid=MLA1713095290&sid=search>
* Baterías de litio: <https://es.aliexpress.com/item/32693679055.html?aff_fcid=733cb6f5280e4f9b88fc1a3f3f5bb07b-1726775505073-04905-_d8cDFDa&aff_fsk=_d8cDFDa&aff_platform=portals-tool&sk=_d8cDFDa&aff_trace_key=733cb6f5280e4f9b88fc1a3f3f5bb07b-1726775505073-04905-_d8cDFDa&terminal_id=9ca2a540130342abbd9ae72dbe4017ac&afSmartRedirect=y&gatewayAdapt=glo2esp>
* L298N: <https://articulo.mercadolibre.com.ar/MLA-680725674-doble-puente-h-driver-l298n-motor-dc-desarrollo-arm-avr-l298-_JM#polycard_client=search-nordic&position=3&search_layout=grid&type=item&tracking_id=48c4a4ad-9809-47f4-b76d-9fc19f544448>b

**Especificaciones Técnicas**

* ESP32: <https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf>
* MPU6050: <https://www.alldatasheet.es/datasheet-pdf/view/517744/ETC1/MPU-6050.html>
* L298N: <https://www.alldatasheet.com/datasheet-pdf/download/22440/STMICROELECTRONICS/L298N.html>
* Motor DC: <https://naylampmechatronics.com/motores-dc/616-motor-dc-jga25-370-12v350rpm-con-encoder.html>