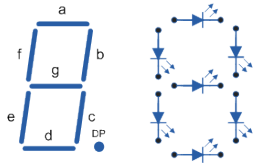


Display Decoder

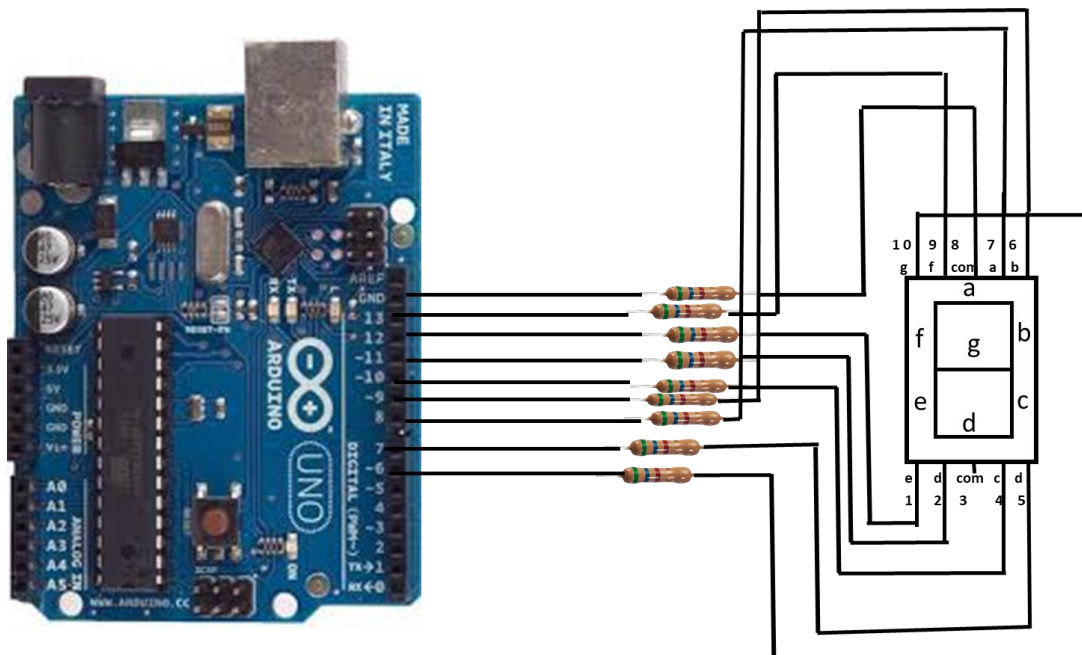


A **Digital Decoder** IC, is a device which converts one digital format into another and one of the most commonly used devices for doing this is called the Binary Coded Decimal (BCD) to 7-Segment Display Decoder.

7-segment **LED** (Light Emitting Diode) or **LCD** (Liquid Crystal Display) type displays, provide a very convenient way of displaying information or digital data in the form of numbers, letters or even alpha-numerical characters.

Hardware Required

- Arduino or Genuino Board
- Display Decoder
- pin headers to solder to the LCD display pins
- Eight (8) -220 ohm resistor
- hook-up wires
- breadboard



