

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 dígito 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.

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.

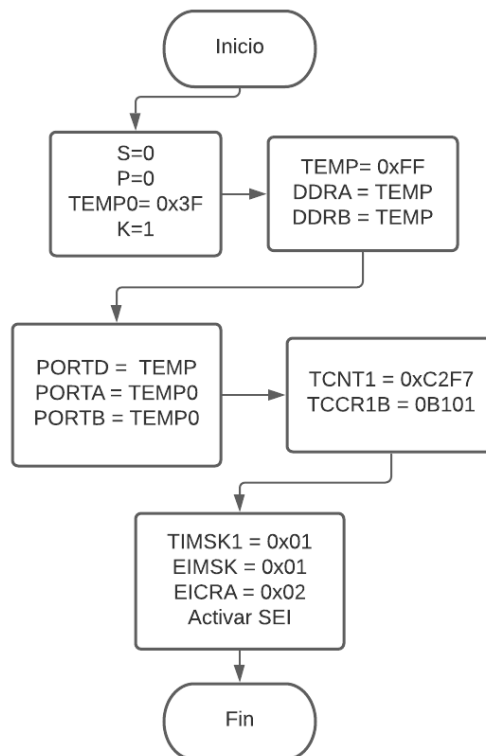


Figura 2. Diagrama de flujo subrutina INICIO (versión 1).

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.

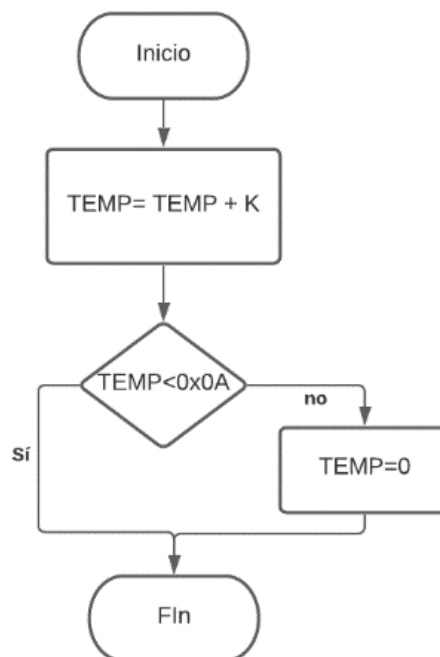


Figura 3. Diagrama de flujo subrutina AUMENTAR (versión 1 y 2).

Para la segunda version de solucion se utilizo el diagrama 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.

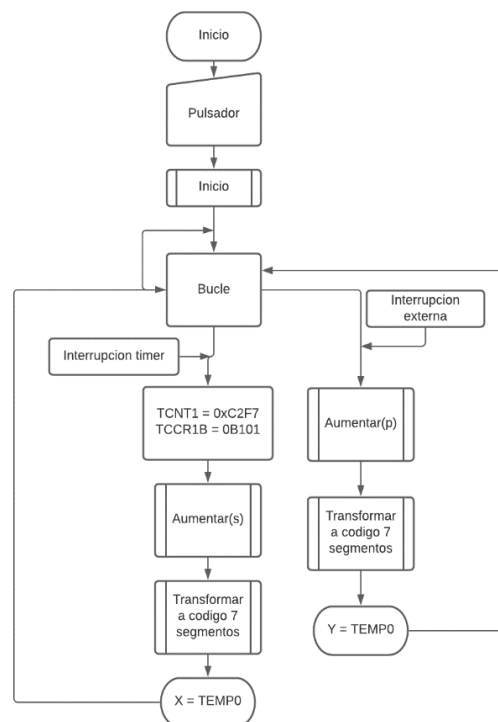


Figura 6. Diagrama de flujo principal (versión 2).

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 timer.

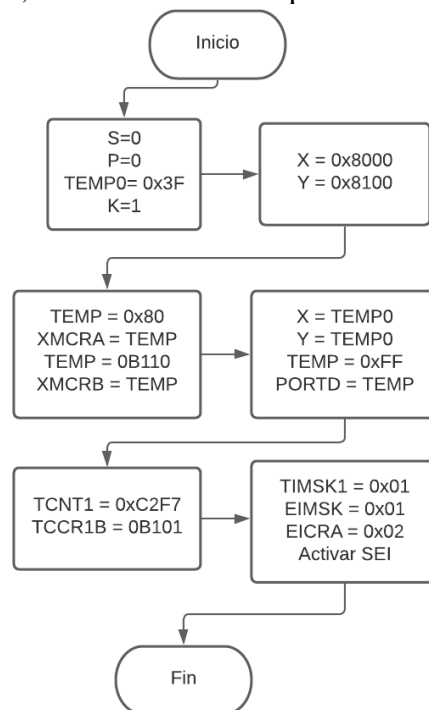


Figura 7. Diagrama de flujo subrutina INICIO (versión 2).

