Laboratorio 7 - Sistemas Embebidos

Wendy Chaparro, Cristopher Ramírez
Escuela de Ciencias Exactas e Ingeniería
Universidad Sergio Arboleda - Bogotá, Colombia
wendy.chaparro01@usa.edu.co, cristopher.ramirez01@usa.edu.co

1. INTRODUCCIÓN

En este laboratorio se diseñó e implementó un protocolo de transmisión con el objetivo de enviar los datos de las RPM (revoluciones por minuto) para ser visualizadas desde una interfaz gráfica de QT Creator, donde se ingresan las RPM requeridas y así el microcontrolador, mediante el uso del PWM, pueda indicar el valor de voltaje para mover el motor y lograr llegar al valor de rmp ingresadas, a esto se le conoce como control feedforward.

Para esta tarea, se utilizó la placa de desarrollo STM32F411 y el entorno de desarrollo STM32CubeIDE, herramientas que permitieron la configuración, envío de los datos de RPM (a través de un protocolo personalizado) y tiempo máximo y recepción de la indicación de apagado y RPM deseadas.

La placa STM32F411 se configuró para recibir los datos de RPM deseadas a través del puerto serial. En la aplicación QT Creator, se desarrolló un programa en C++ que permitió la visualización, el envío de las RPM deseadas y el control de los datos recibidos para el monitoreo en tiempo real de las RPM.

2. MATERIALES

- Sensor infrarrojo de herradura KLH512.
- Modulo puente H TB6612FNG.
- Placa de desarrollo STM32F411.
- Motor DC.
- Disco con ranuras.
- STM32CubeIDE - QT Creator - Hercules.

3. ANÁLISIS Y DESARROLLO

inicialmente se configuró la tarjeta de desarrollo STM32F411 para que recibiera los datos como puerto virtual, en ella se hicieron los ajustes necesarios para hacer la medición de las RPM, como lo son las RPM máximas y el tiempo que se demora en llegar a dichas RPM. Por otro lado se hizo la configuración de un PWM para el manejo de la energía entregada al motor DC (configuración del CP). Con estas configuraciones se recibieron los datos del voltaje deseado para el motor mediante el software Hercules, a fin de hacer las respectivas pruebas de funcionamiento, donde se obtuvieron los datos de las RPM máximas con cada uno de los valores de voltaje (ajustados entre 0 voltios y 7 voltios).

Con estos datos de RPM, se obtuvo una gráfica en la que se observa el comportamiento de las RPM hasta llegar a su valor máximo en un tiempo determinado (3). En esté gráfico se observa que los datos de RPM tienen un ruido los cuales muestran la variación que tienen estos datos, por lo cual se aplica un filtro FIR (Finite Impulse Response) el cual permite dar un valor más exacto de las RPM. En la gráfica también se observan los datos de RPM con el Filtro, donde se evidencia una mayor exactitud y poca variación de las RPM.

Posteriormente, se hizo una relación de los voltios con los valores máximos de RPM para obtener la ecuación de la recta (2) y así identificar su ecuación inversa para dar paso al controlador feedforward el cual controla el voltaje de PWM necesario para llegar a las RPM deseadas por el usuario Y sentido de giro que se necesite.

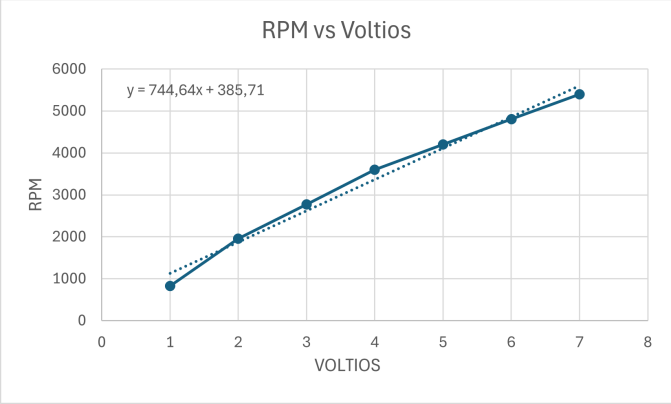


Figura 1: Gráfica de RPM máxima en función del voltaje.

