Microprocesadores - Laboratorio N°1.

Presentado por: Jessica Lorena Arias Ríos – 2185287.

Samantha Gallego Sánchez – 2185162.

Grupo: 4.

Descripción de la práctica.

La práctica de laboratorio consiste en el desarrollo de una aplicación donde se visualicen los números entre 0 y 9 en dos display 7 segmentos, controlando uno con la interrupción de un temporizador que se ejecutara cada segundo. Y el cambio del segundo digito en el otro display, corresponde a un evento externo generado por un pulsador cada vez que es oprimido.

Para lo cual se presentan dos versiones de la solución, en la primera la comunicación entre el microcontrolador y los dispositivos externo se realiza a través de los puertos paralelos del microcontrolador. En la segunda, los dispositivos externos acceden como dispositivos de memoria de datos externa, a través de los buses de datos, dirección y control.

Para la implementación de las soluciones se utiliza un microcontrolador ATMega2560, se realiza el código de programación de este en el programa Microchip Studio, a través de lenguaje Assembler, después se procede a realizar la simulación en Proteus de cada una de las versiones de la solución. Se utiliza una frecuencia de 16MHz para el microcontrolador.

Análisis y cálculos de la solución propuesta.

- Versión 1 de la solución, conexión de dispositivos externos a través de puertos paralelos.

En primer lugar, se hacen las definiciones iniciales, para lo que se plantean dos variables temporales, una variable donde se guardan los segundos transcurridos, otra donde se guarda el número de pulsos entre 0 y 9 y una constante de valor 1.

Luego se definen los vectores de interrupción, en primer lugar, el inicio en la posición de programa 0, donde se configuran inicialmente las variables y los periféricos, luego la interrupción externa y la interrupción del temporizador.

En el inicio se realiza la configuración de las variables y de los periféricos, inicializan los segundos transcurridos y los pulsos iniciales en 0, además de definir el valor de la constante como 1. Después se configuran los puertos A, B y D como puertos de salida, y se inicializan los puertos A y B con el valor de 0 en código 7 segmentos, de cátodo común, y se inicializa el puerto D como 1.

Posteriormente se realiza la configuración del temporizador, de forma que este se desborde en un segundo. Para lo que se realizaron los cálculos pertinentes para definir el valor inicial del temporizador con el registro TCNT1H y TCNT1L, teniendo en cuenta que se escogió trabajar con un divisor de frecuencia de 1024, y que el temporizador con el que se trabajó es el 1, por lo que posee 16 bits.



número de periodos requeridos =
$$\left(\frac{16MHz}{1024}\right)$$

número de periodos requeridos = 15.625
 $TCNT1 = 2^{16} - 15.625$

Por lo que se le asigno a TCNT1H el valor de 0xC2 y a TCNT1L el valor de 0xF7, posteriormente se definió el divisor de frecuencia en 1024, otorgando le valor de 101 a sus bits menos significativos.

TCNT1 = 49.911 = 0xC2F7

Luego se activaron las interrupciones del temporizador, y se configuró la interrupción externa por flanco de bajada, asignándole el valor de 0x02 al registro EICRA, de modo que cada que se oprima el pulsador, este le asigne al bit menos significativo del puerto D el valor de 0 y se active la interrupción externa. Por último, se activó el habilitador global de interrupciones.

Pasando la tarea de fondo, al momento de ocurrir la interrupción de desbordamiento del temporizador, lo primero que se realiza es volver a iniciar el temporizador en 0xC2F7, para que se cuente el nuevo segundo de forma inmediata, posterior se utiliza la variable temporal, donde se guardan los segundos transcurridos y se llama la subrutina de aumentar, que se explicará adelante, y como su nombre lo indica, aumenta en uno la variable temporal y mantiene la variable entre 0 y 9, al salir, se le asigna el nuevo valor a la variable de segundos y se envía a la subrutina hex_7seg que se encarga de transformar el valor de hexadecimal a código de 7 segmentos con cátodo común, valor que se guarda en la segunda variable temporal y se manda al puerto A, que está conectado directamente con el display 7 segmentos.