Code in Assembly:

```
;
; Counter.asm
;
; Created: 2/6/2019 5:49:33 PM
; Author : hmoykwan
;

;.include "m328pdef.inc" ;Assembler library / Libreria en assembler para el micro ATmega8

.org $0000 ;PLACE THE PROGRAM MEMORY ADDRESS COUNTER IN 0000/ COLOCA EL CONTADOR DE
DIRECCION DE MEMORIA DE PROGRAMA EN 0000

.def var = r16 ; DEFINE var EQUAL A r16/ DEFINE var IGUAL A r16
.def contador = r21 ;define counter equal to r24 / define contador igual a r24
.def val = r22 ;define the val comparison value / define el valor de comparacion val
rjmp INIT ; Jump TO INIT LABEL/ SALTA A ETIQUETA INIT

INIT:
; ROUTINE to CONFIGURING B AND D PORTS WITH ENTRY AND EXIT BIT / RUTINA DE INICIO QUE
CONFIGURA PUERTOS B Y D CON BIT DE ENTRADA Y SALIDA

    ldi contador,0x00 ; load the counter =0 /carga el contador con 0
    ldi val,0x01 ; load the counter=1 / carga el contador con 1
    ldi var,0xf0 ;salida nivel alto - entradas nivel bajo--- CARGA EL r16 CON HEXADECIMAL
    APROPIADO
                                ; 0xf0 INDICA LOS 4 BIT MAS ALTOS COMO SALIDA Y LOS 4 MAS BAJOS COMO
ENTRADA
    out DDRD,var ; PASA LOS BIT CARGADOS DE CONFIGURACION EN var (r16) AL REGISTRO DE
CONFIGURACION DE PUERTO B
                                ;CONFIGURA PINES DEL PUERTO D LOS MAS ALTOS COMO SALIDA Y LOS
BAJOS COMO ENTRADA
                                ;PERMITE USAR PD0 PD1 PD2 y PD3 COMO ENTRADAS PD4 COMO SALIDA
DE LA OR

    cbi PORTD,4 ; COLOCA EL BIT 4 DEL PUERTO D A 0 PARA QUE LA SALIDA INICIE EN 0

    ldi var,0xFF ;entradas nivel alto - salidas CARGA EL r16 CON EXADECIMAL APROPIADO
                                ; 0xff INDICA LOS 8 BIT COMO SALIDA
                                ;
    out DDRB,var ; PASA LOS BIT CARGADOS DE CONFIGURACION EN var (r16) AL REGISTRO DE
CONFIGURACION DE PUERTO B
                                ;CONFIGURA PINES DEL PUERTO B COMO SALIDA

; dibujamos el 0 en el display de 7 segmentos para arranque de 0
    sbi PORTB,0 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 0 COLOCANDO
EL SEGMENTO a en 1
    sbi PORTB,1 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 0 COLOCANDO
EL SEGMENTO b en 1
    sbi PORTB,2 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 0 COLOCANDO
EL SEGMENTO c en 1
    sbi PORTB,3 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 0 COLOCANDO
EL SEGMENTO d en 1
```

```

sbi PORTB,4 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 0 COLOCANDO
EL SEGMENTO e en 1
sbi PORTB,5 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 0 COLOCANDO
EL SEGMENTO f en 1
cbi PORTB,6 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 0 COLOCANDO
EL SEGMENTO g en 0
cbi PORTB,7 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 0 COLOCANDO
EL SEGMENTO . en 0

```

start:

```

rcall DELAY ; Llama a la rutina de Retardo de tiempo Delay (evita revotes de
conteo y permite ver numeros 1/2 segundo aprox)
sbic PIND,0 ;prueba si esta presionada la entrada de no ser así realiza la
sentencia siguiente
inc contador ; incrementa el contador

```

```

val_1:
ldi val,0x01 ; carga el contador con 1
CPSE contador,val ; compara contador y val y salta siguiente instruccion si son
iguales
rjmp val_2 ;SALTA A val_2
rjmp esc_1 ; ir a rutina para dibujar un 1

```

```

val_2:
ldi val,0x02 ; carga el contador con 2
CPSE contador,val ; compara contador y val y salta siguiente instruccion si son
iguales
rjmp val_3 ;SALTA A val_3
rjmp esc_2 ; ir a rutina para dibujar un 2

```

```

val_3:
ldi val,0x03 ; carga el contador con 3
CPSE contador,val ; compara contador y val y salta siguiente instruccion si son
iguales
rjmp val_4 ;SALTA A val_4
rjmp esc_3 ; ir a rutina para dibujar un 3

