

Laboratorio transmisión y recepción de datos de la stm y curva del motor

Juan Sebastián Herrán Páez & Daniel Jimenez

I. RESUMEN

En este informe se presenta el diagrama de flujo del proyecto completo de stm32cubeIDE, teniendo nuevas funciones para enviar y recibir datos, así mismo la curva generada en python que representa las RPMs en el tiempo desde que se ejecuta el programa en python y los diagramas de flujo de python y qt para la app.

II. MATERIALES

- **STM32F411CEU6:** Microcontrolador base utilizado para el laboratorio.
- **Motoreductor DC:** Utilizado para ensamblar al cd y generar las RPM.
- **Placa de Prototipado:** Placa de circuito impreso para realizar conexiones temporales entre los componentes electrónicos.
- **Sensores de herradura:** Utilizados para detectar los pulsos y desarrollar los respectivos calculos a partir del cd.
- **Pantalla OLED:** Pantalla programable para mostrar los datos importantes y visualizar los cambios.
- **Cables de Conexión:** Cables eléctricos para realizar conexiones entre los diferentes componentes del circuito.
- **Fuente de Alimentación:** Fuente de alimentación de corriente continua para alimentar el motor.
- **Python:** Programa de alto nivel utilizado para generar la gráfica del motor.
- **Qt:** Programa para desarrollar la aplicación y generar la gráfica deseada.

III. PROCEDIMIENTO DEL LABORATORIO

1. Cálculo de RPM sin filtrar y RPM filtrada

En el código principal se implementa un filtro digital FIR, esto con el fin de suavizar la toma de datos resultado por el

cálculo de las rpm de la rueda, se muestra dos gráficas en excel con los datos tomados por la transmisión por USB, en la primera mostrando los datos sin filtrar y la segunda con los datos filtrados.

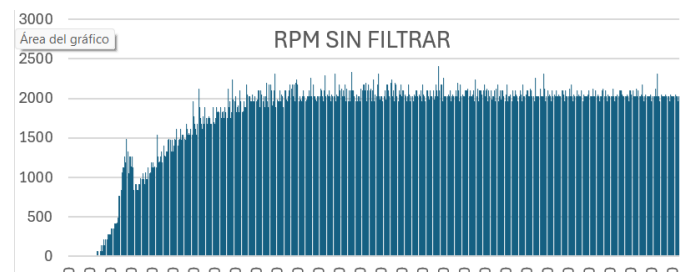


Fig. 1: RPM sin filtrar

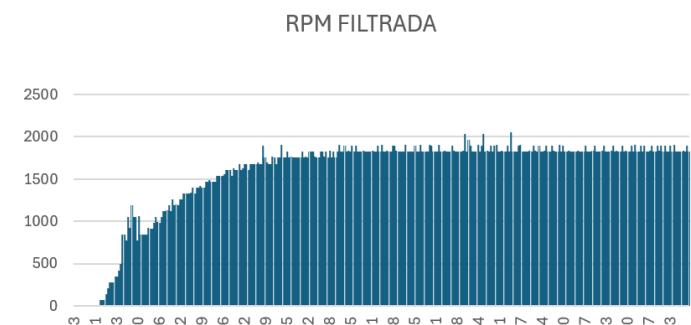


Fig. 2: RPM filtradas

2. Curva de respuesta en voltajes de 1V a 6V

Se realiza la siguiente gráfica de RPM vs tiempo con la función implementada en el proyecto de stm32cubeIDE para transmitir el PWM a la salida del motor, el filtro digital FIR y los datos registrados en hercules para obtener la gráfica en python de 1V a 6V, siendo 6V el voltaje máximo que entrega la fuente.

3. Gráfica de RPM vs voltios

Con base en la gráfica obtenida en python (Fig. 1) se realiza la gráfica de RPM vs voltios (Fig. 2), la cual tiene una recta con base en el promedio de RPM y el ajuste para que la recta sea lineal.

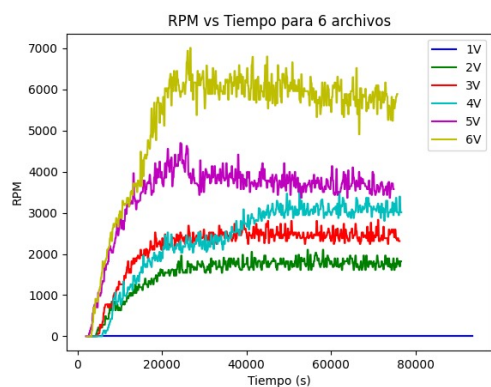


Fig. 3: Curvas RPM vs Tiempo de 1V a 6V

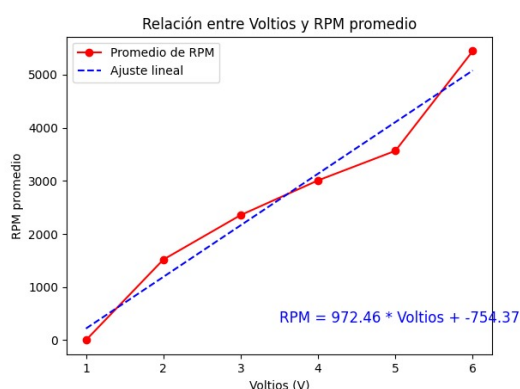


Fig. 4: RPM vs voltios, ecuación de la recta

4. Gráfica en Qt de la curva de las RPM vs tiempo con el boton de rpm deseada añadido

Luego se modificó la función recibirComandoUSB para recibir las RPM que se desean mediante el boton de Qt, utilizando la ecuación de la recta obtenida anteriormente.

