

```
;VARIABLES GLOBALES
ESTADO EQU 0x20
EVENTO EQU 0x21
VALOR_DISPLAY EQU 0x22
T_CONT1 EQU 0x23
T_CONT2 EQU 0x24
TICK EQU 0x25.0
PARPADEO EQU 0x25.1
FADC EQU 0x25.2
CONT_ZUMB EQU 0x26
CONT_PARPADEO EQU 0x27

ADCON EQU 0xC5
ADCH EQU 0xFE
PWMP EQU 0xFE
PWM0 EQU 0xFC
IEN0 EQU 0xA8

;PUERTOS
LED EQU P2.0
ZUMBADOR EQU P2.1
PLUS EQU P1.0
MINUS EQU P1.1
SWITCH EQU P1.2
SENSOR EQU P1.5

;VARIABLES DE ESTADOS
;1
ES1_CONT_MS EQU 0x2A
ES1_CONT_S EQU 0x2B
;2
ES2_CONT_MS EQU 0x2C
ES2_CONT1_S EQU 0x2D
ES2_CONT2_S EQU 0x2E

;=====INICIO=====
ORG 0x0000
    AJMP INICIO

ORG 0x007b
INICIO:
    ACALL INIT

MAIN:
    ACALL FSM
    AJMP MAIN

FSM:
    MOV A, ESTADO
    RL A
    MOV DPTR, #EST_TAB
    JMP @A+DPTR

EST_TAB:
    AJMP ESTADO0          ;REPOSO
    AJMP ESTADO1          ;ESPERA
    AJMP ESTADO2          ;CALENTAR
    AJMP ESTADO3          ;TRANSICION

;=====INIT=====;
INIT:
    ;JUST DEBUGGING!!!!!!!
    ;MOV P0, #0
    MOV P1, #0
    ;MOV P2, #0
    ;ACALL ENCENDER_TIMER
```

```
;ACALL ENCENDER_ADC
```

```
;VALORES INICIALES
```

```
MOV ESTADO, #0
MOV EVENTO, #0
MOV VALOR_DISPLAY, #0
MOV T_CONT1, #0
MOV T_CONT2, #0
MOV CONT_ZUMB, #0
MOV CONT_PARPADEO, #0
CLR PARPADEO
```

```
;PUERTOS ENTRADA
```

```
SETB PLUS
SETB MINUS
SETB SWITCH
SETB SENSOR
```

```
;PUERTOS SALIDA
```

```
CLR P0.0
CLR P0.1
CLR P0.2
CLR P0.3
CLR P0.4
CLR P0.5
CLR P0.6
CLR P0.7
CLR LED
CLR ZUMBADOR
SETB ZUMBADOR
```

```
;TIMER
```

```
ORL TMOD, #00000010b
ANL TMOD, #11110010b
CLR TICK
```

```
;VARIABLES DE ESTADOS
```

```
;1
MOV ES1_CONT_MS, #0
MOV ES1_CONT_S, #0
;2
MOV ES2_CONT_MS, #0
MOV ES2_CONT1_S, #0
MOV ES2_CONT2_S, #0
```

```
RET
```

```
;=====ESTADO0_REPOSO=====;
```

```
ESTADO0:
```

```
ACALL ES0_GEN_EV
MOV A, EVENTO
RL A
MOV DPTR, #ES0_EV_TAB
JMP @A+DPTR
```

```
ES0_EV_TAB:
```

```
AJMP ES0_EV0 ;EVENTO VACIO
AJMP ES0_EV1 ;ENCENDIDO
```

```
ES0_EV0: ;EVENTO VACIO
```

```
RET
```

```
ES0_EV1: ;EVENTO ESPERA
```

```
;PONER UN 0 EN PANTALLA
MOV VALOR_DISPLAY, #0
ACALL UPDATE_DISPLAY
```

```

    ACALL ENCENDER_PARPADEO
    ;LED ON
    ACALL ENCENDER_LED
    ;ZUMBIDO ON - (SE APAGARA EN 200MS)
    ACALL ENCENDER_ZUMBADOR
    ;ENCENDER TIMER
    ACALL ENCENDER_TIMER
    ;CAMBIAR AL ESTADO 1
    MOV ESTADO, #1
    ;CAMBIAR EL EVENTO AL 0
    MOV EVENTO, #0
    RET
ES0_GEN_EV:
    ;COMPROBAR SI SE HA PULSADO EL BOTON
    MOV C, SWITCH
    JC ES0_SWITCH
    RET
ES0_SWITCH:
    MOV EVENTO, #1
    RET
;=====ESTADO1_ESPERA=====
ESTADO1:
    ACALL ES1_GEN_EV
    MOV A, EVENTO
    RL A
    MOV DPTR, #ES1_EV_TAB
    JMP @A+DPTR
ES1_EV_TAB:
    AJMP ES1_EV0
    AJMP ES1_EV1
    AJMP ES1_EV2
ES1_EV0:
    ;EVENTO VACIO
    RET
ES1_EV1:
    ;EVENTO REPOSO
    ;RESETEAR VARIABLES DE ESTADO
    MOV ES1_CONT_MS, #0
    MOV ES1_CONT_S, #0
    MOV CONT_ZUMB, #0
    MOV CONT_PARPADEO, #0
    ;
    ACALL APAGAR_DISPLAY
    ACALL APAGAR_PARPADEO
    ACALL APAGAR_ZUMBADOR
    ACALL APAGAR_LED
    ACALL APAGAR_TIMER

    MOV ESTADO, #0
    MOV EVENTO, #0
    RET
ES1_EV2:
    ;EVENTO CALENTAR
    ;RESETEAR VARIABLES DE ESTADO
    MOV ES1_CONT_MS, #0
    MOV ES1_CONT_S, #0
    MOV CONT_ZUMB, #0
    MOV CONT_PARPADEO, #0
    ;
    ACALL APAGAR_PARPADEO
    ACALL ENCENDER_DISPLAY
    ACALL APAGAR_ZUMBADOR
    ;ENCENDER PWM
    ACALL ENCENDER_PWM

```

```

MOV ESTADO, #2
MOV EVENTO, #0
RET
ES1_GEN_EV:
;COMPROBAR SI SE HA PULSADO EL SWITCH
MOV C, SWITCH
JC ES1_SWITCH
;COMPROBAR SI SE HA COLOCADO RECIPIENTE
MOV C, SENSOR
JC ES1_SENSOR
;COMPROBAR SI HA PASADO 100MS
MOV C, TICK
JC ES1_TICK
RET
ES1_SWITCH:                                ;SE DETECTA ON/OFF
MOV EVENTO, #1
RET
ES1_SENSOR:                                ;SE DETECTA RECIPIENTE
MOV EVENTO, #2
RET
ES1_TICK:                                  ;PASA 100MS SIN RECIPIENTE
CLR TICK
INC ES1_CONT_MS
MOV A, #10
CLR C
SUBB A, ES1_CONT_MS
JZ ES1_1S
RET
ES1_1S:                                    ;PASA 1S SIN RECIPIENTE
MOV ES1_CONT_MS, #0
INC ES1_CONT_S
MOV A, #15
CLR C
SUBB A, ES1_CONT_S
JZ ES1_15S
RET
ES1_15S:                                   ;PASA 15S SIN RECIPIENTE
MOV ES1_CONT_S, #0
MOV EVENTO, #1
RET
;=====ESTADO2_CALENTAR=====
ESTADO2:
ACALL ES2_GEN_EV
MOV A, EVENTO
RL A
MOV DPTR, #ES2_EV_TAB
JMP @A+DPTR
ES2_EV_TAB:
AJMP ES2_EV0
AJMP ES2_EV1
ES2_EV0:                                   ;EVENTO VACIO
RET
ES2_EV1:                                   ;EVENTO TRANSICION
;ENCENDER ZUMBADOR
ACALL ENCENDER_ZUMBADOR
;APAGAR LED
ACALL APAGAR_LED
;ENCENDER ADC
ACALL ENCENDER_ADC

MOV ESTADO, #3
MOV EVENTO, #0

```

```

    RET
ES2_GEN_EV:
    ;SE HA PULSADO EL BOTON ON/OFF
    MOV C, SWITCH
    JC ES2_SWITCH
    ;SE HA PULSADO EL BOTON DE MENOS
    MOV C, MINUS
    JC ES2_MINUS
    ;SE HA PULSADO EL BOTON DE MAS
    MOV C, PLUS
    JC ES2_PLUS
    ;PASA 100MS SIN RECIPIENTE
    MOV C, TICK
    JC ES2_TICK
    RET
ES2_SWITCH:
