

Arquitectura de Computadores:

Práctica Final de Laboratorio 2022-2023

**Departamento:**

Tecnología Electrónica

**Titulación:**

Grado en Informática de Gestión y Sistemas de Información

Xabier Gabiña Barañano

Ainhize Martínez Duran

Contenido

[INTRODUCCION 2](#_Toc119418518)

[TABLAS Y DIAGRAMAS 3](#_Toc119418519)

[TABLAS DE ESTADO/EVENTO/ACCION 3](#_Toc119418520)

[DIAGRAMA DE ESTADO/EVENTO/ACCION 5](#_Toc119418521)

[DIAGRAMAS DE FLUJO 6](#_Toc119418522)

[CALCULOS Y COMENTARIOS 10](#_Toc119418523)

[ADC 10](#_Toc119418524)

[TIMER 10](#_Toc119418525)

[PWM 11](#_Toc119418526)

[COMENTARIOS 11](#_Toc119418527)

[CODIGO 12](#_Toc119418528)

# INTRODUCCION

En esta práctica se pretende realizar el software de control de una **placa de inducción de un fogón** mediante la utilización del **microcontrolador** **80C552 de Philips**.

Nuestra tarea será programar diversas funciones tales como dos pulsadores capacitivos, un **DISPLAY**, un **led**, un **zumbador**, un **ADC**, un **PWM** y un **TIMER** con los conocimientos obtenidos en las clases de practica de aula y de teoría.

Todo esto como he dicho antes se hará haciendo uso del **microcontrolador 80C552**, un derivado del **80C51**. Y será programado y testado en el entorno de desarrollo “**Keil uVision 2**”.



# TABLAS Y DIAGRAMAS

Todos estos diagramas estarán incluidos en la carpeta de la entrega para mayor calidad del visionado.

## TABLAS DE ESTADO/EVENTO/ACCION

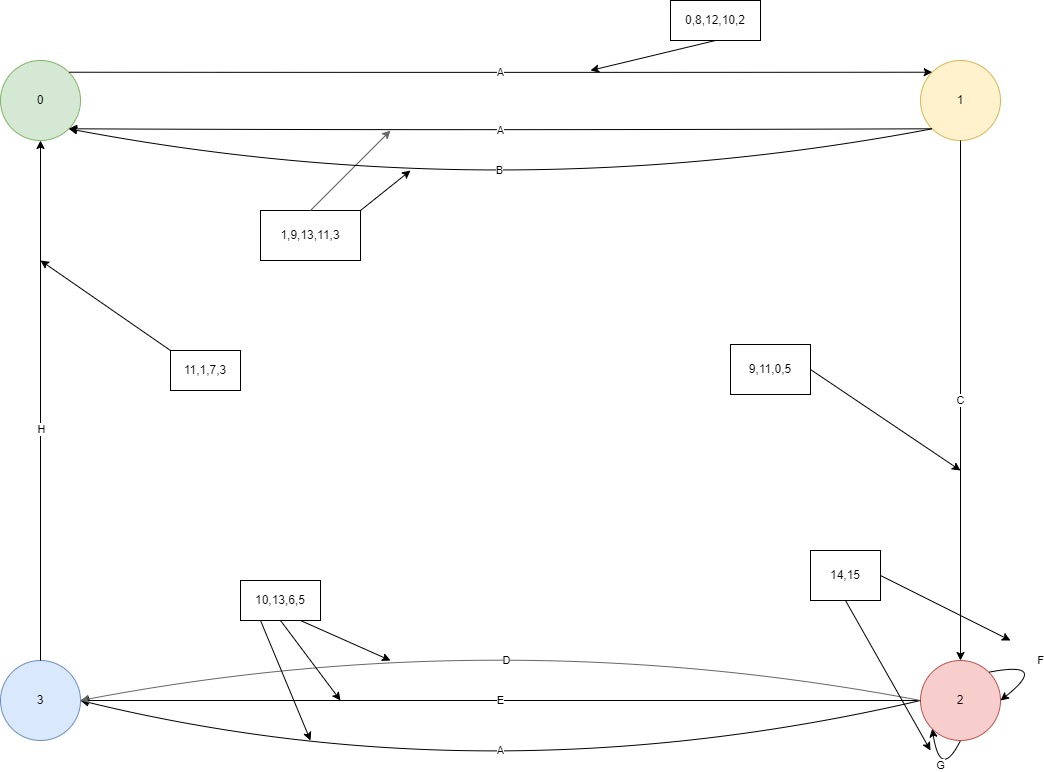
|  |  |
| --- | --- |
| **TABLA DE ESTADOS** | |
| *COD* | *NOMBRE* |
| 0 | REPOSO |
| 1 | ESPERA |
| 2 | CALENTAR |
| 3 | TRANSICION |