Una vez finalizado, vuelve a la tarea de fondo, y al momento que se realiza una pulsación con el botón, conectado al bit menos significativo del puerto D, el bit pasa a ser 0, por lo que ocurre un flanco de bajado y se activa la interrupción externa, que se encarga de mandar el número de pulsos a la subrutina aumentar, y posterior de guardarse el número de pulsos, se transforma a código 7 segmentos para pasar al display conectado en el puerto B.

Ampliando un poco más en las subrutinas, la subrutina aumentar, aumenta la constate uno al registro temporal, posterior, lo compara con 10, si el resultado de la suma es mayor o igual a 10, carga el número 0 en el registro, para iniciarlo de nuevo y sale de subrutina, pues los display van de 0 a 9, si el número es menor, sale de inmediato de la subrutina.

La subrutina hex_7seg, se encarga de cambiar el número de entrada de hexadecimal a código 7 segmentos, esto lo hace a través de una lista de instrucciones que funcia como una sentencia de Switch, donde depende el número que entre, realiza una carga diferente que es el numero ya transformado para enviarlo directamente al display 7 segmentos cátodo común.

- Versión 2 de la solución, conexión de dispositivos externos conectados como dispositivos de memoria externa.

El programa transcurre igual que en la primera versión, con las mismas definiciones iniciales y los mismos vectores de interrupción, al igual que en el inicio, hay algunas definiciones iguales, como los valores iniciales de segundos, pulsos y de la constante K.

Sin embargo, en los registros X, Y se guardan las direcciones de memoria externa donde estarán conectados los dispositivos externos, X=0x8000 (Display de los segundos), Y=0x8100 (Display del número de pulsos).

Se habilitan los puertos A y B para habilitar los pines de memoria externa, de forma que queden funcionando como buses de datos, dirección y control. En la simulación de Proteus se conecta un decodificador habilitado por las salidas A8 y A9 del bus de direcciones, además de ser controlado por la salida Write del bus de control, este decodificador se encarga de habilitar dos latches a los que se conecta los puertos ADO – AD7, y la salida de estos se conecta a los display 7 segmentos.

Después la configuración del temporizador y la habilitación de las interrupciones se realiza igual que en la versión anterior.

Las interrupciones, se comportan de igual manera que en la versión 1, con la diferencia que ahora el dato final, en código 7 segmentos no está de salida en los puertos, sino que se guarda en su dirección correspondiente.

Diagrama de flujo y circuito.

Se desarrollaron los respectivos diagramas de flujo para cada una de las versiones de solución, teniendo en cuenta que las interrupciones se comportan de igual manera para ambos casos, se desarrolló un solo diagrama para estos. Para la primera versión de solución se utilizó el diagrama de la figura 1, este diagrama es el principal de esta versión.

Primero se recibe la señal del pulsador, una vez recibida empieza la subrutina INICIO, luego de esto se llega a un bucle donde se espera a alguna de las interrupciones para continuar con las subrutinas AUMENTAR y HEX_7SEG, una vez terminadas las subrutinas se carga el valor en los Puertos A o B dependiendo de cuál de las interrupciones se llevó a cabo.

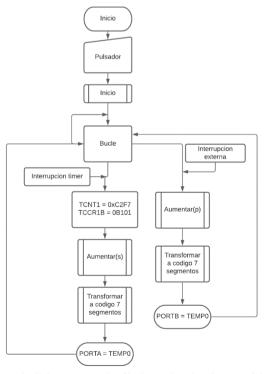


Figura 1. Diagrama de flujo principal (versión 1).

Para la subrutina de INICIO se realizó el flujograma que se observa en la figura 2, donde se les dan los valores iniciales a las variables, se asignan los DDRA y DDRB como puertos de salida, se da inicio al temporizador 1 y se activan las interrupciones.

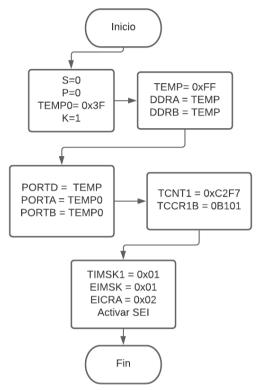


Figura 2. Diagrama de flujo subrutina INICIO (versión 1).

