

**INSTITUTO POLITECNICO NACIONAL**  
**"ESCOM"**



**INGENIERIA EN SISTEMAS COMPUTACIONALES**

**REPORTE: CONTADOR UTILIZANDO INTERRUPTORES**

**ESQUIVEL PÉREZ JONATHAN ALFREDO**

**HERNÁNDEZ LÓPEZ ÁNGEL ZAIT**

**SALGADO GALLEGOS JESUS**

**SANCHEZ PIZANO IRVING DANIEL**

**INTRODUCCION A LOS MICROCONTROLADORES**

**PEREZ PEREZ JOSE JUAN**

**3CM8**

**09/JUNIO/2020**

## OBJETIVO. –

El objetivo de esta práctica es poder desarrollar lo aprendido sobre el manejo de interrupciones y también ver físicamente como funcionan las interrupciones en el microcontrolador.

## INTRODUCCIÓN. –

Las interrupciones es una de las características de los microcontroladores, de las más importantes que constituye la capacidad de sincronizar la ejecución de programas con acontecimientos externos; es decir, cuando se produce una interrupción, el micro automáticamente deja lo que esto haciendo, va a la dirección 04h de programa y ejecuta lo que encuentre a partir de allí hasta encontrarse con la instrucción RETFIE que le hará abandonar la interrupción y volver al lugar donde se encontraba antes de producirse dicha interrupción



Existen 2 tipos de interrupciones:

1. - Mediante una acción interna. El desbordamiento de la Pila (Stack) por una operación indebida, por ejemplo:

- Al completarse la escritura de datos en una EEPROM.
- Por desbordamiento del registro TMR0 al rebasar el valor 255 (FFh) a 0.

2. - Mediante una acción externa, la más útil. Al producirse un cambio del nivel en uno de sus pines por una acción externa.

- Estando en el modo de reposo (SLEEP), un cambio de nivel en el pin RB0/INT.
- Un cambio de nivel en uno de los pines RB4 a RB7 estando configurados como entrada.

## MATERIAL Y EQUIPO UTILIZADO

- Protoboard
- push button
- resistencias
- display 7 segmentos
- microcontrolador ATmega 8535
- software proteus

## QUE SE REALIZARA EN LA PRACTICA

Se realizará un contador el cual dependerá de 3 funciones, las cuales se realizarán mediante las interrupciones int0, int1 e int2. El resultado del contador debe ser mostrado en el display de 7 segmentos.

Las funciones que se mostraran son las siguientes:

- Init0: Debe mostrar en el display del 0 a 9.
- Init1: Cuando el interruptor init1 es activado, se tendrá que mostrar en el display del 9 al 0.
- Init2: Cuando se activa init2, se congela el conteo en donde se haya quedado.

## DESARROLLO DE LA PRACTICA INCLUYENDO DIAGRAMAS DE FLUJO, CODIFICACION COMENTADA

CODIFICACION:

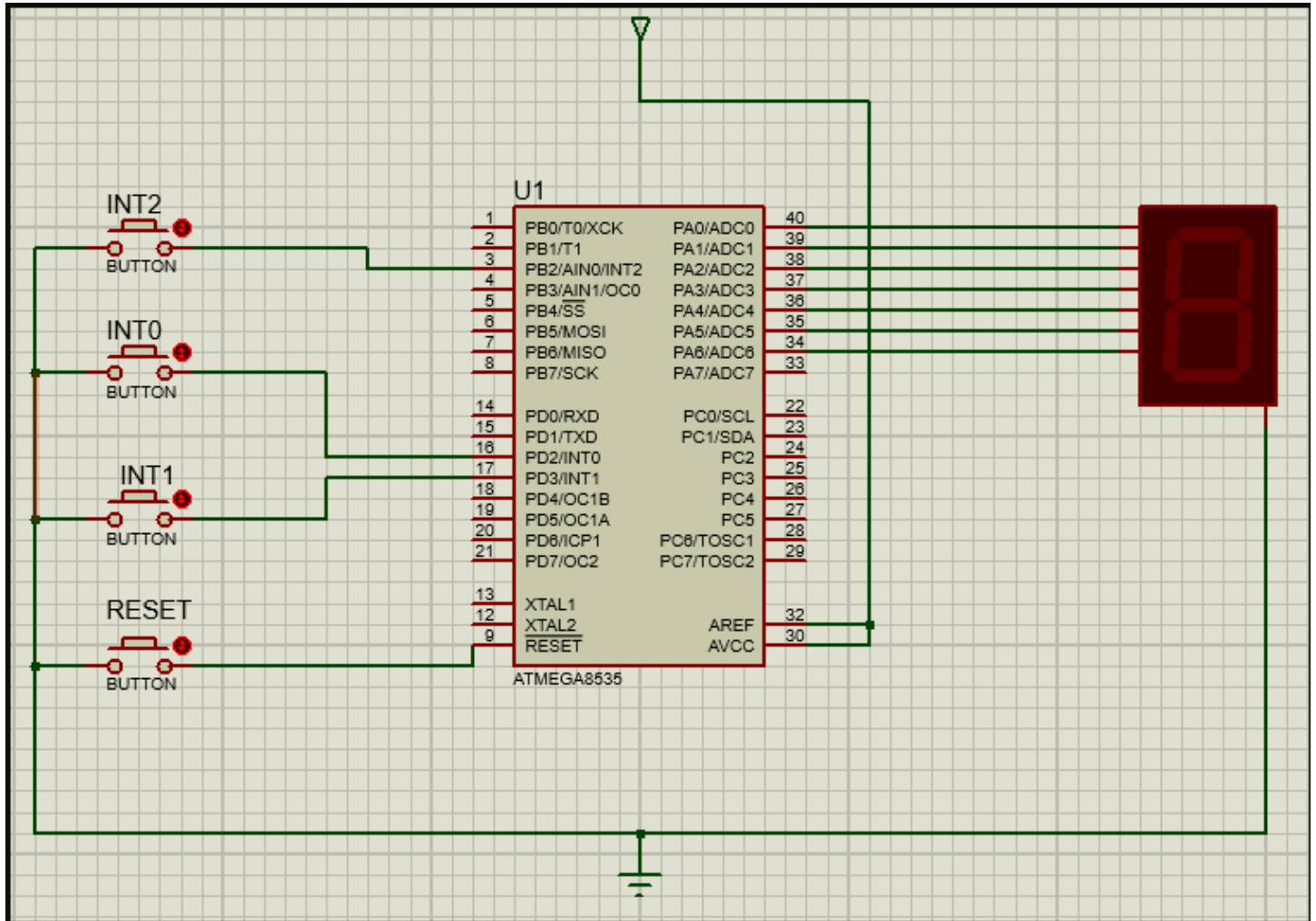
```
1. .include "m8535def.inc"
2. .def aux=r16
3. .def cont=r17
4. .def aux2=r18
5.
6. rjmp main          ; el programa brinca a esta instruccion cuando ocurre reset
7. rjmp incre          ; Registro $001 (INT0) tiene rutina 'incre'
8. rjmp decre          ; Registro $002 (INT1) tiene rutina 'decre'
9. .org $009           ; Salto al registro $009
10. rjmp sube          ; Se agrega la rutina para cada pulso
11. .org $012           ; Salto al registro $012
12. rjmp pause         ; Se agrega la rutina 'pause'
13.
14. main:
15.     ldi aux, low(RAMEND)
16.     out spl, aux
17.     ldi aux, high(RAMEND)
18.     out sph, aux    ; Estas 4 instr. inicializan el SP
19.     ser aux
20.     out ddra, aux   ; Activamos los puetos 'A' para la salida
21.     ; out ddrc, aux
22.     out portd, aux  ; Activamos los puertos 'D' como entrada
23.     out portb, aux  ; Activamos los puertos 'B' como entrada
24.     ldi aux,$0E
25.     out MCUCR,aux   ; Se establece INT0 para flanco de bajada e INT1 para flanco de subida
26.     ldi aux,$40
27.     out MCUCSR,aux  ; Se establece INT2
28.     ldi aux,$E0
29.     out GICR,aux    ; Se habilitan INT0 INT1 INT2
30.     ldi aux, 5
31.     out tccr0, aux
32.     ldi aux, 1
33.     out tmsk, aux
34.     sei
35.
36. load:                ; Valores del 0 al 9 en BCD
37.     ldi R20, $3F
38.     ldi R21, $06
39.     ldi R22, $5B
40.     ldi R23, $4F
41.     ldi R24, $66
42.     ldi R25, $6D
43.     ldi R26, $7D
44.     ldi R27, $27
```