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.

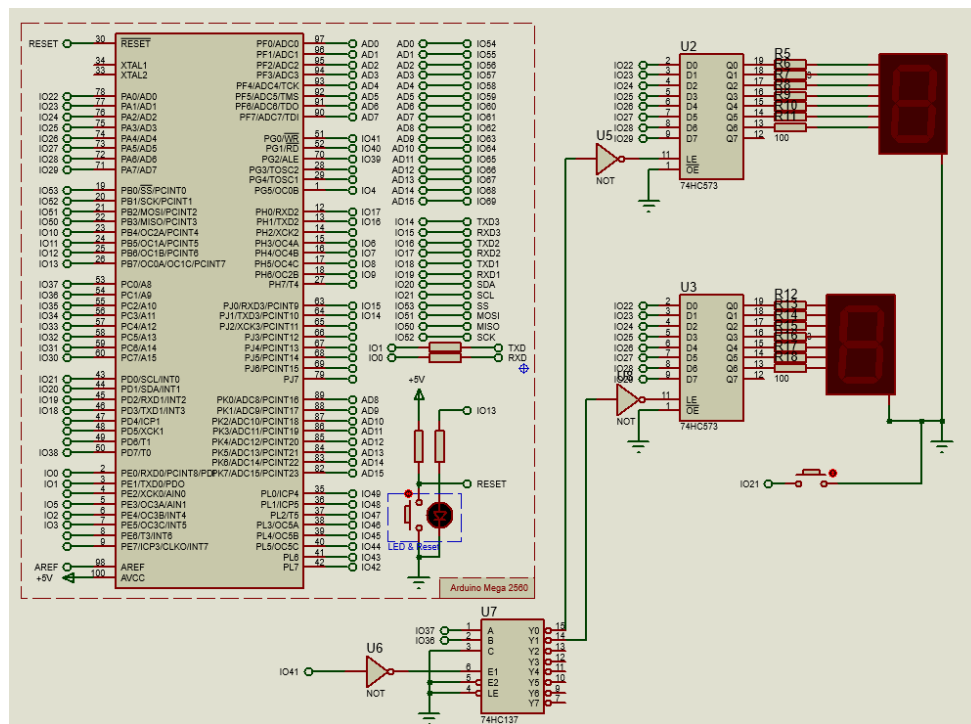


Figura 5. Circuito en Proteus (versión 2).

Resultados de simulación y análisis de estos.

Se probó 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.

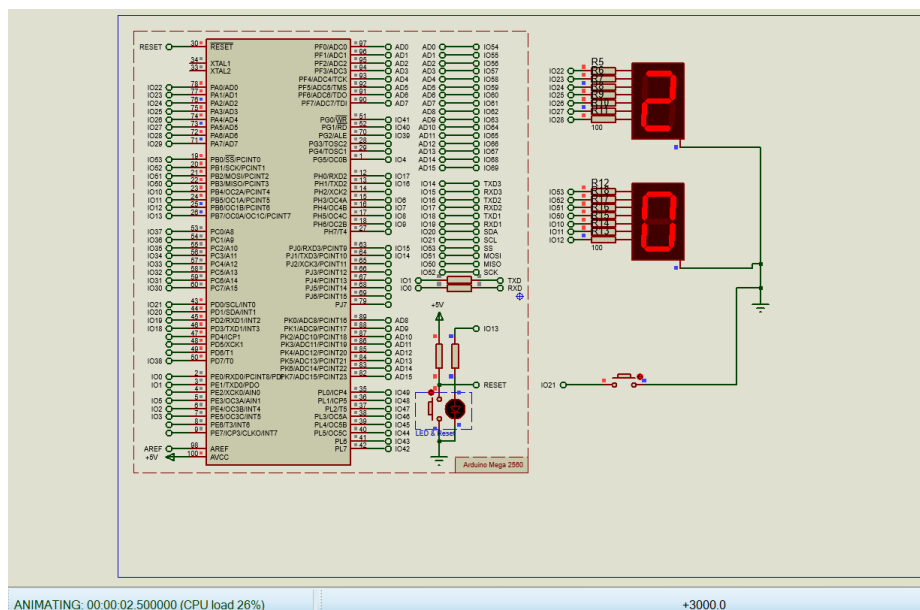


Figura 6. Circuito transcurridos 2 segundos de simulación (versión 1).

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.