En el caso de la subrutina AUMENTAR se desarrolló el diagrama de flujo de la figura 3, como se mencionó anteriormente, este diagrama se utiliza en ambas versiones de solución, esta subrutina es la encargada aumentar el contador tanto de los segundos como de los pulsos.

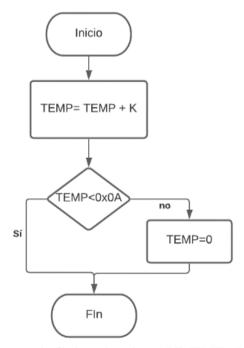


Figura 3. Diagrama de flujo subrutina AUMENTAR (versión 1 y 2).

La subrutina HEX_7SEG es la encargada de transformar el valor de hexadecimal a código de siete segmentos en ambas versiones de solución para luego verlo reflejado en los display 7 segmentos, de acuerdo con el número en hexadecimal que traiga el código así mismo se realizara la transformación al código para el display, como se observa en la figura 4.

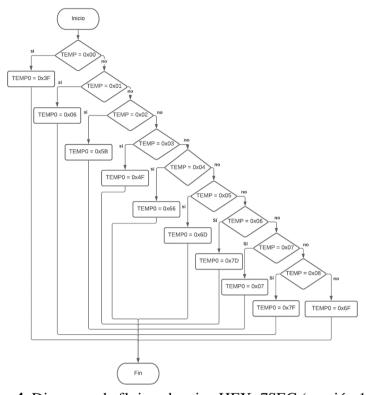


Figura 4. Diagrama de flujo subrutina HEX_7SEG (versión 1 y 2).

A continuacion en la figura 5 se presenta el circuito de la version 1 realizado con la herramienta proteus donde se puede observar los dos display 7 segmentos conectdos a los puertos A y B, donde el conectado al puerto B es el display que cambia su valor con una interrupcion externa en este caso el boton.

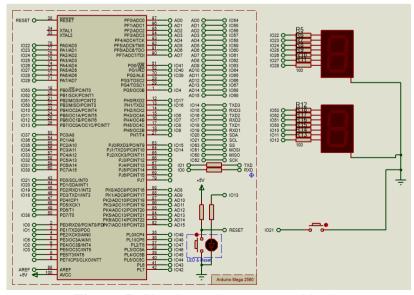


Figura 5. Circuito en Proteus (versión 1).

Para la segunda version de solucion se utilizo el diagra de la figura 6, el cual funciona de la misma manera que la version 1 con el unico cambio de que una vez terminadas las subrutinas se carga el valor en los puertos de salida X y Y.

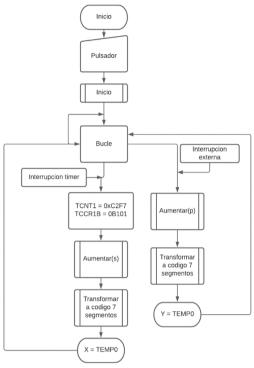


Figura 6. Diagrama de flujo principal (versión 2).

Para la subrutina de inicio de la versión 2 de solución se utilizó el flujograma de la figura 7, en este se puede observar cada uno de los pasos que se realizan en esta subrutina como lo son la asignación de valores iniciales, la definición de los puertos de salida y el inicio del timmer.

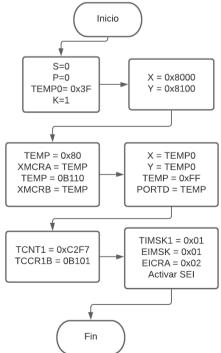


Figura 7. Diagrama de flujo subrutina INICIO (versión 2).

Por último, se presenta el circuito de la versión 2 de solución desarrollada con la herramienta Proteus, donde se puede ver el uso de memorias externas y la conexión de ambos display de siete segmentos.

