

Problemas B: Assembly II

Importante: No olvide realizar los diagramas de flujo correspondientes para cada ejercicio y consultar el Set de Instrucciones (<http://goo.gl/nuZC20>)

Ejercicio B.1:

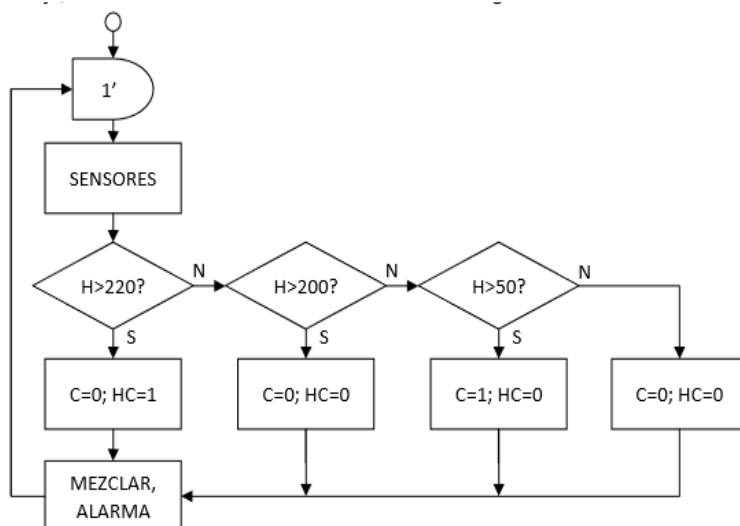
- Un sistema embebido basado en la familia AVR opera con un cristal de 1 MHz. ¿Cómo implementaría una función de retardo de 0,05 ms? ¿Qué instrucciones condicionales podría utilizar? ¿Cuáles son las entradas y salidas (y su función)?
- ¿Y un retardo de 10 ms?
- ¿Y de 1 segundo?

Ejercicio B.2:

- Un tanque mezclador acumula la descarga de dos procesos químicos para elaborar un preparado. La capacidad del tanque es 250 m³ pero se estableció operar como máximo a 200 m³ y establecer 220 m³ como nivel de alarma. Además, como el mezclador no puede trabajar con poca carga, no conviene encenderlo con menos de 50 m³.

La especificación de la rutina SENSORES indica que toma la lectura de nivel “H”, la convierte a m³, y guarda el valor en el Registro 10. La rutina MEZCLAR maneja el motor según el estado del carry (C=0 apaga; C=1 enciende) y la rutina ALARMA maneja la sirena según el estado del half-carry (HC=0 apaga; HC=1 enciende).

A ustedes se les pide que desarrollen el resto del programa de manera tal que 1 vez por minuto se mida el nivel del tanque y se actúe en consecuencia. Dado el siguiente diagrama de flujo, ¿cómo lo resolverían? ¿Qué características del diagrama les llaman la atención?



- El Jefe de Mantenimiento nos avisó que una vez que el motor se enciende debe permanecer funcionando al menos 5 minutos, y cuando se apaga no se lo puede encender por los próximos 10 minutos. ¿Cómo modifican el diagrama de flujo y el programa estas restricciones?

Problemas B: Assembly II

- c. Cuando el Ingeniero de Procesos se enteró de todas las restricciones nos pidió que el sistema genere una señal READY a los procesos anteriores para avisarles cuándo pueden descargar en el mezclador y cuándo no. Para esta señal podría usarse PORTA.0. ¿Cómo cambia ahora?

Ejercicio B.3:

- a. Si quisiéramos hacer analogías entre Assembly y lenguajes de alto nivel como C, ¿cómo construiría la estructura del do-while?
DO {
 Lazo
} WHILE (condición = TRUE)
- b. ¿Y la del for?
FOR (índice, conteo) {
 Lazo
}
- c. ¿Y el if-then-else?
IF (variable = C) THEN {
 Verdadero
} ELSE {
 Falso
}