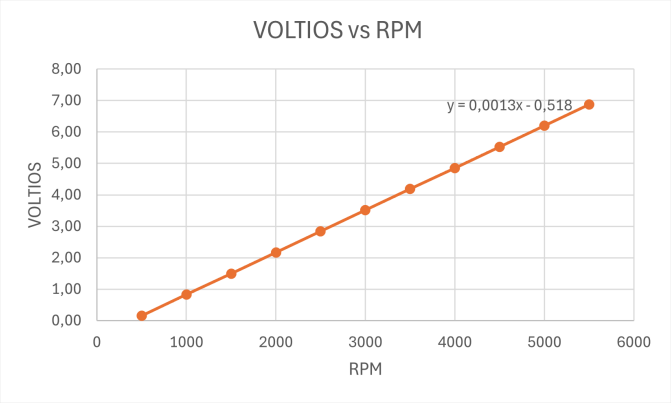


Figura 2: Gráfica de voltaje en función de las RPM deseadas.

- RPM en función del voltaje:

$$\text{RPM} = 744{,}64 \times \text{volts} + 385{,}71 \tag{1}$$

- Voltaje en función de las RPM:

$$\text{volts} = (\text{RPM} - 385{,}71) \div 744{,}64 \tag{2}$$

Lo siguiente que se hizo para esta practica fue enviar las RPM deseadas a través de QT y recibir los datos en tiempo real de las RPM y el tiempo transcurrido.

- **problemas en la ejecución:**
  - **Recibir las RPM en qt:** al ser un dato de tipo flotante no se sabia como implementar el algoritmo de decodificacion por lo que se decidió por hacer un cast de variable a tipo entero para facilitar el manejo y también por convención.
  - **enviar las rpm deseadas:** llegaban todos los datos en una misma posición del arreglo donde se hacia la recepción en la placa de desarrollo lo cual generaba que los datos enviados tomaran un valor diferente cuando eran negativos. La solución que se hallo fue en el desplazamiento de bits ya que por medio de este proceso se logro partir el dato en 2 para el caso de un numero negativo y hacer la respectiva decodificación en la stm.

4. RESULTADOS

En el laboratorio, se lograron varios resultados importantes relacionados con la medición y control de las RPM de un motor DC utilizando la STM32F411 y un protocolo personalizado para la transmisión de datos a una interfaz gráfica en QT Creator.

Se pudo medir el comportamiento de las RPM hasta llegar a su valor máximo en función del voltaje, obteniéndose una gráfica que mostraba variaciones de los datos debido al ruido. Posteriormente, se aplicó un filtro FIR para reducir ese ruido y obtener datos más precisos. Se estableció una relación entre el voltaje aplicado al motor y las RPM alcanzadas, obteniendo las funciones que permitieron la configuración de la STM32F411 para que se obtuviera el voltaje necesario para hacer girar el motor a las RPM deseadas.

