

PROYECTO para el c 8051: Peluco

OBJETIVO INICIAL DEL PROYECTO:

Se realizara un reloj – despertador con visualización de horas y minutos por medio de cuatro displays de siete segmentos. Dicho reloj funcionara en modo 24 horas (las horas avanzarán desde 00 hasta 23). Tanto la hora como la alarma dispondrán de puesta en hora, operación que se realizara mediante pulsadores. La alarma se activara cuando la hora en curso sea igual a la hora programada en la alarma.

El programa deberá ser multiplataforma, es decir deberá ser posible mediante sencillos cambios en las rutinas de salida y recogida de datos por los puertos que el programa funcione en distintos entornos (placa disen, simulador u otros).

Este programa deberá funcionar correctamente en la placa disen, por lo cual tendrá varias restricciones impuestas por este entorno, como la imposibilidad de usar el puerto 0 o el temporizador 1.

FUNCIONAMIENTO DEL RELOJ:

El funcionamiento de este reloj es idéntico al de cualquier reloj despertador de sobremesa. El enchufarlo por primera vez tanto el reloj como la alarma estará en 00:00. Para poner en hora el reloj se deberá mantener pulsado un pulsador (P3.5) mientras se pulsa repetidamente otro (P3.4) para que vayan avanzando los minutos hasta la hora deseada. Para poner en hora la alarma al proceso es similar, se mantendrá pulsado otro pulsador (P3.6) visualizándose en el reloj la hora de la alarma, luego se pulsa repetidamente el pulsador de puesta en hora (P3.4) para que vaya avanzando la hora de alarma hasta la hora deseada. Al dejar de pulsar el pulsador de cambio de alarma volverá a visualizarse la hora actual.

LA PROGRAMACION:

Este programa dispone básicamente de tres partes diferenciadas.

Primeramente se realizo la parte de temporización, para ello se utilizo el temporizador interno 0 de microcontrolador programado en modo 1 es decir, en modo 16 bits. A este temporizador se le introduce le valor 5000 para que descuenta hasta 0. Cuando el temporizador termina de contar el programa principal es llamado por la interrupción (previamente habilitada), entonces se incrementa una variable y se vuelve a poner en marcha el temporizador con los mismos valores iniciales. Así cuando esta variable llega a 200 sabemos que ha pasado un segundo. Disponemos de tres variable que almacenan los segundos, minutos y horas, así cuando pasa un segundo estas variables se incrementan convenientemente.

Una vez que la temporización funcionó correctamente realizamos la parte de visualización, esta consiste en un bucle infinito, en cada bucle se selecciona un display mediante los cuatro primeros bit del puerto3 y se le manda por el puerto 2 los bits correspondientes para que visualice el numero almacenado en la variable correspondiente a dicho display. En el siguiente bucle se selecciona otro display y así sucesivamente. Este bucle es infinito y el programa solo se saldrá de el para atender la interrupción del temporizador para luego volver.

La ultima parte que realizamos fue la de puesta en hora del reloj y de la alarma. Cuando el pin de activación de puesta en hora este a 1 y cada vez que el pin correspondiente al pulsador de puesta en hora del reloj pasa de 0 a 1 y luego de nuevo a 1 se incrementa en un minuto las variable de la hora del reloj. Para la alarma se crearon 2 variables que almacenaran la hora de la alarma, así cuando se pulsa el pin de activación de puesta en hora de la alarma se salva el valor de las variables que almacenan la hora actual y se copia en las mismas el

valor de las variables que almacenan la alarma. Luego, como el la puesta en hora del reloj cada pulsación del pin de puesta en hora se aumenta en un minuto la hora de alarma. Cuando se deja de pulsar el pin de activación de puesta en hora de la alarma se guarda la hora de la alarma en sus variables correspondiente y se recupera la hora actual. Cuando la hora actual sea igual a la hora de la alarma se activa el pin de alarma durante un minuto.

