



**CENTRO DE CIENCIAS BÁSICAS**  
**DEPARTAMENTO DE SISTEMAS ELECTRÓNICOS**  
**ORGANIZACIÓN COMPUTACIONAL**  
**4° "A"**

**PROYECTO FINAL: RELOJ**

**M. en CC. Juan Pedro Cisneros Santoyo**

**Alumnos:**

**Espinoza Sánchez Joel Alejandro**

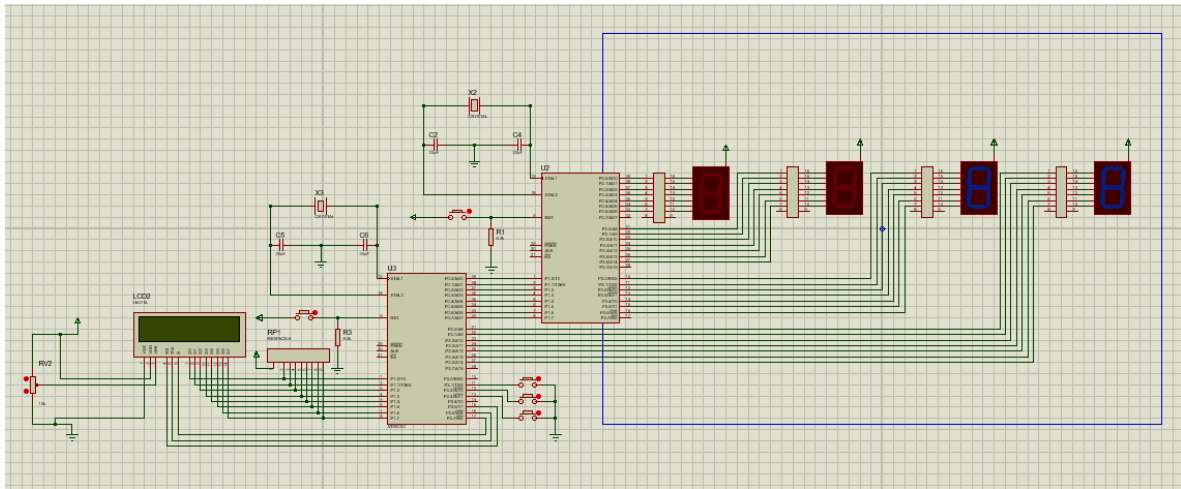
**Gómez Garza Dariana**

**González Arenas Fernando Francisco**

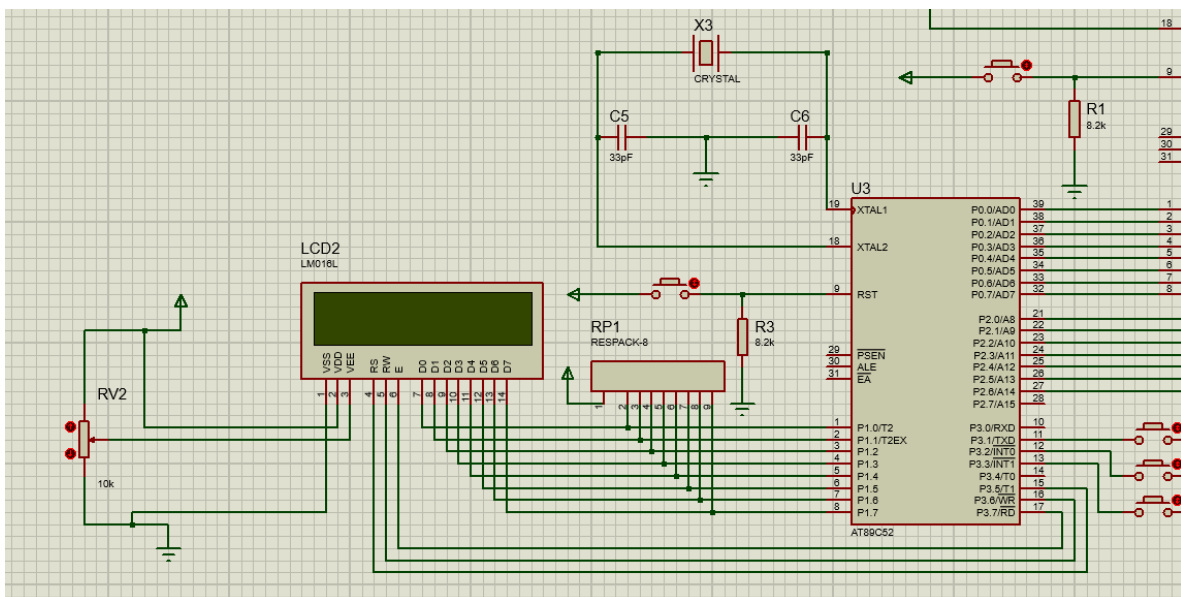
**Fecha de Entrega:** Aguascalientes, Ags., 3 de julio de 2020