```

```

val_4:
ldi val,0x04 ; carga el contador con 4
CPSE contador,val ; compara contador y val y salta siguiente instruccion si son
iguales
rjmp val_5 ;SALTA A val_5
rjmp esc_4 ; ir a rutina para dibujar un 4

```

```

val_5:
ldi val,0x05 ; carga el contador con 5
CPSE contador,val ; compara contador y val y salta siguiente instruccion si son
iguales
rjmp val_6 ;SALTA A val_6
rjmp esc_5 ; ir a rutina para dibujar un 5

```

```

val_6:
ldi val,0x06 ; carga el contador con 6

```

```

    CPSE contador,val      ; compara contador y val y salta siguiente instruccion si son
iguales
    rjmp val_7             ;SALTA A val_7
    rjmp esc_6             ; ir a rutina para dibujar un 6

val_7:
    ldi val,0x07           ; carga el contador con 7
    CPSE contador,val      ; compara contador y val y salta siguiente instruccion si son
iguales
    rjmp val_8             ;SALTA A val_8
    rjmp esc_7             ; ir a rutina para dibujar un 7

val_8:
    ldi val,0x08           ; carga el contador con 8
    CPSE contador,val      ; compara contador y val y salta siguiente instruccion si son
iguales
    rjmp val_9             ;SALTA A val_9
    rjmp esc_8             ; ir a rutina para dibujar un 8

val_9:
    ldi val,0x09           ; carga el contador con 9
    CPSE contador,val      ; compara contador y val y salta siguiente instruccion si son
iguales
    rjmp val_0             ;SALTA A val_9
    rjmp esc_9             ; ir a rutina para dibujar un 9

    rjmp start             ;SALTA A START PARA REPETIR

val_0:
    ;reinicio de valores a 0
    ldi val,0x00
    ldi contador,0x00
    rjmp esc_0             ; ir a rutina para dibujar un 0

    rjmp start             ;SALTA A START PARA REPETIR

```

```

esc_0:
    sbi PORTB,0 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 0 COLOCANDO
EL SEGMENTO a en 1
    sbi PORTB,1 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 0 COLOCANDO
EL SEGMENTO b en 1
    sbi PORTB,2 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 0 COLOCANDO
EL SEGMENTO c en 1
    sbi PORTB,3 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 0 COLOCANDO
EL SEGMENTO d en 1
    sbi PORTB,4 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 0 COLOCANDO
EL SEGMENTO e en 1
    sbi PORTB,5 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 0 COLOCANDO
EL SEGMENTO f en 1
    cbi PORTB,6 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 0 COLOCANDO
EL SEGMENTO g en 1
    cbi PORTB,7 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 0 COLOCANDO
EL SEGMENTO . en 0
    rjmp start             ;SALTA A START PARA REPETIR

```

```

esc_1:
    cbi PORTB,0 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 1 COLOCANDO
EL SEGMENTO a en 0
    sbi PORTB,1 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 1 COLOCANDO
EL SEGMENTO b en 1
    sbi PORTB,2 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 1 COLOCANDO
EL SEGMENTO c en 1
    cbi PORTB,3 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 1 COLOCANDO
EL SEGMENTO d en 1
    cbi PORTB,4 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 1 COLOCANDO
EL SEGMENTO e en 1
    cbi PORTB,5 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 1 COLOCANDO
EL SEGMENTO f en 1
    cbi PORTB,6 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 1 COLOCANDO
EL SEGMENTO g en 1
    cbi PORTB,7 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 1 COLOCANDO
EL SEGMENTO . en 0
    rjmp start ;SALTA A START PARA REPETIR