DIAGRAMA DE FLUJO:

m

EL PROGRAMA: (codigo fuente completo)

```
;-----  
;PROGRAMA: PELUCON  
;-----  
;DESCRIPCION: UN RELOJ CON VISUALIZACION DE HORAS Y MINUTOS CON ALARMA  
; Y PUESTA EN HORA.  
;-----  
;-----  
;DEFINE EL ORIGEN:  
  
INCLUDE MACROS.INC ;INCLUYE LAS MACROS  
  
ORG 00H  
  
JMP 33H  
  
ORG 000BH  
  
CLR TR0  
  
JMP FINTEMP0  
  
ORG 33H  
  
INICIA_IIC ;INICIALIZA EL BUS I2C  
  
;DEFINICION DE VARIABLES  
  
TEMP0H EQU 5AH ;TIEMPO DE TEMPORIZACION (16 BITS).  
  
TEMP0L EQU 5BH  
  
SEGUNDOS EQU 5CH ;VARIABLES DE TIEMPO, SE VAN DECREMENTANDO DESDE SU
```

MINUTOS EQU 5DH ;VALOR MAXIMO.

HORAS EQU 5EH

CONTEMP EQU 5FH ;N° DE CONTADAS.

UNI_MIN EQU 6AH

DEC_MIN EQU 6BH

UNI_HOR EQU 6CH

DEC_HOR EQU 6DH

DISPLAY EQU 6EH

NBUCLE EQU 6FH

SEG EQU 70H

MIN EQU 71H

HOR EQU 72H

SEL_DISP EQU 73H

PULSADO EQU 74H

PULSADO2 EQU 75H

HORAS_AL EQU 76H

MINUTOS_AL EQU 77H

SALVA_HORAS EQU 78H

SALVA_MINUTOS EQU 79H

;INICIO PROGRAMA

MOV PULSADO,#00D

MOV PULSADO2,#00D

MOV TEMP0H,#ECH ;LLENO LAS VARIABLES DEL TEMPORIZADOR CON 5000D

MOV TEMP0L,#77H ;COMPLEMENTADO (5000D = 1388H ; FFFF-1388 = EC77H).

MOV SEGUNDOS,#59D ;INIZIALIZO LAS VARIABLES DE TIEMPO A 0.

MOV MINUTOS,#59D

```

MOV HORAS,#23D

MOV CONTEMP,#10d ;#200D ;LE DIGO EL N° DE CONTADAS QUE TIENE QUE HACER.

MOV MINUTOS_AL,#59D ;PONGO LAS VARIABLES DE LA ALARMA A 0.

MOV HORAS_AL,#23D

MOV NBUCLE,#0

MOV TL0,TEMP0L ;LE METO LOS VALORES QUE TIENE QUE CONTAR Y

MOV TH0,TEMP0H ;PROGRAMO EL TIMER 0 EN MODO 1 (16BITS).

MOV A,TMOD

ORL A,#01H

MOV TMOD,A

SETB ET0

SETB EA ;HABILITO LA INTERRUPCION DEL TIMER 0.

SETB TR0 ;PONGO EN MARCHA EL TEMP0.

VISUALIZA: ;AQUI COMIENZA EL BUCLE QUE VISUALIZA LA HORA.

MOV A,#59D ;INVIERTO LOS VALORES DE MINUTOS Y HORAS

CLR C ;PARA SACAR LOS VALORES REALES.

SUBB A,MINUTOS

MOV MIN,A

MOV A,#23D

CLR C

SUBB A,HORAS

MOV HOR,A

MOV A,MIN ;SEPARO LAS UNIDADES Y LAS DECIMAS DE LAS

MOV B,#10D ;HORAS Y LOS MINUTOS ALMACENÁNDOLOS EN SUS

DIV AB ;RESPECTIVAS VARIABLES.

MOV UNI_MIN,B