|  |  |
| --- | --- |
| **TABLA DE EVENTOS** | |
| *COD* | *NOMBRE* |
| A | SE HA PULSADO ON/OFF |
| B | HAN PASADO 15s SIN RECIPIENTE |
| C | SE HA COLOCADO RECIPIENTE |
| D | HAN PASADO 60s SIN AUMENTAR POTENCIA |
| E | HAN PASADO 30s SIN RECIPIENTE |
| F | SE HA PULSADO EL BOTON + |
| G | SE HA PULSADO EL BOTON - |
| H | LA TEMPERATURA ES INFERIOR A 40 |

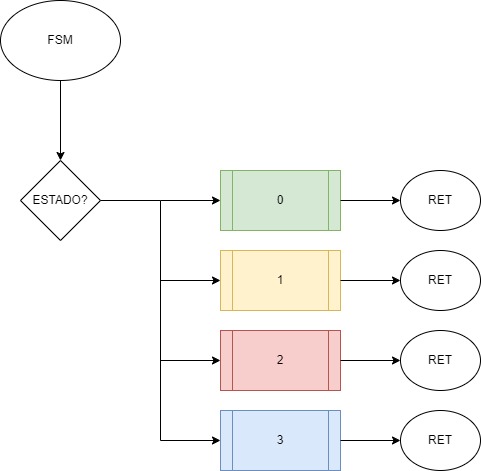
|  |  |  |
| --- | --- | --- |
| **TABLA DE EVENTOS POR ESTADO** | | |
| *COD ESTADO* | *COD EVENTO* EN ESTADO | *COD EVENTO* |
| 0 | 1 | A |
| 1 | 1 | A, B |
| 2 | C |
| 2 | 1 | A, D, E |
| 2 | G |
| 3 | F |
| 3 | 1 | H |

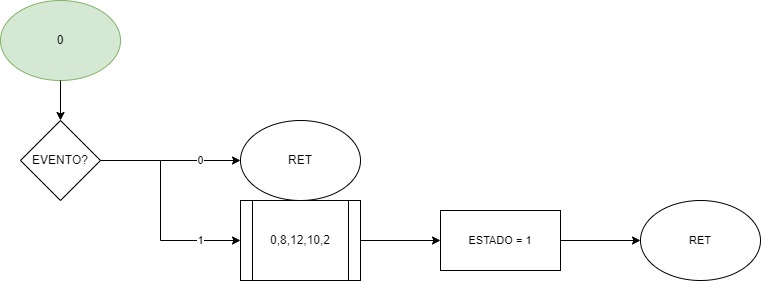
|  |  |
| --- | --- |
| **TABLA DE ACCIONES** | |
| 0 | ENCENDER DISPLAY |
| 1 | APAGAR DISPLAY |
| 2 | ENCENDER TIMER |
| 3 | APAGAR TIMER |
| 4 | ENCENDER PWM |
| 5 | APAGAR PWM |
| 6 | ENCENDER ADC |
| 7 | APAGAR ADC |
| 8 | ENCENDER PARPADEO |
| 9 | APAGAR PARPADEO |
| 10 | ENCENDER ZUMBADOR |
| 11 | APAGAR ZUMBADOR |
| 12 | ENCENDER LED |
| 13 | APAGAR LED |
| 14 | +/- VALOR FOGÓN |
| 15 | ACTUALIZAR DISPLAY |