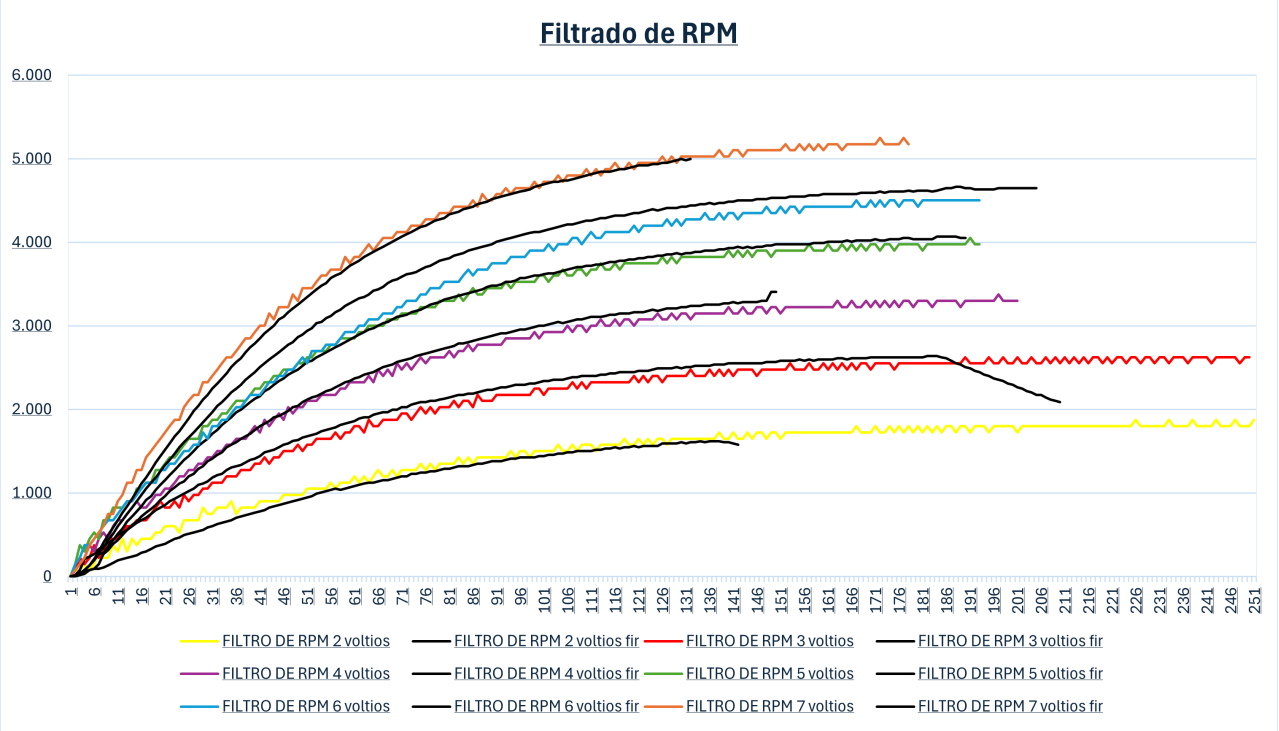
Pese a las dificultades en QT Creator con la recepción de los datos, se logró crear la interfaz (4), que permite ver de manera gráfica el comportamiento de las RPM a medida que avanza el tiempo, el tiempo máximo que se demora el motor en llegar a las RPM máximas, el valor de las RPM maximas y tener la opción de encendido y apagado del motor.

5. CONCLUSIONES

- La relación lineal entre el voltaje y las RPM facilita predecir el comportamiento del motor ante diferentes entradas de voltaje, lo que es útil para futuras implementaciones en sistemas de control más complejos
- La implementación del filtro FIR fue fundamental para obtener mediciones de RPM más precisas y confiables, reduciendo el ruido presente en los datos iniciales.
- El uso de PWM junto con la relación voltaje-RPM permitió un control eficiente del motor, ajustando el voltaje para alcanzar las RPM deseadas de manera rápida y precisa.
- Los problemas con la transmisión de datos, especialmente en la representación de números flotantes y negativos, se resolvieron eficazmente mediante la manipulación de bits y conversión de tipos, mejorando la robustez del sistema.
- La interfaz de QT Creator permitió visualizar las RPM del motor en tiempo real, lo que facilitó el monitoreo constante de su comportamiento y la respuesta ante diferentes valores de voltaje. Esto ayudó a ajustar rápidamente los parámetros durante el proceso.

Apéndice

Figura 3: Gráfica de filtrado de RPM



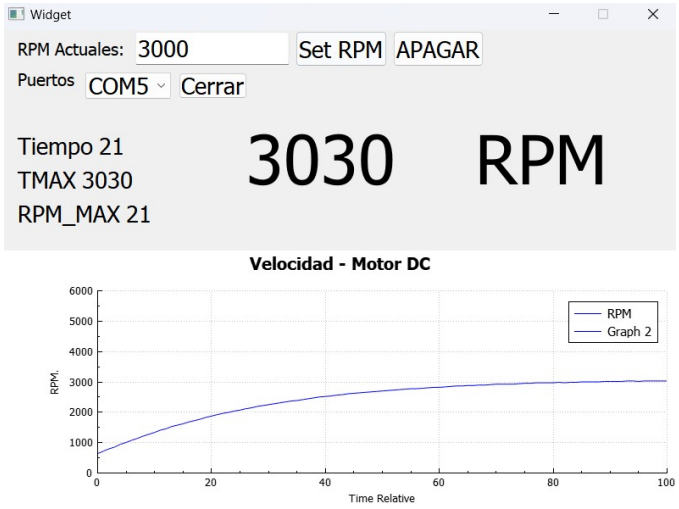


Figura 4: Interfaz gráfica en QT Creator.