```

```

MOV DEC_MIN,A
MOV A,HOR
MOV B,#10D
DIV AB
MOV UNI_HOR,B
MOV DEC_HOR,A
MOV A,#11111111B ;ESCRIBO NADA EN EL DISPLAY.
ACALL SACAPOR_2
MOV A,NBUCLE ;SELECCIONO EL DISPLAY SEGUN
CJNE A,#0,NBUCLENO0 ;EL NUMERO DE BUCLE Y MUEVO A
MOV DISPLAY,UNI_MIN ;A LA VARIABLE DISPLAY EL NUMERO
MOV A,#11111110B ;A VISUALIZAR EN EL DISPLAY SELECCIONADO.
ACALL SACAPOR_3 ;LA VARIABLE NBUCLE SE INCREMENTARA
NBUCLENO0: ;EN CADA PASO DEL BUCLE Y SI EN 3
MOV A,NBUCLE ;PASARA A 0.
CJNE A,#1,NBUCLENO1
MOV DISPLAY,DEC_MIN
MOV A,#11111101B
ACALL SACAPOR_3
NBUCLENO1:
MOV A,NBUCLE
CJNE A,#2,NBUCLENO2
MOV A,#11111011B
ACALL SACAPOR_3
MOV DISPLAY,UNI_HOR
NBUCLENO2:

```

```

MOV A,NBUCL
CJNE A,#3,NBUCLNO3

MOV A,#11110111B

ACALL SACAPOR_3

MOV DISPLAY,DEC_HOR

MOV NBUCL,#0

JMP NBUCL0

NBUCLNO3:

INC NBUCL

NBUCL0:

MOV A,DISPLAY ;ESCRIBE EN EL DISPLAY QUE ESTE

CJNE A,#00d,DISPLAYNO0 ;SELECCIONADO EL NUMERO QUE ESTE

MOV A,#00D ;ALMACENADO EN LA VARIABLE

ADD A,#TABLA-$-3 ;DISPLAY.

MOVC A,@A+PC

ACALL SACAPOR_2

DISPLAYNO0:

MOV A,DISPLAY

CJNE A,#01d,DISPLAYNO1

MOV A,#01D

ADD A,#TABLA-$-3

MOVC A,@A+PC

ACALL SACAPOR_2

DISPLAYNO1:

MOV A,DISPLAY

CJNE A,#02d,DISPLAYNO2

```

```

MOV A,#02D

ADD A,#TABLA-$-3

MOVC A,@A+PC

ACALL SACAPOR_2

DISPLAYNO2:

MOV A,DISPLAY

CJNE A,#03d,DISPLAYNO3

MOV A,#03D

ADD A,#TABLA-$-3

MOVC A,@A+PC

ACALL SACAPOR_2

DISPLAYNO3:

MOV A,DISPLAY

CJNE A,#04d,DISPLAYNO4

MOV A,#04D

ADD A,#TABLA-$-3

MOVC A,@A+PC

ACALL SACAPOR_2

DISPLAYNO4:

MOV A,DISPLAY

CJNE A,#05d,DISPLAYNO5

MOV A,#05D

ADD A,#TABLA-$-3

MOVC A,@A+PC

ACALL SACAPOR_2

DISPLAYNO5:

```

```

MOV A,DISPLAY
CJNE A,#06d,DISPLAYNO6
MOV A,#06D
ADD A,#TABLA-$-3
MOVC A,@A+PC
ACALL SACAPOR_2
DISPLAYNO6:
MOV A,DISPLAY
CJNE A,#07d,DISPLAYNO7
MOV A,#07D
ADD A,#TABLA-$-3
MOVC A,@A+PC
ACALL SACAPOR_2
DISPLAYNO7:
MOV A,DISPLAY
CJNE A,#08d,DISPLAYNO8
MOV A,#08D
ADD A,#TABLA-$-3
MOVC A,@A+PC
ACALL SACAPOR_2
DISPLAYNO8:
MOV A,DISPLAY
CJNE A,#09d,DISPLAYNO9
MOV A,#09D
ADD A,#TABLA-$-3
MOVC A,@A+PC

