

- Nombre y CI en cada hoja
- Numere las hojas, indique el total en la primera
- Utilice solo un lado de las hojas

- Incluya un solo problema por hoja
- **Sea prolijo**
- **Aprobación:** mínimo UN problema

## PROBLEMA 1

Se deberá diseñar un sistema de control para un clasificador de productos que se mueven sobre una cinta transportadora.

### Descripción general

Por la cintas viajan productos, que al llegar a una zona de control son pesados y si el peso es menor o igual a un valor de referencia, un selector debe posicionarse para que caigan en la caja A, por el contrario si el peso es mayor, el selector debe ubicarse para que caiga en la caja B.

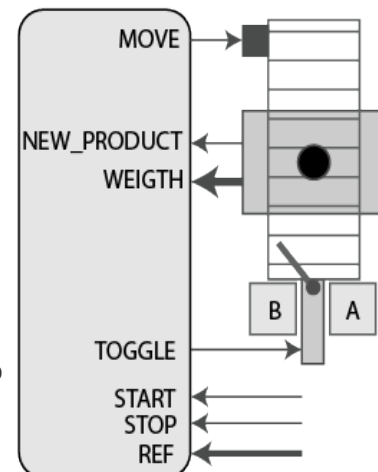
Dos botones START y STOP son utilizados para controlar el funcionamiento de la cinta.

El peso de referencia se debe leer de la entrada REF en el momento que se da la orden de START

### Detalles de la implementación

Señales de control y monitoreo:

- El motor de la cinta transportadora funciona mientras su entrada MOVE se encuentra en "1"
- Cada vez que llega un producto a la zona de pesado, se genera un pulso de corta duración en la señal NEW\_PRODUCT.
- Una balanza en la zona de control, reporta el peso del producto (en grs) en forma continua a través de su salida WEIGTH de 8 bits.
- El selector comienza direccionando los productos a la caja A. En cada flanco de subida de su entrada TOGGLE cambia de posición.
- START y STOP son flancos de subida.
- La entrada REF de 8 bits tiene un valor válido sólo en el flanco de subida de START



El monitoreo de la señal NEW\_PRODUCT deberá realizarse mediante interrupciones modo 1 y el control del selector debe realizarse en la rutina de atención a la interrupción.

El programa principal se encargará de inicializar el sistema (comenzando con el motor detenido), controlar el motor y leer el peso de referencia.

El sistema es utilizado para otras tarea, por ello las direcciones del espacio de entrada/salida 20h a 2Fh y C0h a DFh no están disponibles.

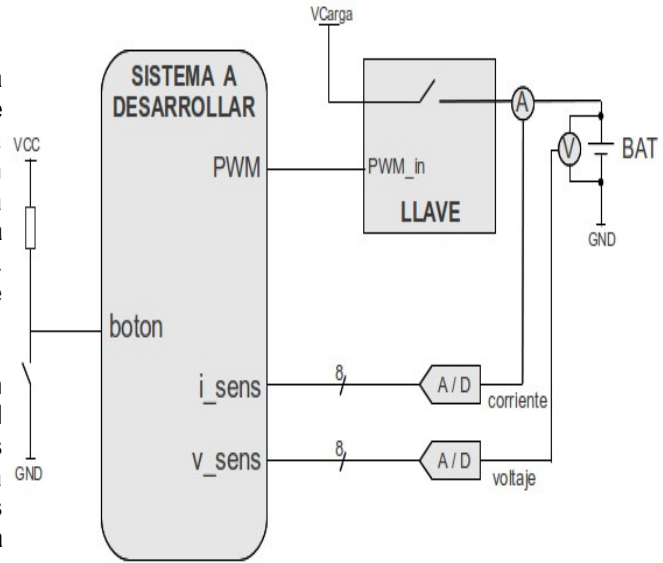
Se pide:

- Todo el hardware (memoria, periféricos, puertos) necesario para implementar el sistema
- Rutina de atención a la interrupción
- Inicialización, reserva de memoria y programa principal que maneje el arranque y la detención de la cinta, y lectura de REF. Para el manejo de las otras tareas del sistema, durante la inicialización debe invocarse la subrutina *ini\_otros* y dentro de los bucles de polling de las señales de control de la cinta debe invocarse la rutina *otras\_tareas*. Ambas subrutinas se supondrán implementadas y ya grabadas en ROM.

