



Control de Sistemas: Taller 1

“Motor de corriente continua”

Instrucciones

- Este taller es grupal.
- Completen el trabajo del “pre-taller” antes de la sesión (en forma individual). Se revisará que tengan los elementos instalados y los temas estudiados (martes 5 de abril de 2022).
- El objetivo del implementar un sistema de control para un motor de corriente continua y guardar los datos para su posterior analisis.

1. Pre-Taller

1. Instalar software Arduino IDE: <https://www.arduino.cc/en/main/software>.
2. Estudiar el comportamiento y uso de las funciones `delay()`, `millis()`, `micros()`, `attachInterrupt()` y los conceptos básicos del lenguaje de programación de Arduino.
3. Leer documentación de motores y encoders: <https://www.pololu.com/product/2823>
4. Instalar la librería de Arduino del Puente H: <https://github.com/pololu/dual-vnh5019-motor-shield>
5. Estudiar el código de demostración de la librería del puente H.
6. Traer un computador (por grupo) con todo el software instalada para todas las sesiones de laboratorio.
7. Contactar al profesor en caso de tener algún inconveniente.

2. Taller

1. Control Puente H y Escalón de 3VDC: Ejecutar el código de demostración incluido en la librería del puente H. Luego modificar el código (nuevo archivo) que permita hacer un cambio en escalón cada 10 s de 3VDC en un motor. El voltaje debe estar en 3VDC por 5 s antes de volver a 0VDC por otros 5 s. Hint: `delay()`.
2. Imprimir voltaje aplicado: Modificar el programa anterior, imprimiendo un mensaje en la compuerta serial cada 100 ms indicando el voltaje aplicado al motor (no es necesario medirlo solo indicarlo). No esta permitido utilizar la función `delay()`. Hint: `millis()` o `micros()`.
3. Lectura tren de Pulsos del Encoder: Hacer un nuevo programa que cuente el número de pulsos (subida y bajada) de uno de los pines del encoder de uno de los motores (ver Tabla 1 con los pines). Reporte en el terminal serial el valor del contador. Hint: `attachInterrupt()`.

4. Lectura Encoders: Hacer un nuevo programa que reporte en el terminal serial el valor del contador de pasos del encoder del motor seleccionado.
La tabla siguiente muestra los pines del Arduino Mega 2560 utilizados para conectar los encoders de cada motor.

| | A | B |
|-----------|----|----|
| Encoder 0 | 19 | 18 |
| Encoder 1 | 20 | 21 |

Hint: `attachInterrupt()`.

5. Calcular la velocidad de las ruedas en revoluciones por minuto: Calcular la relación entre el contador de pasos del encoder del motor y la velocidad de las ruedas en revoluciones por minuto y reportarla cada 100 ms.
6. Integrar el código del puente H con los encoders:
Hacer un programa que permita hacer una entrada en escalón de 3VDC cada 10 s y al mismo tiempo imprimir la velocidad de la ruedas en revoluciones por minuto (RPM) (para ambos motores).
Reportar cada 10 ms los siguientes valores separados por coma:
 - a) Tiempo actual: en micros.
 - b) Pos0: Contador de la posición del encoder 0.
 - c) Pos1: Contador de la posición del encoder 1.
 - d) RefVel: Valor de la referencia (util cuando cerremos el lazo).
 - e) Vel0: Velocidad de la rueda 0 en RPM.
 - f) Vel1: Velocidad de la rueda 1 en RPM.
 - g) MotorOut0: Valor en PWM enviado a al motor 0.
 - h) MotorOut1: Valor en PWM enviado a al motor 1.

*****NOTA***:** Guardar un archivo con los datos para su posterior análisis en la tarea.

3. Resultados

1. ¿Cómo funciona y para qué se utiliza un encoder? Identificar el factor que relaciona las mediciones en encoder y velocidad angular de la rueda en RPM.
2. Hacer un diagrama que muestre la conexión del sistema (indicando entradas y salidas con sus unidades).
¿Corresponde a un lazo abierto o un lazo cerrado?

Enviar sus respuestas en un archivo comprimido con su número de grupo y su nombre (**Taller0_G#NUMGRUPO_NOMBRE.zip**) al cuestionario habilitado en Canvas, conteniendo un archivo PDF con la resolución de las preguntas y todos los códigos desarrollados con su grupo en el transcurso de la sesión, debidamente comentados.

Plazo de entrega: Viernes 8 de abril - 15:30 hrs.