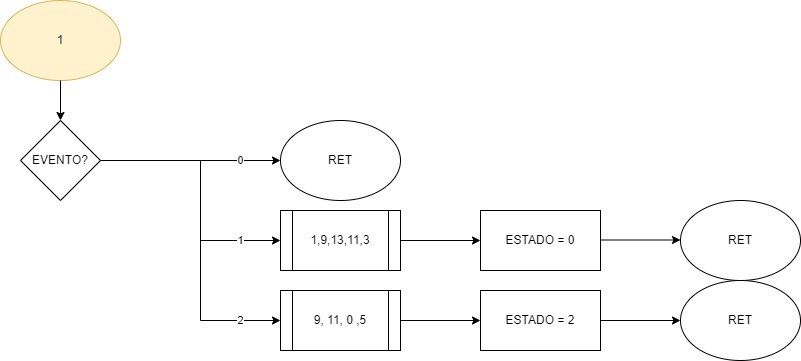
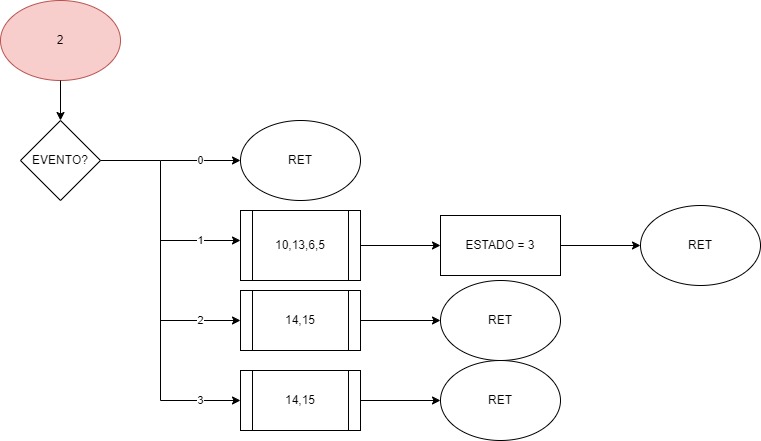
## DIAGRAMA DE ESTADO/EVENTO/ACCION