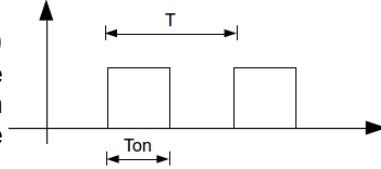
**PROBLEMA 2**

El bloque de control de un cargador de batería controla una llave que conecta una fuente de tensión  $V_{carga}$ , mayor que la tensión deseada en la batería. Para manejar la llave se utiliza modulación por ancho de pulso (PWM). La llave se acciona periódicamente a intervalos  $T$ , estando encendida durante un tiempo  $T_{on}$  y apagada durante  $T - T_{on}$ . Observar que a mayor  $T_{on}$  será mayor la corriente de carga a la batería.

Se desea agregar a un sistema con microprocesador existente la función de control del cargador de batería, Se trabajará por interrupciones para no interferir con el resto del sistema. Una interrupción periódica de período fijo  $T = 500 \mu s$  debe subir la salida **pwm**. En cada período una segunda interrupción se debe producir **Ton** después de la primera para bajar la salida **pwm**. Para generar ambas interrupciones se utilizará un CTC.



El valor de  $T_{on}$  debe ajustarse cada 10 períodos  $T$  (en una cada 10 instancias de la interrupción periódica), modificando el valor de la variable  $n_{ton}$  ( $T_{on} = n_{ton} \times 5\mu s$ ) en función de los valores de corriente y tensión leídos (entradas  $v_{sens}$  e  $i_{sens}$  de 8 bits sin signo) como se describe más adelante.



Luego de un reset el cargador debe iniciar en **modo apagado**, con la salida **pwm** en 0. El cargador pasa a **modo encendido** con un pulso a 0 de corta duración en la entrada **boton**. En este modo  $T_{on}$  debe ajustarse procurando mantener la corriente de carga igual a **IREF** (constante de 8 bits). Para eso se debe leer el sensor de corriente y modificar la variable  $n_{ton}$  dependiendo del valor  $i_{sens}$  leído:

- $i_{sens} > IREF \Rightarrow n_{ton}$  se decrementa en 1, hasta un mínimo de  $n_{ton\_min}$
- $i_{sens} = IREF \Rightarrow n_{ton}$  no se modifica
- $i_{sens} < IREF \Rightarrow n_{ton}$  se incrementa en 1, hasta un máximo de  $n_{ton\_max}$

El cargador se mantiene en **modo encendido** hasta que llegue un nuevo pulso en **boton** o la tensión (obtenida del sensor de tensión) supere **VFLOT** (constante de 8 bits). En cualquiera de los dos casos debe volverse al **modo apagado** y esperar por un nuevo pulso en **boton**.

En todos los casos cuando arranca el cargador debe comenzar con  $n_{ton} = n_{ton\_min}$ .

La entrada **boton** debe sensarse en la interrupción periódica.

Se pide:

- a) Conexión del CTC y puertos necesarios para la funcionalidad del cargador, incluyendo su decodificación. Se puede utilizar para ese fin el rango de direcciones 0x80-0xBF de E/S.
- b) Inicialización de sistema interrupciones del Z80 y de todo lo necesario para la función de cargador de batería.
- c) Escribir el código assembler de la interrupción periódica de acuerdo al pseudocódigo dado en el recuadro. La subrutina **new\_n\_ton()**, que se supone implementada, recibe como parámetro el valor actual de  $n_{ton}$ , lee el sensor de corriente, lo compara con **IREF** y devuelve como resultado el valor actualizado de  $n_{ton}$ . La subrutina utiliza el acumulador para recibir el parámetro y devolver el resultado.
- d) Subrutina de servicio de interrupción al final de  $T_{on}$ .

**rutint\_periodica**

```

Si llegó pulso botón{
  complemento modo
  Si ( modo encendido) {
    inicializo cont10veces
    inicializo var n_ton
  }
}
Si (modo encendido) {
  pwm = 1
  arranco temporizador ton
  modifiko cont10veces
  Si es la décima interrupción {
    n_ton = new_n_ton(n_ton)
    inicializo cont10veces
  }
  Si tensión > VFLOT {
    modo = apagado
  }
}

```

**Datos:**

$VFLOT = 120$      $IREF = 85$   
 $fclk = 51.2MHz$      $5\mu s = (1 / fclk) \times 256$      $T = 500 \mu s = (1 / fclk) \times 256 \times 100$