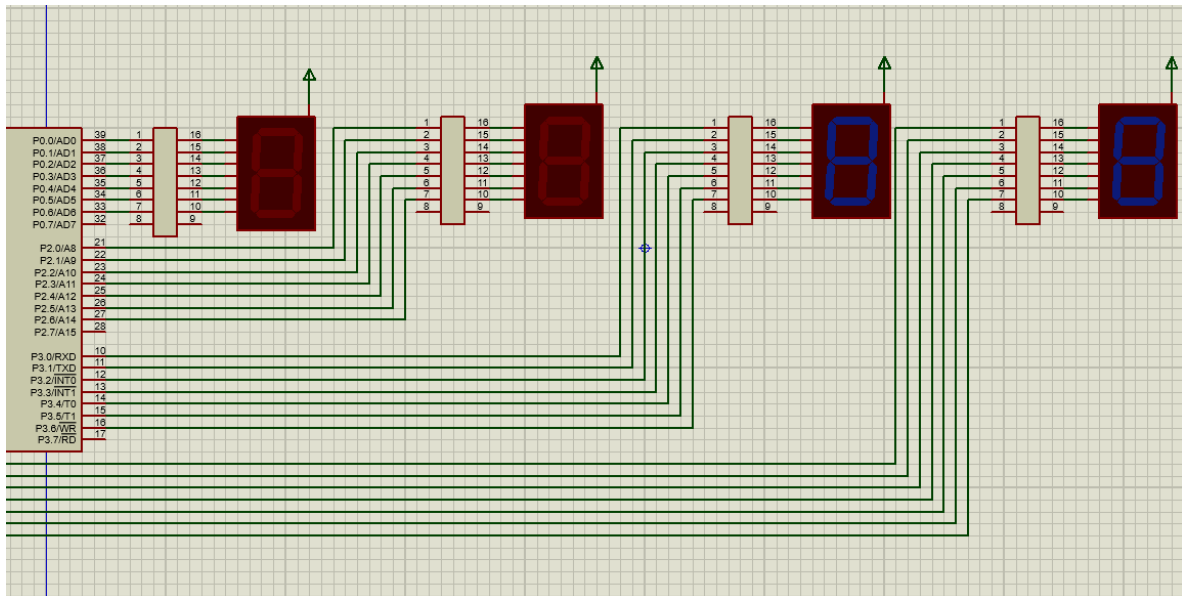