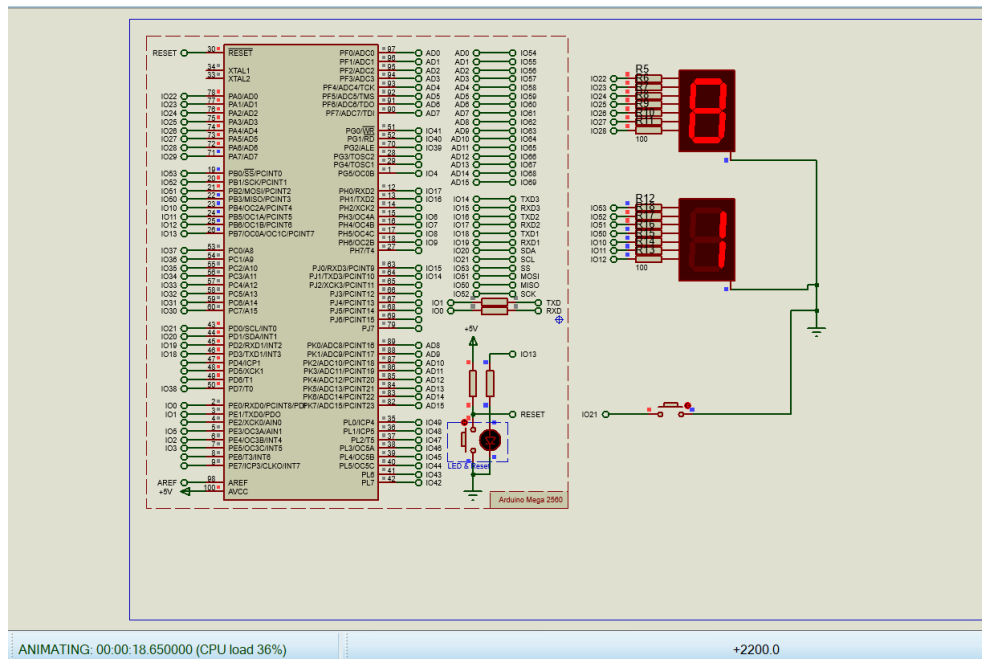


Figura 7. Circuito transcurridos 18 segundos de simulación, después de una pulsación (versión 1).

De igual forma, se probó 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.

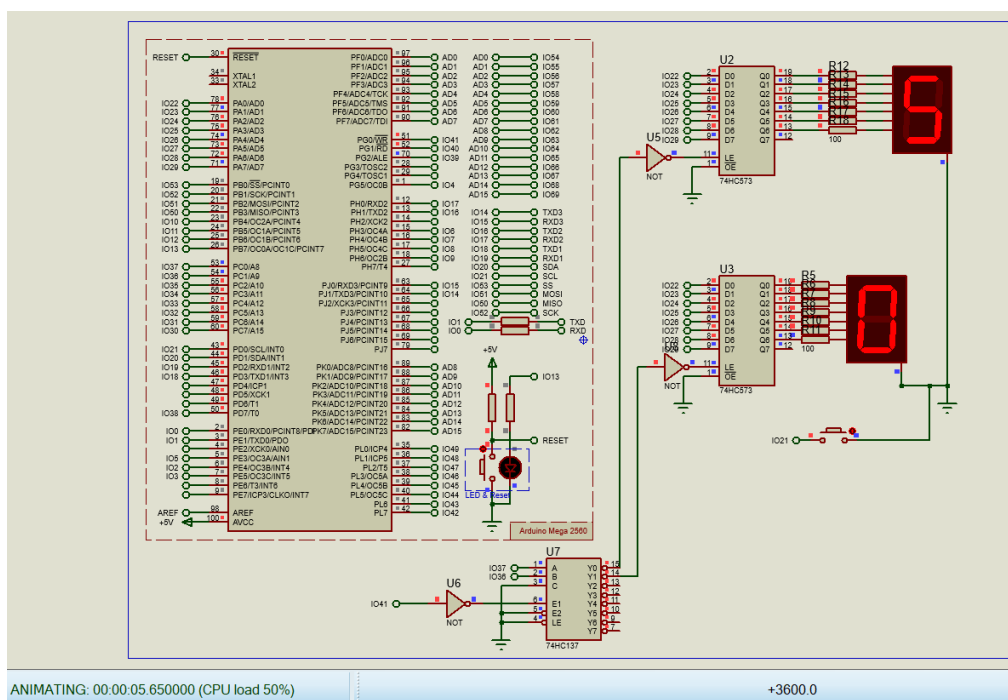


Figura 8. Circuito transcurridos 5 segundos de simulación (versión 2).

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.