```

```

ACALL SACAPOR_2

DISPLAYNO9:

ACALL RECOGE_POR_3

JNB A.6,SIGUE00

JNB A.5,SIGUE00

LJMP VISUALIZA

SIGUE00:

ACALL RECOGE_POR_3

JNB A.5,NO_INC_RELOJ ;LA PUESTA EN HORA.

ACALL RECOGE_POR_3 ;CADA PULSACION DE P3.4 ADELANTA EN UN

JB A.4,PULSADO0 ;MINUTO EL RELOJ

MOV PULSADO,#00D

PULSADO0:

MOV A,PULSADO

CJNE A,#00D,NO_INC

ACALL RECOGE_POR_3

JNB A.4,NO_INC

MOV PULSADO,#01D

MOV A,MINUTOS

CJNE A,#0D,INC_MINUTOS

MOV MINUTOS,#59D

DEC HORAS

LJMP NO_INC

INC_MINUTOS:

DEC MINUTOS

NO_INC:

```

```

NO_INC_RELOJ:

JNB A.6,NO_INC_AL ;SI ESTA PUSADO A.6 SIGUE.

CLR ET0 ;DESABILITA LA INTERRUPCION DEL TEMP0

MOV A,PULSADO2 ;SI ES LA PRIMERA VEZ QUE ENTRA SALVA
CJNE A,#00D,NO_GUARDA ;LAS HORAS Y METE LA HORA DE ALARMA EN
MOV SALVA_HORAS,HORAS ;EL RELOJ.

MOV SALVA_MINUTOS,MINUTOS

MOV HORAS,HORAS_AL

MOV MINUTOS,MINUTOS_AL

NO_GUARDA:

MOV PULSADO2,#01D ;CADA PULSACION DE P3.4 ADELANTA EN UN
ACALL RECOGE_POR_3 ;MINUTO EL RELOJ (LA ALARMA).

JB A.4,PULSADO01

MOV PULSADO,#00D

PULSADO01:

MOV A,PULSADO

CJNE A,#00D,NO_INC_AL0

ACALL RECOGE_POR_3

JNB A.4,NO_INC_AL0

MOV PULSADO,#01D

MOV A,MINUTOS

CJNE A,#0D,INC_MINUTOS_AL

MOV MINUTOS,#59D

DEC HORAS

LJMP NO_INC_AL

INC_MINUTOS_AL:

```

DEC MINUTOS

NO_INC_AL0:

LJMP VISUALIZA ;CIERRA EL BUCLE GENERAL DE VISUALIZACION.

TABLA: db 11000000B,11111001B,10100100B

db 10110000B,10011001B,10010010B

db 10000010B,11111000B,10000000B,10011000B

NO_INC_AL: ;SI LLEGA AQUI ES QUE NO SE CAMBIA LA ALARMA.

MOV A,PULSADO2 ;SI ES LA PRIMERA VEZ QUE ENTRA GUARDA EL

CJNE A,#01D,NO_REC ;VALOR DE LA ALARMA Y RECUPERA EL VALOR DE

MOV PULSADO2,#00D ;LA HORA ANTERIORMENTE SALVADOS.

MOV HORAS_AL,HORAS

MOV MINUTOS_AL,MINUTOS

MOV HORAS,SALVA_HORAS

MOV MINUTOS,SALVA_MINUTOS

SETB ET0 ;HABILITO LA INTERRUPCION DEL TEMP0.

NO_REC:

ACALL PARA_ALARMA

MOV A,MINUTOS

CJNE A,MINUTOS_AL,SIGUE02

MOV A,HORAS

CJNE A,HORAS_AL,SIGUE02

ACALL SUENA_ALARMA

SIGUE02:

LJMP VISUALIZA ;CIERRO EL BUCLE GENERAL DE VISUALIZACION.