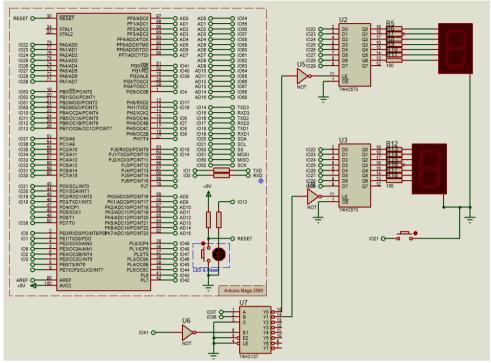


Figura 5. Circuito en Proteus (versión 2).

Resultados de simulación y análisis de estos.

Se probo el resultado de los circuitos en el programa de Proteus, el resultado después de transcurrido 2 segundos y sin haber oprimido el botón pulsador, de la versión 1 de la solución se puede ver en la figura 6.

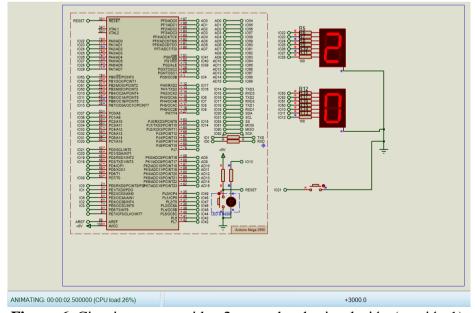


Figura 6. Circuito transcurridos 2 segundos de simulación (versión 1).

Posterior, se procedió a presionar una vez el botón, y se ve que en el display encargado de mostrar los números de 0 al 9 de los pulsos, también posee el valor correspondiente. Además de mostrarse que pasados 18 segundos el display de los segundos muestra el numero 8, pues la cuenta del display se reinicia en el 10.

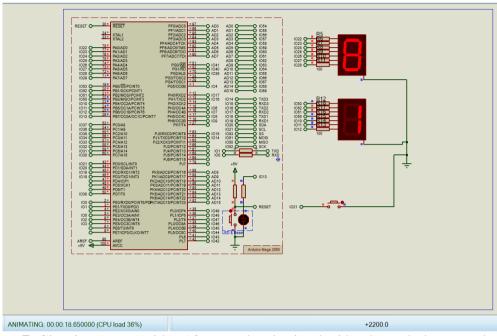


Figura 7. Circuito transcurridos 18 segundos de simulación, después de una pulsación (versión 1).

De igual forma, se probo la versión 2 de la solución, donde se ve que el display encargado de mostrar los segundos transcurridos, muestra el numero 5, al haber pasado 5 segundos de animación del circuito, y 0 al no haberse oprimido el botón pulsador.

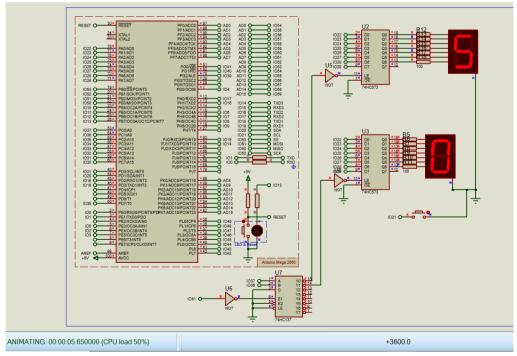


Figura 8. Circuito transcurridos 5 segundos de simulación (versión 2).

Por último, se probó la versión 2 de la solución, donde se ve que el display encargado de mostrar los segundos transcurridos, muestra el número 2, al haber pasado 12 segundos de animación del circuito, pues este también se reinicia en 10, al pedirse solo los números del 0 al 9, y 1 en el display de las pulsaciones, al haberse oprimido una sola vez el botón pulsador.