```

```

esc_2:
    sbi PORTB,0 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 2 COLOCANDO
EL SEGMENTO a en 1
    sbi PORTB,1 ; COLOCA EL BIT 1 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 2 COLOCANDO
EL SEGMENTO b en 1
    cbi PORTB,2 ; COLOCA EL BIT 2 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 2 COLOCANDO
EL SEGMENTO c en 0
    sbi PORTB,3 ; COLOCA EL BIT 3 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 2 COLOCANDO
EL SEGMENTO d en 1
    sbi PORTB,4 ; COLOCA EL BIT 4 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 2 COLOCANDO
EL SEGMENTO e en 1
    cbi PORTB,5 ; COLOCA EL BIT 5 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 2 COLOCANDO
EL SEGMENTO f en 0
    sbi PORTB,6 ; COLOCA EL BIT 6 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 2 COLOCANDO
EL SEGMENTO g en 0
    cbi PORTB,7 ; COLOCA EL BIT 7 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 2 COLOCANDO
EL SEGMENTO . en 0
    rjmp start ;SALTA A START PARA REPETIR

```

```

esc_3:
    sbi PORTB,0 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 3 COLOCANDO
EL SEGMENTO a en 1
    sbi PORTB,1 ; COLOCA EL BIT 1 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 3 COLOCANDO
EL SEGMENTO b en 1
    sbi PORTB,2 ; COLOCA EL BIT 2 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 3 COLOCANDO
EL SEGMENTO c en 1
    sbi PORTB,3 ; COLOCA EL BIT 3 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 3 COLOCANDO
EL SEGMENTO d en 1
    cbi PORTB,4 ; COLOCA EL BIT 4 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 3 COLOCANDO
EL SEGMENTO e en 0
    cbi PORTB,5 ; COLOCA EL BIT 5 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 3 COLOCANDO
EL SEGMENTO f en 0
    sbi PORTB,6 ; COLOCA EL BIT 6 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 3 COLOCANDO
EL SEGMENTO g en 1
    cbi PORTB,7 ; COLOCA EL BIT 7 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 3 COLOCANDO
EL SEGMENTO . en 0
    rjmp start ;SALTA A START PARA REPETIR

```

```

esc_4:
    cbi PORTB,0 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 4 COLOCANDO
EL SEGMENTO a en 0
    sbi PORTB,1 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 4 COLOCANDO
EL SEGMENTO b en 1
    sbi PORTB,2 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 4 COLOCANDO
EL SEGMENTO c en 1
    cbi PORTB,3 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 4 COLOCANDO
EL SEGMENTO d en 0
    cbi PORTB,4 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 4 COLOCANDO
EL SEGMENTO e en 0
    sbi PORTB,5 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 4 COLOCANDO
EL SEGMENTO f en 1
    sbi PORTB,6 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 4 COLOCANDO
EL SEGMENTO g en 1
    cbi PORTB,7 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 4 COLOCANDO
EL SEGMENTO . en 0
    rjmp start ;SALTA A START PARA REPETIR

```

```

esc_5:
    sbi PORTB,0 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO a en 1
    cbi PORTB,1 ; COLOCA EL BIT 1 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO b en 0
    sbi PORTB,2 ; COLOCA EL BIT 2 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO c en 1
    sbi PORTB,3 ; COLOCA EL BIT 3 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO d en 1
    cbi PORTB,4 ; COLOCA EL BIT 4 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO e en 0
    sbi PORTB,5 ; COLOCA EL BIT 5 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO f en 1
    sbi PORTB,6 ; COLOCA EL BIT 6 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO g en 1
    cbi PORTB,7 ; COLOCA EL BIT 7 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO . en 0
    rjmp start ;SALTA A START PARA REPETIR

```

```

esc_6:
    sbi PORTB,0 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO a en 1
    cbi PORTB,1 ; COLOCA EL BIT 1 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO b en 0
    sbi PORTB,2 ; COLOCA EL BIT 2 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO c en 1
    sbi PORTB,3 ; COLOCA EL BIT 3 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO d en 1
    sbi PORTB,4 ; COLOCA EL BIT 4 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO e en 1
    sbi PORTB,5 ; COLOCA EL BIT 5 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO f en 1
    sbi PORTB,6 ; COLOCA EL BIT 6 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO g en 1
    cbi PORTB,7 ; COLOCA EL BIT 7 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO . en 0
    rjmp start ;SALTA A START PARA REPETIR