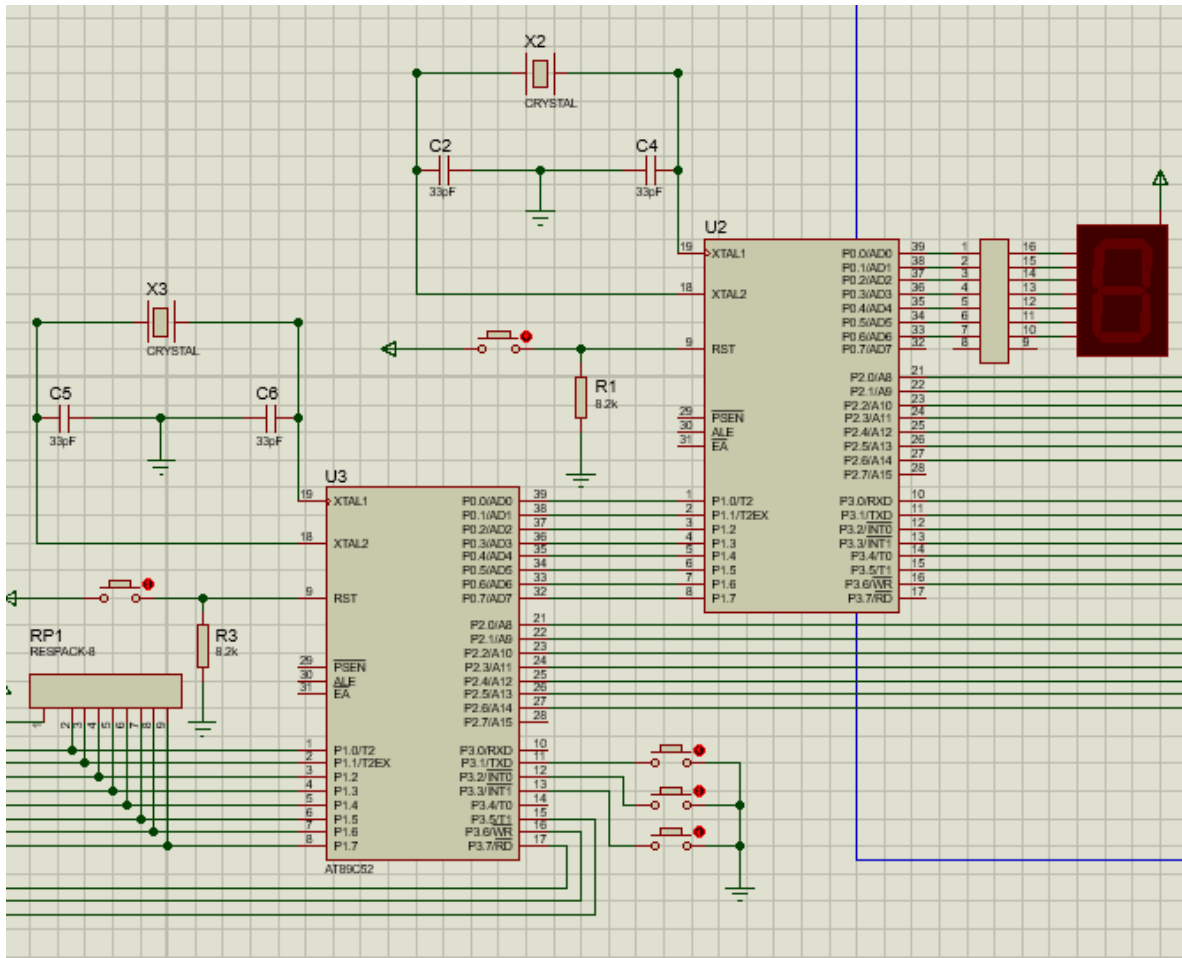
# Proyecto Final: Reloj