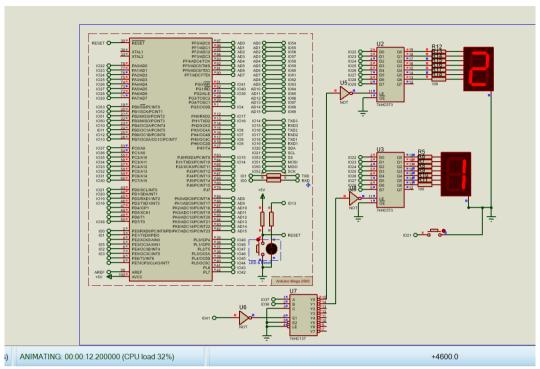


Figura 9. Circuito transcurridos 12 segundos de simulación, después de una pulsación (versión 2).

De lo cual se puede decir que ambas versiones del circuito funcionan de forma adecuada y cumplen con el objetivo, el cual era mostrar en dos display los dígitos del 0 al 9, uno de los cuales se modifica después de cada segundo, y en el otro con cada pulso generado por oprimir el botón del circuito.

Código de programa.

A continuación, se presenta el código de programa de la versión 1 de solución.

```
; VECTORES DE INTERRUPCION
.ORG 0x0000
    JMP INICIO
    .ORG 0x0002
    JMP INT_EXTERNA
    .ORG 0x0028
    JMP INT_TEMP
; ------
; CONFIGURACIÓN DE VARIABLES Y PERIFÉRICOS: PUERTOS A, B y D
INICIO:
            LDI S, 0x00 ; Segundos Initial LDI P, 0x00 ; Pulsos inicio 0.

LDI TEMPO, 0x3F ; 0 en 7segmentos cátodo común LDI K, 0x01 ; Constante K=1
            LDI S, 0x00
                        ; Segundos inicio 0.
            LDI TEMP, 0xFF
                        ; Definir puertos A, B y D de salida
            OUT DDRA, TEMP
            OUT DDRB, TEMP
            OUT PORTD, TEMP
                        ; Iniciar D en 1
            OUT PORTA, TEMP0
                        ; Iniciar los puertos en 0 en 7 segmentos
            OUT PORTB, TEMP0
; CONFIGURACIÓN DE PERIFÉRICOS: TIMER 1
LDI TEMP, 0xC2
               ; Iniciar Temporizador en C2F7 =49911, para desbordar 1s
    LDI TEMP0, 0xF7
    STS TCNT1H, TEMP
    STS TCNT1L, TEMP0
                   ; Divisor de frecuencia de 1024
    LDI TEMP, 0B101
    STS TCCR1B, TEMP
; HABILITAR INTERRUPCIONES INT_EXT, TIMER 1 Y GLOBALES
[------
    LDI TEMP, 1<<TOIE1 ; Habilitar interrupción timmer, con desbordamiento 1s.
    STS TIMSK1, TEMP
    LDI TEMP, 1<<INTO ; Habilitar interrupción externa
    OUT EIMSK, TEMP
    LDI TEMP, 0x02
                ; Interrupción por flanco de bajada
    STS EICRA, TEMP
; BUCLE INFINITO: TAREA FONDO
BUCLE: NOP
                        JMP BUCLE
; Subrutina de Atención a la Interrupción del Timmer 1
```

```
INT TEMP:
               LDI TEMP, 0xC2
                             ; Volver a iniciar Temporizador en C2F7, para
desbordar 1s
                LDI TEMP0, 0xF7
               STS TCNT1H, TEMP
               STS TCNT1L, TEMP0
               MOV TEMP, S
               CALL AUMENTAR
               MOV S, TEMP
               CALL HEX_7SEG
               OUT PORTA, TEMP0
                                          ; Display cambia cada segundo.
               RETI
; Subrutina de Atención a la Interrupción Externa 0
INT_EXTERNA: MOV TEMP, P
                          CALL AUMENTAR
                          MOV P, TEMP
                          CALL HEX_7SEG
                          OUT PORTB, TEMP0
                                         ; Display cambia cada pulso.
                          RETI
; ------
; Subrutinas aumentar y transformar a código 7 segmentos
AUMENTAR:
                               ADD TEMP, K
                               CPI TEMP, 0x0A
                               BRMI FIN AUM
                               LDI TEMP, 0x00
          FIN AUM:
                          RET
          HEX 7SEG:
                               CERO:
                                          CPI TEMP,0x00
                                               BRNE UNO
                                               LDI TEMP0, 0x3F
                                               JMP FIN_HEX_7SEG
                               UNO:
                                          CPI TEMP,0x01
                                               BRNE DOS
                                               LDI TEMP0, 0x06
                                               JMP FIN_HEX_7SEG
                                          CPI TEMP,0x02
                               DOS:
                                               BRNE TRES
                                               LDI TEMP0, 0x5B
                                               JMP FIN HEX 7SEG
                               TRES:
                                          CPI TEMP,0x03
                                               BRNE CUATRO
                                               LDI TEMP0, 0x4F
                                               JMP FIN_HEX_7SEG
                               CUATRO:
                                               CPI TEMP,0x04
                                               BRNE CINCO
                                               LDI TEMP0, 0x66
                                               JMP FIN_HEX_7SEG
```

