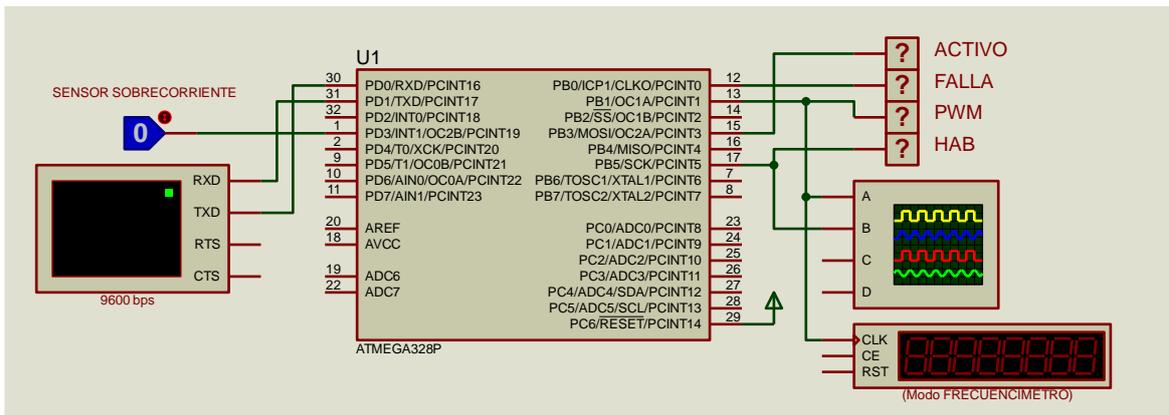


MICROCONTROLADORES Y ELECTRÓNICA DE POTENCIA

Examen global Integrador – Parte 2: Programación de un control de eje

IMPORTANTE: Forma de entrega:

- 1- Nombrar al **proyecto** en Microchip Studio y la **solución** con APELLIDO_NroLegajo, por ejemplo: **PEREZ_12345**
- 2- (optativo). Una vez compilado, crear un circuito similar al de la figura y guardar en la carpeta Debug del proyecto.
- 3- Una vez ensayado el programa, comprimir toda la carpeta proyecto/solución en un archivo GLOBAL_APELLIDO_NroLegajo, por ejemplo: **GLOBAL_PEREZ_12345.rar (rar o zip)**
- 4- Enviar el archivo comprimido a micro.uncu@gmail.com, con el **asunto** Global Apellido Legajo, por ejemplo: **Global Perez 12345. En caso de falla de WiFi se instrumentará la entrega en pendrive.**
- 5- **NO SE SIMULARÁ EL MOTOR, SOLAMENTE SE VERIFICARÁ EL FUNCIONAMIENTO CON LOS INSTRUMENTOS (osciloscopio, frecuencímetro, leds)**



Consigna:

Realice con un ATmega328P a 16MHz el programa de una interfaz de driver de motor DC por PWM, que responda a comandos por puerto serie (UART), a **9600 bps**.

El microcontrolador debe comandar una habilitación **HAB** (PB5) y una salida PWM (OC1A) a la frecuencia **solicitada** (4000, 5000, 6400) hacia un driver tipo L298, y tendrá una entrada de detección de sobrecorriente en **INT1**. Adicionalmente tendrá dos salidas a leds, **ACTIVO** (PB3), y **FALLA** (PB0).

Inicialmente el controlador estará en estado **E_inactivo**, con salida **HAB** = 0 duty cycle = 0 en la salida PWM y salida **ACTIVO**=0. Cuando pase a modo **E_activo** mediante un comando **:A1** (ver tabla abajo) la salida **ACTIVO** pasará a oscilar a 1Hz.

Las consignas y las respuestas que debe dar se resumen en la tabla:

Si hay sobrecorriente un comparador externo conectado en **INT1** pasará a '1' (simulado con el inyector lógico LogicToggle). Esto debe provocar el paso al estado **E_falla**, en el cual hará el led **HAB**=0, pondrá el duty cycle en 0, pondrá led **FALLA**=1 y transmitirá el mensaje **“:FALLA\r”**. Se requerirá un **hard reset** del microcontrolador para salir del estado **E_falla**

El mensaje inicial en la UART debe ser su apellido y Nro de Legajo, por ejemplo “PEREZ 12345\r\n”

	Comando	Descripción	Observación
1	:A1\r	Activación	Responde con eco. Pasa al estado E_activo . En este estado HAB =1 y parpadea la salida ACTIVO (frecuencia de 1Hz aproximadamente)
2	:A0\r	Desactivación	Responde con eco. Pasa al estado E_inactivo . En este estado HAB =0 y ACTIVO = 0
3	:Dnnn\r	Consigna de duty cycle, de 0 a 999 (corresponde a 0.0 a 99.9 % en incrementos de 0.1)	Actualiza el valor de duty cycle y responde con eco (:Dnnn\r). Si el comando no tiene argumentos responde con el valor actual de duty cycle. Si la consigna excede los límites no actualiza el valor y responde con el valor actual de duty cycle (formato 0 a 999)
4	:Tnnnn\r (opcional)	Período de PWM en microsegundos	Permite configurar el período, mínimo 100us (10kHz), máximo 1000us (1kHz). Responde con eco. Deberá escalarse el duty cycle.

Planteo de estructura general hardware/configuración/inicialización/uso de periféricos/interrupciones: hasta 50 pts.

Interpretación y ejecución de comandos, verificación de rangos etc: hasta 30 pts.

Implementación de estados: hasta 20 pts.

Opcional: 10 pts extra si funciona correctamente