El equipo planteó el siguiente circuito, pues creíamos que con el uso de displays de 7 segmentos se conseguirían mejores resultados, sin embargo, no manejábamos el uso de paquetes de múltiples displays, por lo que para poder hacer uso de cuatro displays, una LCD y botones de usuario, se tuvo que proponer un microcontrolador anexo a otro, es decir, se hicieron dos programas y se compilaron en el circuito que se llama "ProyectoV1", sin embargo, el denominado "Proyecto" fue la base principal para obtener las ideas que se desarrollaron a continuación:

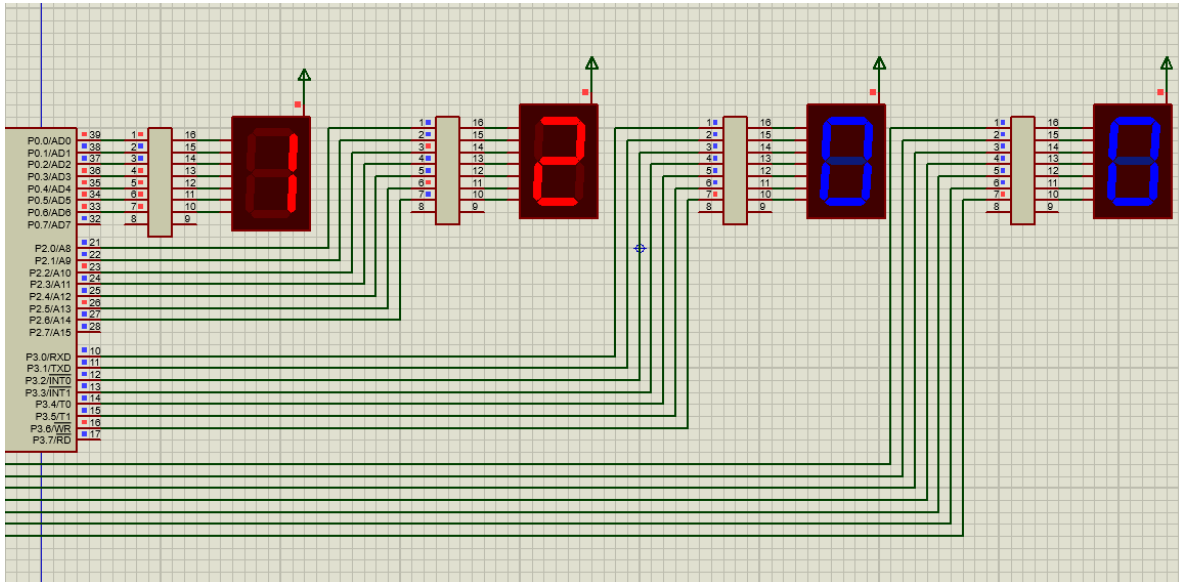


Imágenes más detalladas del circuito se presentan a continuación:





Por ello mismo, también se aprecia el estado inicial de los displays al comienzo de la ejecución:



Antes de mencionar el código, en los archivos de Keil se presentan algunos avances graduales del código. Finalmente, se anexan los dos códigos necesarios para cada microcontrolador. El microcontrolador que posee la LCD y los botones se le denotó como “Micro 1” y el otro se le llamó “Micro 2”.

En el archivo adjunto, el código de Micro 1 se llama “ProyectoV3\_1” mientras que el código de Micro 2 se llama “ProyectoV3\_2” pero existen problemas con las interrupciones que desconocemos que detienen totalmente el funcionamiento del programa (incluso lo rompen). El ejemplo del código llamado “ProyectoV3\_1Alternativo” demuestra que previo a la introducción de las interrupciones, la idea de la realización del cronómetro funcionaba correctamente; error que no supimos cómo solucionar.

## Micro 1

```
;A
;B
;R0
;R1 Ayuda a la tabla
;R2
;R3
;R4 Va a delimitar si inicia o para el cronómetro
;R5
;R6 Guardará el valor del display 4 cuando entre al cronómetro
;R7 Extiende un delay de un segundo a 60
en equ P3.7
rs equ P3.5
rw equ P3.6
datos equ P1
ORG 0000H ;Reset
```

```

    LJMP iniciowo

ORG 0003H ;Interrupción externa 0
    ;Código para la interrupción
    ACALL print_cronom
    CJNE R4,#01H,start ;Salta a start para empezar el cronómetro
    MOV A,P3
    CJNE A,#02H,stop
    AJMP final
    stop: ;Acude a stop si ya se pasó por start
    MOV R4,#00H
    RETI

    start:
    MOV R4,#01H
    MOV A,P2
    MOV R2,A
    MOV P0,01H
    inicio:
    MOV R2,#00H
    ACALL iniciar_reclock

    ciclo2:
    MOV R1,#00H

    ciclo1:
    MOV R0,#20
    MOV R7,#60

    ciclock:
    CJNE R0,#0,ciclock
    MOV A,R1
    MOV DPTR,#display_7seg ;Apuntar a la dirección de la etiqueta
display_7seg
    MOVC A,@A+DPTR
    MOV P2,A
    /*delay_min:
    ACALL delay
    ACALL delay
    DJNZ R7,delay_min*/
    INC R1 ;Incrementará el display 4
    CJNE R1,#0AH,ciclo1
    AJMP ciclo2