CINCO:

CPI TEMP,0x05

```
JMP FIN HEX 7SEG
                           CPI TEMP,0x06
                    SEIS:
                              BRNE SIETE
                              LDI TEMP0, 0x7D
                              JMP FIN_HEX_7SEG
                    SIETE:
                           CPI TEMP,0x07
                              BRNE OCHO
                              LDI TEMP0, 0x07
                              JMP FIN_HEX_7SEG
                           CPI TEMP,0x08
                    OCHO:
                              BRNE NUEVE
                              LDI TEMP0, 0x7F
                              JMP FIN_HEX_7SEG
                    NUEVE:
                           LDI TEMP0, 0x6F
                              JMP FIN_HEX_7SEG
Por último, se presenta el código de la versión 2 de solución.
; variable segundos pasados en
                              ; Se trabaja display 7
                           ; Crear constante 1
                           ; Variable número pulsos
```

FIN_HEX_7SEG: RET

; Arias_Gallego_Laboratorio1_v2.asm

; Arias_Gallego_Laboratorio1.asm

; DEFINICIONES INICIALES

segmentos cátodo común .DEF K=R19

.DEF P=R20

; VECTORES DE INTERRUPCION

.ORG 0x0000 JMP INICIO

.ORG 0x0002

.DEF TEMP=R16 .DEF TEMP0=R17 .DEF S=R18

; FIN DEL PROGRAMA

; Autor: Jessi

; Autor: Jessi

hex.