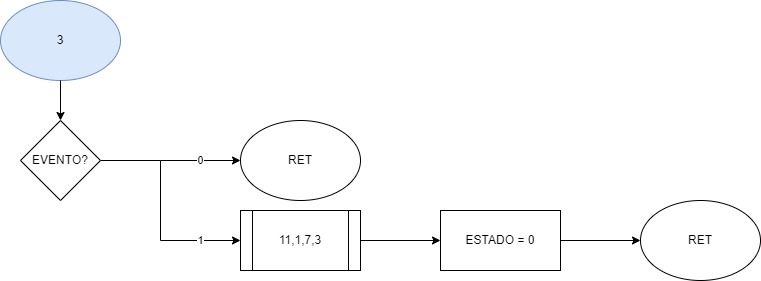
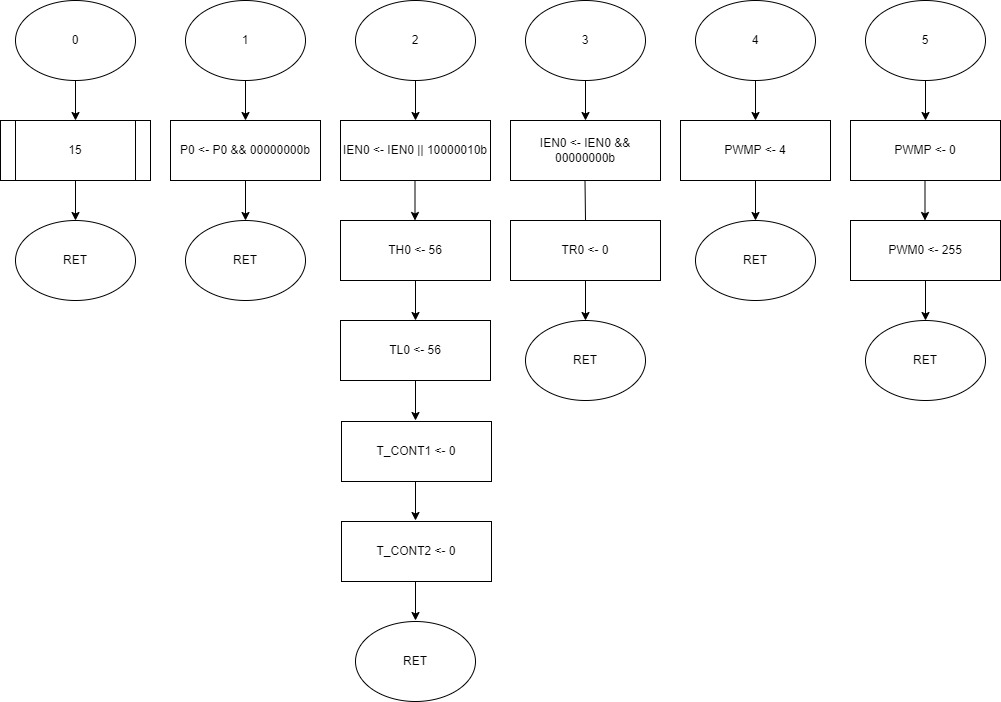
[](IMG/DIAGRAMA%20EEA.jpg)

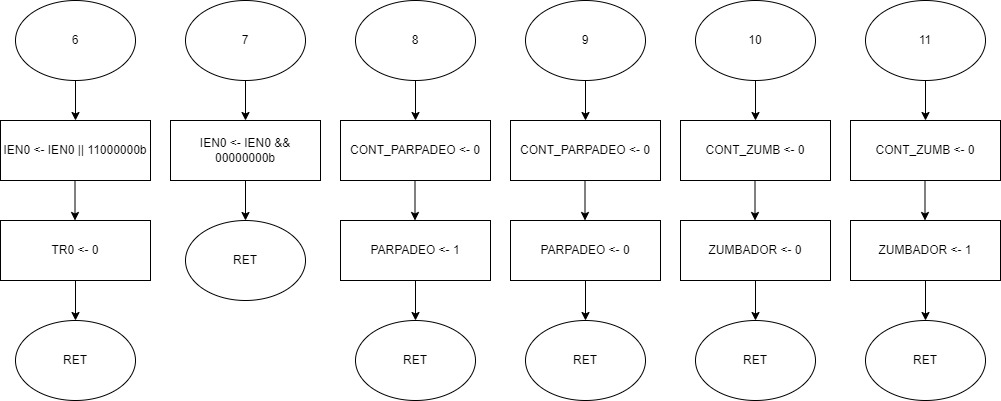
## DIAGRAMAS DE FLUJO

[](IMG/FSM.jpg)

[](IMG/REPOSO.jpg)

[](IMG/ESPERA.jpg)[](IMG/CALENTAR.jpg)

[](IMG/TRANSICION.jpg)



# CALCULOS Y COMENTARIOS

## ADC

La temperatura del fogón de inducción se mide mediante el canal 0 del convertidor analógico-digital del **80C552**. El valor que este nos arroja es proporcional a la temperatura que el sensor lee.

Para este trabajo necesitamos conocer el valor que arrojara cuando la temperatura del fogón sea de **40** y de **80** grados centígrados. Sabemos que el sensor sigue una progresión de 10mV por cada Cº luego podemos calcular todo lo necesario.

Para calcular el resultado del **SFR ADCH** podemos usar la siguiente formula.