```

```

    iniciar_reclock:
    MOV P2,#0x40 ;Prende el cuarto display
    MOV P0,#0x00 ;Deja en 0 el puerto de comunicación
    RET

    final: ;Se accede por la verificación del botón que regresa
al programa de inicio
    MOV P0,#00H
    RETI

ORG 0013H ;Interrupción externa 1
    ;Código para la interrupción
    ACALL iniciar_reclock1

    iniciar_reclock1:
    MOV P2,#0x40 ;Prende el cuarto display
    MOV P0,#0x02 ;Manda un 2 por el puerto de comunicación al
micro 2
    RET
    RETI

ORG 001BH ;Interrupción por desborde (Overflow) del timer 1
    MOV TH1,#004BH
    MOV TL1,#00FDH
    DEC R0
    RETI ;Retorno de interrupción

ORG 0030H
    iniciowo:
    MOV R2,#00H
    ACALL iniciar_reloj ;Inicia el display asociado a este micro
y envía la información de inicio al puero de transmisión del micro
2
    ACALL iniciar_lcd
    SETB EA
    SETB ET1
    ACALL init_timer1
    ACALL initint0_config

    ciclowo2: ;Toda una iteración de 0 a 9
    ACALL print_reloj
    MOV R1,#00H

    ciclowo1: ;Una iteración de un número

```

```

MOV R0,#20
MOV R7,#60

ciclo:
CJNE R0,#0,ciclo
MOV A,R1
MOV DPTR,#display_7seg ;Apuntar a la dirección de la etiqueta
display_7seg
MOVC A,@A+DPTR
MOV P2,A
delay_min:
ACALL delay
ACALL delay
DJNZ R7,delay_min
INC R1 ;Incrementará el display 4
CJNE R1,#0AH,ciclowo1
AJMP ciclowo2

print_cronom: ;Imprimimos a la LCD la leyenda "Cronometro"
;Regreso a casa
ACALL en_h
MOV A,#02H
MOV datos,A
ACALL en_l

ACALL en_h
ACALL rs_h
MOV A,#67 ;C
MOV datos,A
ACALL en_l

ACALL en_h
ACALL rs_h
MOV A,#82 ;R
MOV datos,A
ACALL en_l

ACALL en_h
ACALL rs_h
MOV A,#79 ;0
MOV datos,A
ACALL en_l

ACALL en_h

```

```
ACALL rs_h
MOV A,#78 ;N
MOV datos,A
ACALL en_l
```

```
ACALL en_h
ACALL rs_h
MOV A,#79 ;0
MOV datos,A
ACALL en_l
```

```
ACALL en_h
ACALL rs_h
MOV A,#77 ;M
MOV datos,A
ACALL en_l
```

```
ACALL en_h
ACALL rs_h
MOV A,#69 ;E
MOV datos,A
ACALL en_l
```

```
ACALL en_h
ACALL rs_h
MOV A,#84 ;T
MOV datos,A
ACALL en_l
```

```
ACALL en_h
ACALL rs_h
MOV A,#82 ;R
MOV datos,A
ACALL en_l
```

```
ACALL en_h
ACALL rs_h
MOV A,#79 ;0
MOV datos,A
ACALL en_l
RET
```

```
print_reloj: ;Imprimimos a la LCD la leyenda "Reloj"
;Regreso a casa
```



```
ACALL en_h
MOV A,#02H
MOV datos,A
ACALL en_l
```

```
ACALL en_h
ACALL rs_h
MOV A,#82 ;R
MOV datos,A
ACALL en_l
```

```
ACALL en_h
ACALL rs_h
MOV A,#69 ;E
MOV datos,A
ACALL en_l
```

```
ACALL en_h
ACALL rs_h
MOV A,#76 ;L
MOV datos,A
ACALL en_l
```

```
ACALL en_h
ACALL rs_h
MOV A,#79 ;0
MOV datos,A
ACALL en_l
```

```
ACALL en_h
ACALL rs_h
MOV A,#74 ;J
MOV datos,A
ACALL en_l
RET
```

```
init_timer1:
MOV TMOD,#10H ;Activamos el modo 1 del timer 1
MOV TH1,#004BH
MOV TL1,#00FDH
SETB TR1
RET
```

```
iniciar_reloj:
```

```

MOV P2,#0x40 ;Prende el cuarto display
MOV P0,#0x00 ;Deja en 0 el puerto de comunicación con el
segundo micro
RET

