IMPORTANTE : Hay un error en el diagrama de flujo que luego se traslada al código

Resaltado los errores de menor importancia

NOTA: La interrupción no debe incluirse en el diagrama de flujo del programa main

DSteiner

;-------------------------------------------------------------------------------

; Fecha: 08/06/2021

;

; Programador: Marina Donis

;

; Descripcion: PARCIAL 2021

/\*

Utilizando Timer 2 en un 8052,

configurarlo para producir interrupciones cada centesima de segundo (100 Hz)

y generar mediante variables los valores de: centesima, segundo, minuto y hora.

Al dar RESET, dicho reloj se carga con h,m,s,c 00:00:00:00

Se pide encender una salida en P2.0 cuando transcurran:

2h, 1m, 10s, 90 centesimas .

Nota: el proceso SOLAMENTE se repite si se da nuevamente RESET

\*/

;-------------------------------------------------------------------------------

$bitdef.h ; manejo de a bits

$ioAT89C52.h ; incluye los registros del microcontrolador

NAME main

PUBLIC main

;--ISRs

PUBLIC Isr\_Tmr2

;--Declaro publicas las subrutinas

PUBLIC InitPerif\_Interrupciones

;---CONSTANTES------------------------------------------------------------------

;Registros generales

CTE\_SUMA EQU 0x3E ; Numero cte para operar

LONG\_BUFFER EQU 10 ; Longitud del buffer

ADDR\_INICIAL EQU 0x40 ; Direccion de memoria inicial

;Constantes para los segmentos

SEG\_CODIGO EQU 0030h ; inicio seg codigo

SEG\_DATOS1 EQU 0020h ; seg datos

SEG\_CONST EQU 0090h ; Constantes en rom

SEG\_STACK EQU 00C0h ; Segmento de stack

LONG\_STACK EQU 10 ; Longitud en bytes del stack

;--------------REGISTROS DE LOS TIMERS------------------------------------------

;Para interrupciones

VALOR\_IP EQU 0x00 ; prioridad baja para todas

; IE 8052: EA X ET2 ES ET1 EX1 ET0 EX0 / VER PAGINA 10 DEL MANUAL

VALOR\_IE EQU b'10100000 ; EA=1, ET2=1, TIMER 2 INT

; habilita interrupciones e interrupcion del timer 2

; 8052 T2MOD: X X X X X X T2OE DCEN / VER PAGINA 8 DEL MANUAL

VALOR\_T2MOD EQU b'00000000 ; TIMER 2 MODO AUTORECARGA DCEN=0

; Pone al timer 2 en modo autorecarga

; 8052 T2CON: TF2 EXF2 RCLK TCLK EXEN2 TR2 CT2 CPRL2 / VER PAGINA 5 DEL MANUAL

VALOR\_T2CON EQU b'00000100 ; TR2=1 TIMER 2 ON, EXEN2=0, CT2=0

; Prende el timer 2 para que empiece a contar

;----------ACA CALCULO LA CUENTA DEL TIMER Y EL VALOR DE RECARGA----------------

; Tengo de frecuencia de entrada al timer 12Mhz/12 -> 1MHz (cristal de 12MHz)

; Frecuencia de salida del timer de 100 Hz ya que lo pide el enunciado (1 centesima de segundo)

; Cuenta: 1Mhz / 100 Hz = 10000d = 2710H, estos son los ciclos que debe contar el timer.

; Como es timer ascendente, calculo valor complementario de lo que quiero contar

; Como la interrupcion se realiza al desbordar de FFFF(timer 2 16 bits)

; Debo recargar el timer con FFFFH + 1 - 2170H = D8F0H

VALOR\_TIMER EQU 2170H

RECARGA\_TMR EQU (0xFFFF+1-VALOR\_TIMER) ; Valor de recarga del timer = D8F0H

;----------ACA DEFINO MIS VARIABLES Y CONSTANTES--------------------------------

; Para el reloj

HORA EQU 030H

MINUTO EQU 031H

SEGUNDO EQU 032H

CENTESIMA EQU 033H

; Puerto de salida por el que aviso

PUERTO\_SALIDA EQU P2.0

;-------------------------------------------------------------------------------

;---SEGMENTO DE CODIGO----------------------------------------------------------

; Direct the reset vector to the program entry label.