    MOV EVENTO, #1
    RET
ES2_MINUS:
    MOV A, #0
    CLR C
    SUBB A, VALOR_DISPLAY
    JZ ES2_NOTHING
    DEC VALOR_DISPLAY
    ACALL UPDATE_DISPLAY
    RET
ES2_PLUS:
    MOV A, #10
    CLR C
    SUBB A, VALOR_DISPLAY
    JZ ES2_NOTHING
    INC VALOR_DISPLAY
    ACALL UPDATE_DISPLAY
    RET
ES2_TICK:
    CLR TICK
    INC ES2_CONT_MS
    MOV A, #10
    CLR C
    SUBB A, ES2_CONT_MS
    JZ ES2_1S
    RET
ES2_1S:
    MOV ES2_CONT_MS, #0
    ACALL ES2_SENSOR
    RET
ES2_30S:
    MOV ES2_CONT1_S, #0
    MOV ES2_CONT2_S, #0
    MOV EVENTO, #1
    RET
ES2_60S:
    MOV ES2_CONT1_S, #0
    MOV ES2_CONT2_S, #0
    MOV EVENTO, #1
ES2_SENSOR:
    MOV C, SENSOR
    JC ES2_SENSOR_ON
    JNC ES2_SENSOR_OFF
ES2_SENSOR_ON:
    MOV A, VALOR_DISPLAY
    JNZ ES2_NOTHING

```

```

    MOV ES2_CONT1_S, #0
    INC ES2_CONT2_S
    MOV A, #60
    CLR C
    SUBB A, ES2_CONT2_S
    JZ ES2_60S
    RET
ES2_SENSOR_OFF:
    MOV ES2_CONT2_S, #0
    INC ES2_CONT1_S
    MOV A, #30
    CLR C
    SUBB A, ES2_CONT1_S
    JZ ES2_30S
    RET
ES2_NOTHING:
    MOV ES2_CONT1_S, #0
    MOV ES2_CONT1_S, #0
    RET
;=====ESTADO3_TRANSICION=====
ESTADO3:
    ACALL ES3_GEN_EV
    MOV A, EVENTO
    RL A
    MOV DPTR, #ES3_EV_TAB
    JMP @A+DPTR
ES3_EV_TAB:
    AJMP ES3_EV0
    AJMP ES3_EV1
ES3_EV0:                                ;EVENTO VACIO
    RET
ES3_EV1:                                ;EVENTO ESPERA
    MOV ESTADO, #0
    MOV EVENTO, #0
    RET
ES3_GEN_EV:
    MOV C, FADC
    JC ES3_ADC
    RET
ES3_ADC:
    ACALL LEER_ADC                        ;AHORA TENGO EN B EL VALOR DEL ADC

    ;MENOS DE 40
    MOV A, B
    CLR C
    SUBB A, #40
    JC ES3_40
    ;MAS DE 80
    MOV A, B
    CLR C
    SUBB A, #80
    JNC ES3_80
    ;ENTRE 40 Y 80
    AJMP ES3_60
ES3_40:
    MOV EVENTO, #1
    RET
ES3_60:
    MOV VALOR_DISPLAY, #11
    ACALL UPDATE_DISPLAY
    ACALL ENCENDER_ADC
    RET

```

```

ES3_80:
    MOV VALOR_DISPLAY, #12
    ACALL UPDATE_DISPLAY
    ACALL ENCENDER_ADC
    RET

;=====LED=====
ENCENDER_LED:
    SETB LED
    RET
APAGAR_LED:
    CLR LED
    RET

;=====PWM=====
ENCENDER_PWM:
    MOV PWMP, #1
    RET
APAGAR_PWM:
    MOV PWMP, #0
    RET

;=====DISPLAY=====
ENCENDER_PARPADEO:
    MOV CONT_PARPADEO, #0
    SETB PARPADEO
    RET
APAGAR_PARPADEO:
    MOV CONT_PARPADEO, #0
    CLR PARPADEO
    RET
ESTADO_PARPADEO:
    MOV C, PARPADEO
    JC PARPADEO_500MS
    RET
PARPADEO_500MS:
    ;HAN PASADO 500MS?