```

```

display_7seg:

```

```

DB 40H ;0
DB 79H ;1
DB 24H ;2
DB 30H ;3
DB 19H ;4
DB 12H ;5
DB 02H ;6
DB 78H ;7
DB 00H ;8
DB 18H ;9

```

```

delay:                ;2 ciclos Retardo.5 segundos          =2
MOV R6,#0FAH ;250d    ;1ciclo                                =1
d1:
MOV R7,#0F9H ;249d    ;1ciclo      1*250=250
NOP                        ;1ciclo                                =250
NOP                        ;1ciclo                                =250
NOP                        ;1ciclo                                =250
NOP                        ;1ciclo                                =250
NOP                        ;1ciclo                                =250
d2:
NOP                        ;1ciclo      1*249*250=62250
NOP                        ;1ciclo                                =62250
NOP                        ;1ciclo                                =62250
NOP                        ;1ciclo                                =62250
NOP                        ;1ciclo                                =62250
NOP                        ;1ciclo                                =62250
DJNZ R7,d2                ;2ciclos      2*249*250=124500
DJNZ R6,d1                ;2ciclos      2*250=500
RET                        ;2cilos          =2
                                ;total=500005

```

```

initint0_config:

```

```

SETB EX0
SETB EA
RET

```

```

iniciar_lcd:

```

```

ACALL rw_1

```

```
ACALL rs_1
```

```
;Comienza la configuración de encendido de la LCD
```

```
ACALL en_h
```

```
MOV A,#38H
```

```
MOV datos,A
```

```
ACALL en_l
```

```
ACALL en_h
```

```
MOV A,#38H
```

```
MOV datos,A
```

```
ACALL en_l
```

```
ACALL en_h
```

```
MOV A,#38H
```

```
MOV datos,A
```

```
ACALL en_l
```

```
ACALL en_h
```

```
MOV A,#38H
```

```
MOV datos,A
```

```
ACALL en_l
```

```
;Apaga la pantalla
```

```
ACALL en_h
```

```
MOV A,#08H
```

```
MOV datos,A
```

```
ACALL en_l
```

```
;Limpiar pantalla
```

```
ACALL en_h
```

```
MOV A,#01H
```

```
MOV datos,A
```

```
ACALL en_l
```

```
;Modo de entrada
```

```
ACALL en_h
```

```
MOV A,#06H
```

```
MOV datos,A
```

```
ACALL en_l
```

```
;Encender la pantalla
```

```
ACALL en_h
```

```
MOV A,#0FH
```

```

MOV datos,A
ACALL en_1
RET

en_1:
CLR en
ACALL delay
RET

en_h:
SETB en
ACALL delay
RET

rs_1:
CLR rs
ACALL delay
RET

rs_h:
SETB rs
ACALL delay
RET

rw_1:
CLR rw
ACALL delay
RET

rw_h:
SETB rw
ACALL delay
RET
END