;asegn INTVEC:CODE, 0x0000

ASEG

ORG 0000h ; RESET

LJMP main

;--------VECTORES DE INTERRUPCION-----------------------------------------------

; INT TIMER2 vector

ASEG

ORG 002BH ; Vector de interrupcion del timer 2 (8052)

LJMP Isr\_Tmr2 ; Definir el el salto a la ISR

;---SEGMENTO DE STACK-----------------------------------------------------------

;RSEG ISTACK:IDATA:NOROOT(0)

ASEG

ORG 00b8h

stack\_init DS LONG\_STACK ;Define el segmento de stack

;---SEGMENTO DE DATOS-----------------------------------------------------------

;:::Segmento relocalizable::::::::::::::

; RSEG IDATA\_Z:DATA:NOROOT

;::::::::::::::::::::::::::::::::::::::

;::Segmento absoluto:::::::::::::::::::::

ASEG

ORG SEG\_DATOS1

;::::::::::::::::::::::::::::::::::::::

var\_bit DS 1

;ESPACIO de BUFFER

buffer DS LONG\_BUFFER

;TABLA RAM

RSEG IDATA\_Z:DATA:NOROOT

;-------------------------------------------------------------------------------

;Inicia el c?digo

;RSEG NEAR\_CODE:CODE:ROOT ;defino un segmento de c?digo

RSEG RCODE

ORG SEG\_CODIGO ;define el punto de inicio del Programa (code)

main

;\*\*Inicializaciones

;Inicializo el STACK

MOV SP,#stack\_init-1 ;inicializo el stack

;---------------ACA INICIALIZO LOS VALORES DE MIS VARIABLES---------------------

; Inicializo el reloj

MOV HORA, #0

MOV MINUTO, #0

MOV SEGUNDO, #0

MOV CENTESIMA, #0

; Por ultimo antes de entrar al loop principal inicializo el timer

;Inicializo las Interrupciones

CALL InitPerif\_Interrupciones

CLR PUERTO\_SALIDA falta

; a partir de ahora el timer esta contando y se producen las interrupciones

;\*\*Inicio loop principal

Loop\_ppal:

SALIR: JMP SALIR ; loop infinito

;-------Subrutinas--------------------------------------------------------------

RSEG NEAR\_CODE:CODE:NOROOT(0)

;===========SUBRUTINAS==========================================================

;-------------------------------------------------------------------------------

; Nombre: InitPerif\_Interrupciones

; Parametros:ninguno

; Devuelve:nada

; Modifica:

; IP, IE, TCON, TMOD,TH0,TL0

; Nivel de Stack: 2

;

; Descripci?n:

; Inicializa las interrupciones y algunos perif?ricos como el timer

;-------------------------------------------------------------------------------

InitPerif\_Interrupciones:

;---------ACA INICIALIZO LOS TIMERS, TMOD, TCON, TL, TH, IP, IE-----------------

;Para el timer 2

; Cargo los registros, valor de recarga D8F0H

MOV TL2, #LOW(RECARGA\_TMR) ; Recarga parte baja del timer (0F0H)

MOV TH2, #HIGH(RECARGA\_TMR) ; Recargo parte alta del Timer 2 (0D8H)

; Cargo los registros de recarga

MOV RCAP2H, #LOW(RECARGA\_TMR) ; Valor de recarga alto (0F0H)

MOV RCAP2L, #HIGH(RECARGA\_TMR) ; Valor de recarga bajo (0D8H)

MOV T2MOD, #VALOR\_T2MOD ; Inicializo Timer/Counter Mode Register segun la constante

MOV T2CON, #VALOR\_T2CON ; Inicializo Timer/Counter Control Register segun la constante

; Seleccionar la prioridad de las interrupciones

MOV IP,#VALOR\_IP ; Inicializo prioridades segun la constante

; Habilitar interrupci?n, es lo ?ltimo a hacer antes de entrar al lazo ppal.

MOV IE,#VALOR\_IE ;Habilito las interrupciones seg?n la cte

; las interrupciones las habilito ULTIMAS para que empiece a contar desde ahora,

; sino no esta cargado el timer pero puede interrumpir

; tiene que estar TODO listo ANTES de inicializar las interrupciones

ret

;-------------------------------------------------------------------------------

;===ISRs========================================================================

;-------------------------------------------------------------------------------

; Nombre: Isr\_Tmr2

; Parametros:ninguno

; Devuelve:nada

; Descripcion:

; Rutina de servicio de interrupcion del timer 2

; Avisa cada 2h, 1m, 10s, 90 centesimas por el puerto 2

;-------------------------------------------------------------------------------

Isr\_Tmr2:

;-------------------------------------------------------------------------

; Preparo todo para atender la interrupción (Salvo el contexto)

;-------------------------------------------------------------------------

PUSH PSW ; Resguardo registros modificados por esta subrutina

; como en la interrupcion esta voy a sumar y etc, guardo el contexto para poder usar estos registros

; "salvo" el PSW y el A porque lo use en la rutina, se guarda en el stack

PUSH A ; ahora el stack tiene dir retorno + PSW + A pq los use aca adentro y los pushee

;-------------------------------------------------------------------------

; Aca se realiza lo pertinente a la rutina de Interrupcion

;-------------------------------------------------------------------------

; Primero hago que cuente el tiempo

; 1 segundo = 100 centesimas, 1 minuto = 60 segundos, 1 hora = 60 minutos

; Como contamos por centesimas (100Hz), cada vez que llegue a 100 centesimas

; debo volverlas a 0 y aumentar 1 minuto.