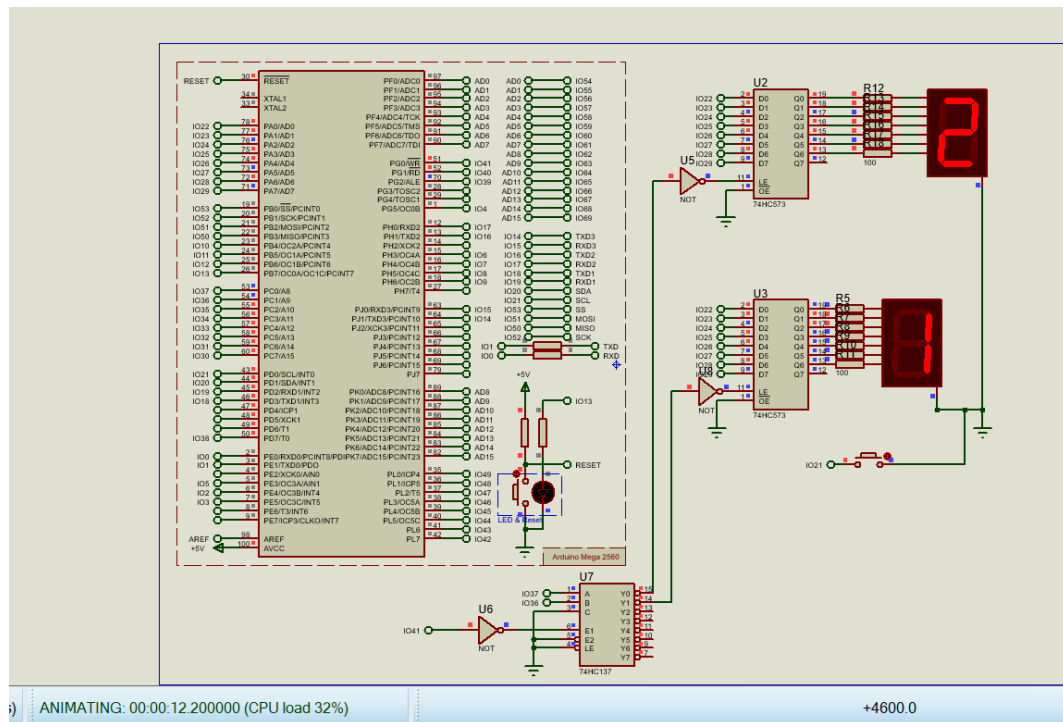


Figura 9. Circuito transcurridos 12 segundos de simulación, después de una pulsación (versión 2).

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.

```
;
; Arias_Gallego_Laboratorio1_v1.asm
;
; Autor: Jessica Arias
;

;=====
; DEFINICIONES INICIALES
;=====

.DEF TEMP=R16
.DEF TEMP0=R17
.DEF S=R18                ; variable segundos pasados en hex.
                           ; Se trabaja display 7 segmentos cátodo común
                           ; Crear constante 1

.DEF K=R19

.DEF P=R20                ; Variable número pulsos
```

```

;=====
; 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 inicio 0.
    LDI P, 0x00        ; Pulsos inicio 0.
    LDI TEMP0, 0x3F    ; 0 en 7segmentos cátodo común
    LDI K, 0x01        ; Constante K=1

    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

    LDI TEMP, 0B101    ; Divisor de frecuencia de 1024
    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<<INT0   ; Habilitar interrupción externa
    OUT EIMSK, TEMP

    LDI TEMP, 0x02      ; Interrupción por flanco de bajada
    STS EICRA, TEMP

    SEI

;=====
; 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

DOS:          CPI TEMP, 0x02
              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

```

```

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
;=====

```

Por último, se presenta el código de la versión 2 de solución.

```

;
; Arias_Gallego_Laboratorio1_v2.asm
;
; Autor: Jessi
;

```

```

;
; Arias_Gallego_Laboratorio1.asm
;
; Autor: Jessi
;

```

```

;=====
; DEFINICIONES INICIALES
;=====

```

```

    .DEF TEMP=R16
    .DEF TEMP0=R17
    .DEF S=R18                                ; variable segundos pasados en
hex.                                           ; Se trabaja display 7

segmentos cátodo común
    .DEF K=R19                                ; Crear constante 1

    .DEF P=R20                                ; Variable número pulsos

```

```

;=====
; VECTORES DE INTERRUPCION
;=====

```

```

    .ORG 0x0000
    JMP INICIO

    .ORG 0x0002

```

```

        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 TEMP0, 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 INTERRUPTACIONES INT0, 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

                SEI

;=====

```

```

; 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

UNO:         CPI TEMP, 0x01
              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 TEMP0, 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.