A continuación se presenta la gráfica generada en Qt (Fig. 3).

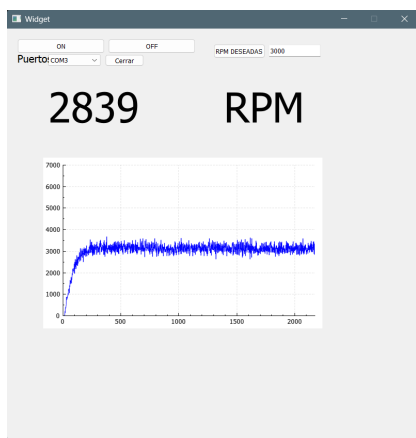


Fig. 5: Widget de Qt, visualizando el recibimiento de las RPM requeridas y de su gráfica en tiempo real

5. Diagramas de flujo

Se presentan los diagramas de flujo con sus modificaciones, de igual forma se añadieron en el archivo zip como un archivo aparte para una visualización más adecuada.

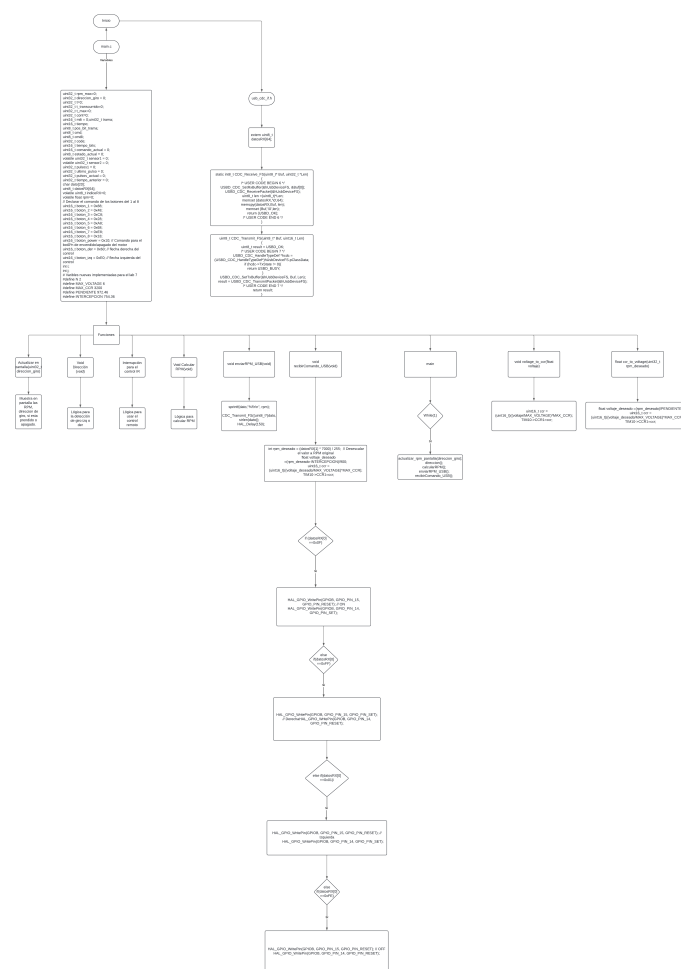


Fig. 6: Diagrama de flujo stm32cubeIDE

Se presenta el diagrama de flujo del archivo de qt para enviar y recibir los datos por medio de la app diseñada (Fig. 5).

IV. CONCLUSION

Se logró obtener la gráfica de los RPM vs Tiempo, utilizando un botón en la ventana de la aplicación de Qt el cual representa las RPM deseadas y se cambia el sentido de giro si el valor de las RPM es negativo.

Se logró determinar comportamiento de nuestro sistema, de modo que se pudo determinar una ecuación lineal que caracteriza al sistema, así se pudo determinar la relación de las RPM con el voltaje que requiere el motor para su funcionamiento.

A pesar de caracterizar el sistema, se logró determinar que nuestro sistema no es un sistema lineal esto por discrepancias

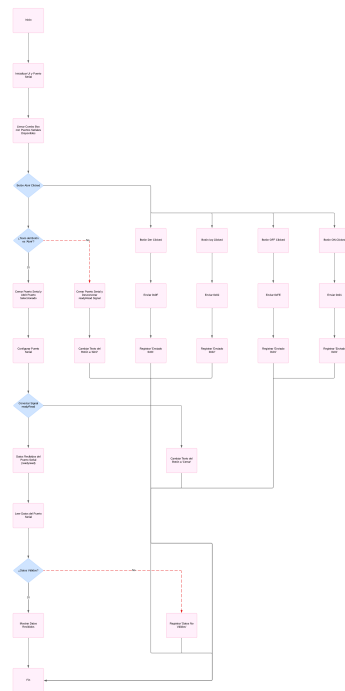


Fig. 7: Diagrama de flujo qt

en la medición de los pulsos, esto puede ser debido a malos cálculos o irregularidades en los sensores de herradura

A pesar de que se le aplicó un filtro FIR para mayor suavidad en la toma de datos de las RPM, este presenta cierta dispersión en la toma de datos que afecta en el funcionamiento del sistema, esto debido por las razones ya mencionadas.