```

## Micro 2

```

ORG 0000H ;Reset
LJMP inicio

ORG 001BH ;Interrupción por desborde (Overflow) del timer 1
MOV TH1,#004BH
MOV TL1,#00FDH
DEC R0
RETI ;Retorno de interrupción

```

[illegible]

```

    ACALL delay
    ACALL delay
    ACALL delay
    ACALL delay
    ACALL delay
    ACALL delay
    ACALL delay
    DJNZ R7,delay_min
    INC R1 ;Incrementará el display 3
    CJNE R1,#06H,ciclo1
    INC R2 ;Incrementará el display 2
    MOV A,R2
    MOV DPTR,#display_7seg ;Apuntar a la dirección de la etiqueta
display_7seg
    MOVC A,@A+DPTR
    MOV P2,A
    AJMP ciclo2

interrupcion0:
    MOV R4,#01H
    MOV A,P2
    MOV R2,A
    MOV P0,01H
iniciowo:
    MOV R2,#00H
    ACALL iniciar_reloj
    AJMP ciclowo2

interrupcion1:
    MOV P0,#0x40 ;Prende el primer display en 0
    MOV P2,#0x40 ;Prende el segundo display en 0
    MOV P3,#0x40 ;Prende el tercer display en 0

ciclowo2:
    MOV R1,#00H

ciclowo1:
    MOV R0,#20
    MOV R7,#60

ciclowo:
    CJNE R0,#0,ciclowo
    MOV A,R1

```

```
MOV DPTR,#display_7seg ;Apuntar a la dirección de la etiqueta  
display_7seg
```

```
MOVC A,@A+DPTR
```

```
MOV P2,A
```

```
delay_minimo:
```

```
ACALL delay
```

```
ACALL delay
```

```
ACALL delay
```

```
ACALL delay
```

```
ACALL delay
```

```
ACALL delay
```

```
ACALL delay
```

```
ACALL delay
```

```
ACALL delay
```

```
ACALL delay
```

```
ACALL delay
```

```
ACALL delay
```

```
ACALL delay
```

```
ACALL delay
```

```
ACALL delay
```

```
ACALL delay
```

```
ACALL delay
```

```
ACALL delay
```

```
ACALL delay
```

```
ACALL delay
```

```
DJNZ R7,delay_minimo
```

```
INC R1 ;Incrementará el display 4
```

```
CJNE R1,#0AH,ciclowo1
```

```
AJMP ciclowo2
```

```
init_timer1:
```

```
MOV TMOD,#10H ;Activamos el modo 1 del timer 1
```

```
MOV TH1,#004BH
```

```
MOV TL1,#00FDH
```

```
SETB TR1
```

```
RET
```

```
iniciar_reloj:
```

```
MOV P0,#0x79 ;Prende el primer display
```

```
MOV P2,#0x24 ;Prende el segundo display
```

```
MOV P3,#0x40 ;Prende el tercer display
```

```
RET
```

```
display_7seg:
```

```

DB 40H ;0
DB 79H ;1
DB 24H ;2
DB 30H ;3
DB 19H ;4
DB 12H ;5
DB 02H ;6
DB 78H ;7
DB 00H ;8
DB 18H ;9

```

```

                delay:                ;2 ciclos Retardo.5 segundos
=2
MOV R6,#0FAH ;250d ;1ciclo                =1
d1:
MOV R7,#0F9H ;249d ;1ciclo                1*250=250
NOP                                ;1ciclo                =250
NOP                                ;1ciclo                =250
NOP                                ;1ciclo                =250
NOP                                ;1ciclo                =250
NOP                                ;1ciclo                =250
d2:
NOP                                ;1ciclo                1*249*250=62250
NOP                                ;1ciclo                =62250
NOP                                ;1ciclo                =62250
NOP                                ;1ciclo                =62250
NOP                                ;1ciclo                =62250
NOP                                ;1ciclo                =62250
DJNZ R7,d2                ;2ciclos                2*249*250=124500
DJNZ R6,d1                ;2ciclos                2*250=500
RET                                ;2cilos                =2
                                ;total=500005
END

```