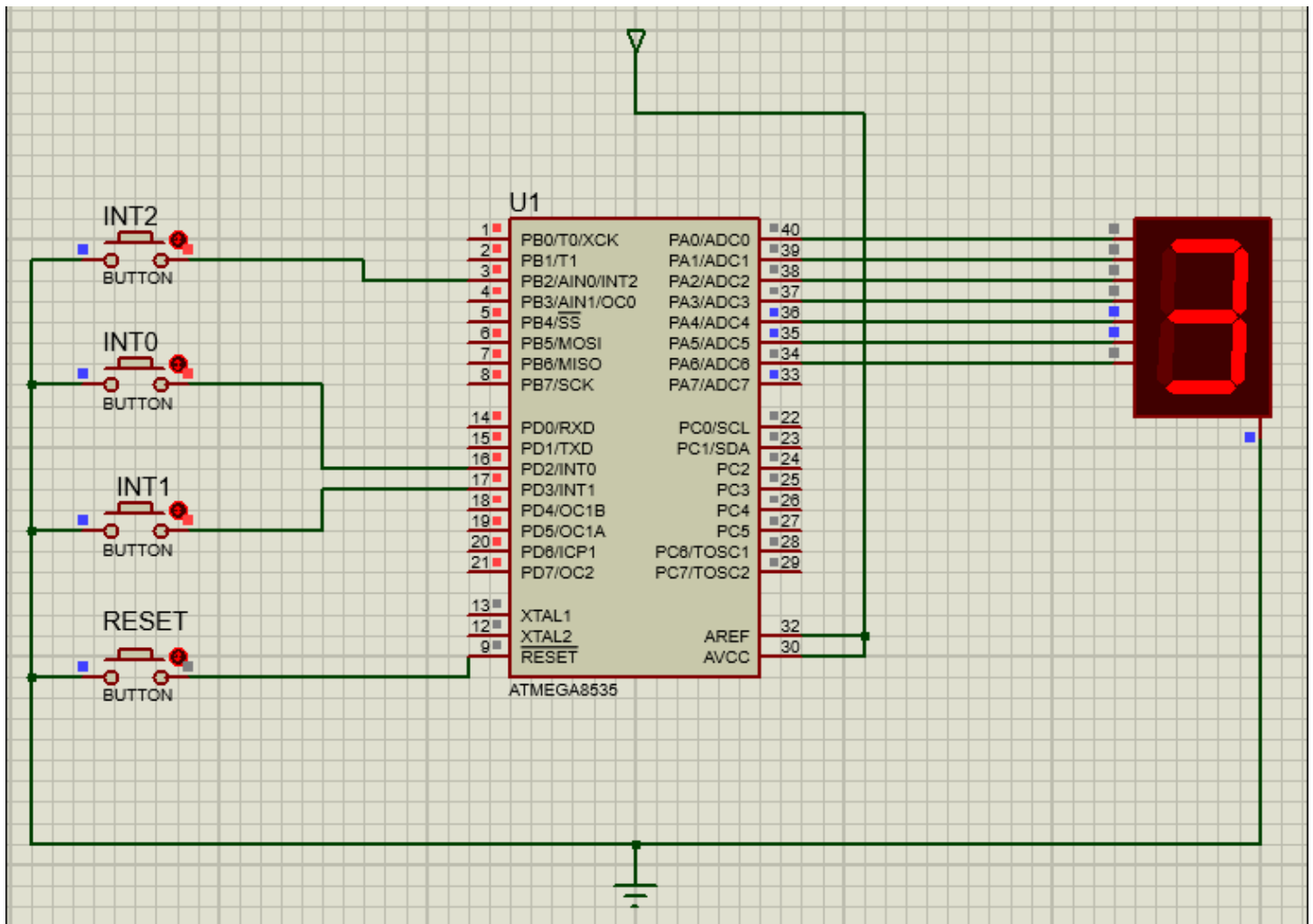
```

45.    ldi R28, $7F
46.    ldi R29, $6F
47.    ldi aux2, $01    ; Inicializa en 1 aux2
48.    clr ZH
49.    clr cont          ; Contador en cero
50.
51. loop:
52.    rcall BCD
53.    out PORTA, aux      ; Se muestra el valor del contador en BCD
54.    ; out PORTC, cont
55.    rjmp loop
56.
57. sube:                ; Se le suma al contador aux2 para saber si
58.    add cont,aux2      ; avanza, retrocede o se queda ahí
59.    cpi cont, 10
60.    breq a_dece00      ; Si el contador es igual a 10, salta a a_dec00
61.    cpi cont, $ff
62.    breq a_dece01      ; Si el contador es igual a -1, salta a a_dec01
63.    reti
64.
65. a_dece00:
66.    clr cont          ; Pone en cero el contador
67.    reti
68.
69. a_dece01:
70.    ldi cont,$09      ; Pone en 9 el contador
71.    reti
72.
73. incre:              ; Si INT0 es activado
74.    ldi aux2, $01      ; aux2 toma el valor de 1
75.    reti
76.
77. decre:              ; Si INT1 es activado
78.    ldi aux2, $ff      ; aux2 toma el valor de -1
79.    reti
80.
81. pause:              ; Si INT2 es activado
82.    ldi aux2, $00      ; aux2 toma el valor de 0
83.    reti
84.
85. BCD:                ; Pone el contador de binario a 7 segmentos
86.    ldi ZL, 20
87.    add ZL, cont
88.    ld aux, Z
89.    ret