; Luego si tengo 60 minutos debo volverlos a 1 y aumentar 1 hora, y asi sucesivamente

;------------------VERIFICO CENTESIMA-------------------------------------

MOV A, CENTESIMA ; Verifico el valor en CENTESIMA

CJNE A, #99, INCREMENTO\_CENT ; Si CENTESIMA es 99, debo incrementar un SEGUNDO

INC SEGUNDO ; Incremento un SEGUNDO (100 centesimas)

MOV CENTESIMA, #0 ; Vuelvo las CENTESIMAs a 0 para que cuente 1 segundo de vuelta

JMP VERIFICAR\_SEGUNDOS ; Verifica SEGUNDOs

; Aca viene si la CENTESIMA es < 99

INCREMENTO\_CENT:

INC CENTESIMA ; Incremento CONTESIMA

JMP VERIFICAR\_TIEMPO falta

VERIFICAR\_SEGUNDOS:

;------------------VERIFICO SEGUNDOS--------------------------------------

MOV A, SEGUNDO ; Verifico valor en SEGUNDO

CJNE A, #59, INCREMENTO\_SEG ; Si SEGUNDO es 59, debo incrementar un MINUTO

INC MINUTO ; Incremento un MINUTO (60 segundos)

MOV SEGUNDO, #0 ; Vuelvo los SEGUNDOs a 0 para que cuente 1 minuto de vuelta

JMP VERIFICAR\_MINUTOS ; Verifica MINUTOs

; Aca viene si el SEGUNDO es < 59

INCREMENTO\_SEG:

INC SEGUNDO ; Incremento SEGUNDO

JMP VERIFICAR\_TIEMPO falta

VERIFICAR\_MINUTOS:

;------------------VERIFICO MINUTOS---------------------------------------

MOV A, MINUTO ; Verifico valor en MINUTO

CJNE A, #59, INCREMENTO\_MIN ; Si MINUTO es 59, debo incrementar una HORA

INC HORA ; Incremento un HORA (60 minutos)

MOV MINUTO, #0 ; Vuelvo los MINUTOSs a 0 para que cuente 1 hora de vuelta

JMP VERIFICAR\_HORAS ; Verifica HORAs

; Aca viene si el MINUTO es < 59

INCREMENTO\_MIN:

INC MINUTO ; Incremento MINUTO

JMP VERIFICAR\_TIEMPO falta

VERIFICAR\_HORAS:

;------------------VERIFICO HORAS-----------------------------------------

MOV A, HORA ; Verifico valor en HORA

CJNE A, #2, INCREMENTO\_HR ; Si HORA es 2, debo verificar si es momento de avisar por P2

JMP VERIFICAR\_TIEMPO ; Va a verificar si es el momento de avisar por P2

; Aca viene si HORA es menor a 2

INCREMENTO\_HR:

INC HORA ; Incremento HORA

JMP SALIR\_IRQ ; Salgo de la interrupcion

;-------------------VERIFICAR EL HORARIO----------------------------------

; Ahora que ya funciona como reloj (no puede contar hasta 3 horas)

; Voy a verificar si es el tiempo de avisar por el puerto.

; Hasta este punto ya verificamos todos los valores de CENTESIMA, SEGUNDO, MINUTO, HORA

; Y solo llegara a esta parte si pasaron al menos 2 horas.

VERIFICAR\_TIEMPO:

; En este punto ya me asegure de que HORA = 2 asi que no la verifico

MOV A, MINUTO ; Verifico MINUTO

CJNE A, #1, SALIR\_IRQ ; Si no es 1 (valor pedido) sale de la interrupcion y sigue contando

MOV A, SEGUNDO ; Verifico SEGUNDO

CJNE A, #10, SALIR\_IRQ ; Si no es 10 (valor pedido) sale de la interrupcion y sigue contando

MOV A, CENTESIMA ; Verifico CENTESIMA

CJNE A, #90, SALIR\_IRQ ; Si no es 90 (valor pedido) sale de la interrupcion y sigue contando

; Aca llega solamente si es la hora pedida (02:01:10:90)

SETB PUERTO\_SALIDA ; Enciende la salida por puerto 2 (aviso)

;----------------------RECUPERAR EL CONTEXTO Y SALIR----------------------------

SALIR\_IRQ:

POP A ; Recupero registros previamente resguardados

POP PSW

RETI ; Sale de la interrupcion

; RECORDAR QUE ES UNA PILA Y SALE PRIMERO LO ULTIMO QUE GUARDE

;===============================================================================

END main