    INC CONT_PARPADEO
    MOV A, #5
    CLR C
    SUBB A, CONT_PARPADEO
    JZ UPDATE_PARPADEO
    RET
UPDATE_PARPADEO:
    MOV CONT_PARPADEO, #0
    MOV A, P0
    ANL A, #01111111b
    JNZ APAGAR_DISPLAY
    JZ ENCENDER_DISPLAY
    RET
APAGAR_DISPLAY:
    ANL P0, #0000000b
    RET
ENCENDER_DISPLAY:
    ACALL UPDATE_DISPLAY
    RET
UPDATE_DISPLAY:
    ANL P0, #00000000b
    MOV A, VALOR_DISPLAY
    RL A
    MOV DPTR, #DISPLAY_FUNC
    JMP @A+DPTR
DISPLAY_FUNC:
    AJMP DISPLAY_0 ;0

```

```

    AJMP DISPLAY_1    ;1
    AJMP DISPLAY_2    ;2
    AJMP DISPLAY_3    ;3
    AJMP DISPLAY_4    ;4
    AJMP DISPLAY_5    ;5
    AJMP DISPLAY_6    ;6
    AJMP DISPLAY_7    ;7
    AJMP DISPLAY_8    ;8
    AJMP DISPLAY_9    ;9
    AJMP DISPLAY_P    ;10
    AJMP DISPLAY_H    ;11
    AJMP DISPLAY_HH   ;12
DISPLAY_0:            ;*gfedcba
    ORL P0, #00111111b
    MOV PWM0, #255    ;0%
    RET
DISPLAY_1:
    ORL P0, #00000110b
    MOV PWM0, #230    ;10%
    RET
DISPLAY_2:
    ORL P0, #01011011b
    MOV PWM0, #204    ;20%
    RET
DISPLAY_3:
    ORL P0, #01001111b
    MOV PWM0, #178    ;30%
    RET
DISPLAY_4:
    ORL P0, #01111111b
    MOV PWM0, #153    ;40%
    RET
DISPLAY_5:
    ORL P0, #01101101b
    MOV PWM0, 127     ;50%
    RET
DISPLAY_6:
    ORL P0, #01111101b
    MOV PWM0, #102    ;60%
    RET
DISPLAY_7:
    ORL P0, #00001111b
    MOV PWM0, #78     ;70%
    RET
DISPLAY_8:
    ORL P0, #01111111b
    MOV PWM0, #52     ;80%
    RET
DISPLAY_9:
    ORL P0, #01101111b
    MOV PWM0, #26     ;90%
    RET
DISPLAY_P:
    ORL P0, #01110011b
    MOV PWM0, #0      ;100%
    RET
DISPLAY_H:
    ORL P0, #01110100b
    RET
DISPLAY_HH:
    ORL P0, #01110110b
    RET

```



```
;=====ZUMBADOR=====
```

```
ENCENDER_ZUMBADOR:
```

```
    MOV CONT_ZUMB, #0
    CLR ZUMBADOR
    RET
```

```
APAGAR_ZUMBADOR:
```

```
    MOV CONT_ZUMB, #0
    SETB ZUMBADOR
    RET
```

```
ZUMBADOR_ESTADO:
```

```
    ;ESTA EL ZUMBADOR ENCENDIDO?
    MOV C, ZUMBADOR
    JNC ZUMBADOR_200MS
    RET
```

```
ZUMBADOR_200MS:
```

```
    INC CONT_ZUMB
    MOV A, #2
    CLR C
    SUBB A, CONT_ZUMB
    JZ APAGAR_ZUMBADOR
    RET
```

```
;=====ADC=====
```

```
ENCENDER_ADC:
```

```
    ANL ADCON, #11111001b
    ORL ADCON, #00001001b
```

```
LEER_ADC:
```

```
    CLR FADC
    MOV B, ADCH
    RET
```

```
ORG 0x53
```

```
INTERRUPCION_ADC:
```

```
    SETB FADC
    RETI
```

```
;=====TIMER=====
```

```
ENCENDER_TIMER:
```

```
    ORL IEN0, #10000010b
    MOV TH0, #6
    MOV TL0, #6
    MOV T_CONT1, #0
    MOV T_CONT2, #0
    SETB TR0
    RET
```

```
APAGAR_TIMER:
```

```
    ANL IEN0, #00000000b
    CLR TR0
    RET
```

```
ORG 0x0B
```

```
INTERRUPCION_TIMER:
```

```
    PUSH PSW
    PUSH ACC
```

```
    INC T_CONT1
    MOV A, #127
    CLR C
    SUBB A, T_CONT1
    JNZ CONDTIMER
```

```
    MOV T_CONT1, #0
    INC T_CONT2
    MOV A, #100
    CLR C
    SUBB A, T_CONT2
```

```
JNZ CONDTIMER

MOV T_CONT2, #0
SETB TICK                      ;HA PASADO 100ms!

ACALL ZUMBADOR_ESTADO
ACALL ESTADO_PARPADEO

POP ACC
POP PSW
RETI

CONDTIMER:
POP ACC
POP PSW
RETI

;=====END=====
END

;HA FECHA 13/10/2022 SOLO YO Y DIOS SABIAMOS LO QUE AQUI HABIA ESCRITO.
;CUENDO ESTES LEYENDO ESTO, SEGURAMENTE, SOLO DIOS LO SEPA.
```