BRNE SEIS LDI TEMP0, 0x6D

```
JMP INT EXTERNA
     .ORG 0x0028
     JMP INT TEMP
; CONFIGURACIÓN DE VARIABLES Y UBICACIONES DE MEMORIA EXTERNA PARA GUARDAR DATOS.
INICIO:
               LDI S, 0x00
                                   ; Segundos inicio 0.
               LDI P, 0x00
                                   ; Pulsos inicio 0.
               LDI TEMPO, 0x3F
                                   ; 0 en 7segmentos catodo comun
               LDI K, 0x01
                                   ; Constante K=1
               LDI R27, 0x80
                             ; Registro X=0x8000, dirección display
segundos.
               LDI R26, 0X00
               LDI R29, 0X81
                             ; Registro Y=0x8100, dirección display numero
de pulsos.
               LDI R28, 0X00
               LDI TEMP, 0x80
                              ; Habilitar memoria externa en pines PortA
               STS XMCRA, TEMP
               LDI TEMP, 1<<XMM2 | 1<<XMM1 | 0<<XMM0
               STS XMCRB, TEMP
               ST X, TEMP0
                            ; Dirección display segundos 0x8000,
dirección display pulsos 0x8100
               ST Y, TEMP0
                              ; Definir segundos y pulsos inicial en 0, en
7segmentos y guardarlos en memoria externa, conectado a display
               LDI TEMP, 0XFF
               OUT PORTD, TEMP
; CONFIGURACIÓN DE PERIFÉRICOS: TIMER 1
LDI TEMP, 0xC2
                                         ; Iniciar Temporizador en C2F7,
para desbordar 1s
               LDI TEMP0, 0xF7
               STS TCNT1H, TEMP
               STS TCNT1L, TEMP0
               LDI TEMP, 1<<CS12 0<<CS11 1<<CS10; Divisor de frecuencia de 1024
               STS TCCR1B, TEMP
; HABILITAR INTERRUPCIONES INTO, TIMER 1 Y GLOBALES
LDI TEMP, 1<<TOIE1
                                   ; Habilitar interrupción timmer, con
desbordamiento 1s.
               STS TIMSK1, TEMP
               LDI TEMP, 1<<INT0
                                   ; Habilitar interrupcion externa
               OUT EIMSK, TEMP
               LDI TEMP, 0x02
                                         ; Interrupción por flanco de
bajada
               STS EICRA, TEMP
```

```
; BUCLE INFINITO: LOOP
BUCLE: NOP
                          JMP BUCLE
; Subrutina de Atención a la Interrupción del Timmer 1
INT_TEMP:
                     LDI TEMP, 0xC2
                                           ; Volver a iniciar
Temporizador en C2F7, para desbordar 1s
                          LDI TEMP0, 0xF7
                          STS TCNT1H, TEMP
                          STS TCNT1L, TEMP0
                          MOV TEMP, S
                          CALL AUMENTAR
                          MOV S, TEMP
                          CALL HEX_7SEG
                          ST X, TEMP0
                                            ; Display cambia
cada segundo.
                          RETI
[-----
; Subrutina de Atención a la Interrupción Externa 0
[-----
        INT EXTERNA: MOV TEMP, P
                          CALL AUMENTAR
                          MOV P, TEMP
                          CALL HEX 7SEG
                          ST Y, TEMP0
                                            ; Display cambia
cada pulso.
                          RETI
; Subrutinas aumentar y transformar a Código 7 segmentos
[------
        AUMENTAR:
                          ADD TEMP, K
                          CPI TEMP, 0x0A
                          BRLO FIN_AUM
                          LDI TEMP, 0x00
        FIN_AUM:
                     RET
        HEX_7SEG:
                          CERO:
                                   CPI TEMP,0x00
                                       BRNE UNO
                                       LDI TEMP0, 0x3F
                                       JMP FIN HEX 7SEG
                                   CPI TEMP,0x01
                          UNO:
                                       BRNE DOS
                                       LDI TEMP0, 0x06
                                       JMP FIN_HEX_7SEG
                          DOS:
                                   CPI TEMP,0x02
                                       BRNE TRES
                                       LDI TEMP0, 0x5B
                                       JMP FIN_HEX_7SEG
```

TRES: CPI TEMP, 0x03

BRNE CUATRO
LDI TEMPO, 0x4F
JMP FIN_HEX_7SEG

CUATRO: CPI TEMP,0x04

BRNE CINCO LDI TEMP0, 0x66 JMP FIN_HEX_7SEG

CINCO: CPI TEMP,0x05

BRNE SEIS

LDI TEMP0, 0x6D

JMP FIN_HEX_7SEG

SEIS: CPI TEMP,0x06

BRNE SIETE

LDI TEMP0, 0x7D

JMP FIN_HEX_7SEG

SIETE: CPI TEMP,0x07

BRNE OCHO

LDI TEMP0, 0x07

JMP FIN_HEX_7SEG

OCHO: CPI TEMP,0x08

BRNE NUEVE LDI TEMP0, 0x7F JMP FIN HEX 7SEG

NUEVE: LDI TEMP0, 0x6F

JMP FIN HEX 7SEG

FIN HEX 7SEG: RET

;-----; FIN DEL PROGRAMA

Conclusiones

- El uso de interrupciones es muy importante ya que permite al microcontrolador realizar otro tipo de actividades de fondo teniendo como prioridad el llamado de la interrupción.
- El realizar las bases de tiempo requeridas a través de las interrupciones del desbordamiento del temporizador, permite que el microcontrolador puede realizar otras tareas, sin tener que quedarse en ciclos donde se perdería su capacidad de procesamiento.
- El uso de algunos de los pines de los puertos paralelos como direcciones de memoria externa brinda una mejora al microcontrolador ya que le otorga más espacio de memoria para el desarrollo de sus aplicaciones.