,*****

;FIN DEL PROGRAMA PRINCIPAL.

,*****

FINTEMP0: ;AQUI VIENE CUANDO TERMINA DE CONTAR EL TEMPORIZADOR.

MOV TL0,TEMP0L ;LE METO AL TEMP0 LOS VALORES QUE TIENE QUE CONTAR.

MOV TH0,TEMP0H

SETB TR0 ;PONGO EN MARCHA EL TEMP0.

INC CONTEMP

DJNZ CONTEMP,NOSEG ;INCREMENTA EL RELOJ EN 1 SEGUNDO

MOV CONTEMP,#03D ;#200D ;SI HA PASADO UN SEGUNDO Y LOS MINUTOS

INC SEGUNDOS

DJNZ SEGUNDOS,NOMIN ;Y LAS HORAS SI CORRESPONDE.

MOV SEGUNDOS,#59D

INC MINUTOS

DJNZ MINUTOS,NOHOR

MOV MINUTOS,#59D

INC HORAS

DJNZ HORAS,NODIA

MOV HORAS,#23D

SJMP VOLVER

NODIA:

DEC HORAS

SJMP VOLVER

NOHOR:

DEC MINUTOS

SJMP VOLVER

NOMIN:

DEC SEGUNDOS

SJMP VOLVER

NOSEG:

DEC CONTEMP

VOLVER:

RETI

,*****

SACAPOR_2: ;SACA LOS VALORES DEL ACUMULADOR POR EL PUERTO 2.

; MOV P2,A

ENVIA_POR 2

RET

,*****

SACAPOR_3: ;SACA LOS VALORES DEL ACUMULADOR POR EL PUERTO 3.

; MOV P3,A

ENVIA_POR 3

RET

,*****

RECOGE_POR_3: ;RECOGE EN EL ACUMULADOR LO QUE HABIA EN EL PUERTO 3.

MOV A,#FFH

ACALL SACAPOR_3

; MOV A,P3

RECIBE_POR 3

RET

,*****

PARA_ALARMA:

MOV A,#F0H

ENVIA_POR 1

RET

,*****

SUENA_ALARMA:

MOV A,#FFH

ENVIA_POR 1

RET

,*****

END

PROBLEMAS, ANECDOTAS Y CURIOSIDADES:

Han sido múltiples los quebraderos de cabeza que nos ha ocasionado la programación de este programa, la mayoría de los cuales consistían en pequeñas chorradas como no poner el dichoso # o confusiones al introducir datos en hexadecimal cuando debían ser decimales, etc, etc. Estos pequeños fallos que en principio parecen sin importancia son los que mayores problemas nos han ocasionado, pues resultaban extremadamente difíciles de localizar, ocasionando graves crisis depresivas a los programadores.

Normalmente tarde o temprano los fallos se iban encontrando, pero en ocasiones ocurrió que al no lograr encontrar el error y ya totalmente desesperados optamos por abandonar ese camino y intentarlo de otra forma, que, aunque mas complicaba funcionaba, así nos ocurrió al intentar sacar la alarma por el pin P3.7, tras 3 días de fracasos optamos por sacarla por P1.0.

CARACTERISTICAS TECNICAS:

Nombre del programa: Peluco

Nombre del archivo de codigo fuente: Pelucon.asm

Nombre del archivo ejecutable: Pelucon.hex

Recursos utilizados: Temporizador 0 en modo 16 bits

Pin 0 del puerto 1 (alarma)

Puerto 2 (displays)

Puerto 3 (0-3 seleccion de display ; 4-6 pulsadores)

Editor utilizado: SE , EDIT

Compilador: X8051

Lincador: ADLINK

Simulador: SIMU8051

Soporte físico: PLACA DISEN (Placa preparada para fines educativos)

Programa de comunicación con disen: DIS (adjjj!)

Bibliografía utilizada: Apuntes sobre ðc8051 y el libro ðc 8051

Proyecto de prácticas

Pg 15