```

## SIMULACION:





## CONCLUSIONES Y OBSERVACIONES INDIVIDUALES

### ❖ Esquivel Pérez Jonathan Alfredo

En la presente práctica se desarrolló un contador en cual por medio del uso de macros repite una secuencia determinada al presionar los botones en el circuito. Los valores de la secuencia fueron almacenados en registros los cuales son llamados dependiendo de lo que se quiera mostrar. Para mostrar el circuito se hizo uso de proteus el cual también se utilizó para simular el código de la presente práctica

### ❖ Hernández López Ángel Zait

En esta práctica se pudo ver el manejo de interrupciones, y que es necesario saber en qué registro está cada uno de ellos para poder ser accionado correctamente.

Además de que se pueden hacer operaciones con valores negativos, ya que no todos los números están en el rango de los reales positivos, ya que se respeta la teoría de  $[2^{n-1}, 2^n]$  para el uso de valores numéricos. Se utilizó el timer disponible en el microcontrolador para poder hacer mejor el conteo, y fuera sumando el valor correspondiente en cada pulso. Aquí hacemos un estilo de desbordamiento personalizado para poder cambiar el valor del contador, es decir, para saber si va aumentando o disminuyendo el conteo, este tiene activados un registro especial que va cambiando, dependiendo el registro presionado. Pero a su vez, se cambia la condición de inicio, si es que el desbordamiento es igual a -1, este reinicia el contador en 9 y si es 10 reinicia el contador en 0.

❖ Salgado Gallegos Jesús

Con las interrupciones en el microcontrolador mediante eventos de flancos de subida o de baja, el programa principal se detiene y comienza a realizar otra tarea, hasta que el microcontrolador recibe otra interrupción y cambie de actividad o se pida un reset y el programa vuelva a su estado original. Para poder habilitar estas interrupciones se hace uso de bits que detectan una interrupción externa y se habilitan o no, dependiendo del flanco que se haya detectado. Cada interrupción como se ve en el programa corresponde a un registro diferente por el cual se debe de activar una instrucción que guarde ese vector en el registro adecuado.

❖ Sanchez Pizano Irving Daniel

En esta practica pude entender como funcionan las interrupciones en el microcontrolador. Las cuales pude observar en esta practica que tenemos un contador cuyo resultado se muestra en el display y que, al presionar el botón de las interrupciones, el programa hace otro funcionamiento el cual es dependiendo de las interrupciones que se señaló. Por ejemplo, puede ir ascendente o descendentemente. Aprendí que el microcontrolador cuando recibe la señal de interrupción este deja de hacer su funcionamiento y posteriormente hace el funcionamiento de la interrupción señalada y básicamente así funcionan las interrupciones.