```

```

esc_7:

```

```

    sbi PORTB,0 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 1 COLOCANDO
EL SEGMENTO a en 1
    sbi PORTB,1 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 1 COLOCANDO
EL SEGMENTO b en 1
    sbi PORTB,2 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 1 COLOCANDO
EL SEGMENTO c en 1
    cbi PORTB,3 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 1 COLOCANDO
EL SEGMENTO d en 1
    cbi PORTB,4 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 1 COLOCANDO
EL SEGMENTO e en 1
    cbi PORTB,5 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 1 COLOCANDO
EL SEGMENTO f en 1
    cbi PORTB,6 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 1 COLOCANDO
EL SEGMENTO g en 1
    cbi PORTB,7 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 1 COLOCANDO
EL SEGMENTO . en 0
    rjmp start ;SALTA A START PARA REPETIR

```

```

esc_8:
    sbi PORTB,0 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO a en 1
    sbi PORTB,1 ; COLOCA EL BIT 1 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO b en 1
    sbi PORTB,2 ; COLOCA EL BIT 2 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO c en 1
    sbi PORTB,3 ; COLOCA EL BIT 3 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO d en 1
    sbi PORTB,4 ; COLOCA EL BIT 4 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO e en 1
    sbi PORTB,5 ; COLOCA EL BIT 5 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO f en 1
    sbi PORTB,6 ; COLOCA EL BIT 6 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO g en 1
    cbi PORTB,7 ; COLOCA EL BIT 7 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO . en 0
    rjmp start ;SALTA A START PARA REPETIR

```

```

esc_9:
    sbi PORTB,0 ; COLOCA EL BIT 0 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO a en 1
    sbi PORTB,1 ; COLOCA EL BIT 1 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO b en 1
    sbi PORTB,2 ; COLOCA EL BIT 2 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO c en 1
    sbi PORTB,3 ; COLOCA EL BIT 3 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO d en 1
    cbi PORTB,4 ; COLOCA EL BIT 4 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO e en 0
    sbi PORTB,5 ; COLOCA EL BIT 5 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO f en 1
    sbi PORTB,6 ; COLOCA EL BIT 6 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO g en 1
    cbi PORTB,7 ; COLOCA EL BIT 7 DEL PUERTO B A 1 PARA QUE LA SALIDA dibuje un 5 COLOCANDO
EL SEGMENTO . en 0
    rjmp start ;SALTA A START PARA REPETIR

```

DELAY: ;Rutina de Retardo Llamada DELAY= 1/2 segundo
 aproximado para evitar los rebotes del suiche de entrada

; Delay 16 000 000 cycles (esta rutina pone al micro en una tarea de cargar y restar
registros con valores adecuados)
; 1s at 16 MHz (se realizan 16 000 000 operaciones a 16 MHz por tanto tarda 1 segundo
en completar)

```
    ldi r18, 82 ;carga el literal o valor 41 en registro 18
    ldi r19, 43 ;carga el literal o valor 43 en registro 19
    ldi r20, 0  ;carga el literal o valor 0 en registro 20
L1: dec r20      ; decrementa en 1 el registro 20
    brne L1      ; si no es igual a 0 pasa a L1 de lo contrario continua en siguiente
instruccion
    dec r19      ; decrementa en 1 el registro 19
    brne L1      ; si no es igual a 0 pasa a L1 de lo contrario continua en
siguiente instruccion
    dec r18      ; decrementa en 1 el registro 18
    brne L1      ; si no es igual a 0 pasa a L1 de lo contrario
continua en siguiente instruccion
    lpm          ;
    nop          ; pierde 1 cycle en hacer nada

ret      ; retorna el llamado
```