En la cual tras sustituir los valores obtendremos los resultados.

## TIMER

En el **80C552** tenemos varios “**Timers**” a nuestra disposición con diferentes modos y funciones especiales. Para nuestro proyecto hemos usado exclusivamente el **Timer1** en modo de **8bits con auto recarga**.

Como se nos especifica en la documentación del programa contamos con un cristal de cuarzo que oscila a **24MHz**. Sabiendo que por cada **doce ciclos de reloj** tenemos **un ciclo maquina** podemos calcular que la frecuencia de instrucción es de **2MHz**, es decir, se tarda **0,5μs** en ejecutar cada instrucción. Con ello queremos que el “**timer**” active una “**flag**” que nos indique el paso de 100ms para lo cual realizamos los siguientes cálculos:

To = 0,5μs = 0,0005ms

Te = 100ms

Te/To = 200000

200000 = 200 \* 125 \* 8

200 -> 256-200 = **56** -> Precarga de timer (**TH0** y **TL0**)

**125** -> Contador 1 del programa

**8** -> Contador 2 del programa

## PWM

La temperatura del fogón de inducción viene controlada por un **PWM.** Dependiendo el nivel en el que se encuentre el fogón dará un porcentaje de potencia equivalente, es decir, en el nivel 0 dará 0% de potencia, en el nivel 1 dará 10% de potencia y así hasta la P que será el 100% de potencia. Por lo tanto, podemos calcular los valores que tendrá **PWM0.**

3

5

9

Ahora nos indican que el **PWM** tiene una frecuencia de conmutación de 10kHz por lo que podemos calcular el valor del **PWMP** usando la siguiente formula:

Sabiendo que la **Fosc** es la del reloj, es decir, 24Mhz y **Fpwm** son los 10kHz que se piden sustituimos:

## COMENTARIOS

En general el proyecto nos ha parecido que tenia una dificultad bastante aceptable y asequible. En nuestro caso íbamos programando cada cosa a medida que dábamos su teoría en clase por lo que no nos ha resultado demasiado difícil ningún apartado quitando el echo de resolver los cálculos de los diferentes módulos como el **ADC**, **TIMER** y **PWM**.

Si tuviese que resaltar algo me centraría sobre todo el en **TIMER** el cual se usa mucho en todas partes del programa y tuvimos alguna dificultad para elegir la frecuencia del reloj, pero nada imposible de hacer.

# CODIGO

;==============================================================================================;

;VARIABLES GLOBALES

ESTADO EQU 0x20 ;BYTE DONDE SE GUARDA EL ESTADO

EVENTO EQU 0x21 ;BYTE DONDE SE GUARDA EL EVENTO

VALOR\_DISPLAY EQU 0x22 ;BYTE DEL VALOR QUE SE MUESTRA EN EL DISPLAY

T\_CONT1 EQU 0x23 ;BYTE DONDE SE GUARDA EL CONTADOR[1] USADO EN EL TIMER

T\_CONT2 EQU 0x24 ;BYTE DONDE SE GUARDA EL CONTADOR[2] USADO EN EL TIMER

TICK EQU 0x25.0 ;BIT DONDE SE GUARDA LA FLAG DE AVISO PARA 100MS

PARPADEO EQU 0x25.1 ;BIT DONDE SE GUARDA LA FLAG DE AVISO DEL ESTADO DE PARPADEO

FADC EQU 0X25.2 ;BIT DONDE SE GUARDA LA FLAG DE AVISO DEL ADC

CONT\_ZUMB EQU 0x26 ;CONTADOR USADO EN EL ZUMBADOR

CONT\_PARPADEO EQU 0x27 ;CONTADOR USADO EN EL PARPADEO

ADCON EQU 0xC5 ;BYTE DE CONFIGURACION DE ADC

ADCH EQU 0xC6 ;BYTE DONDE SE GUARDA EL VALOR DE LECTURA DEL ADC

PWMP EQU 0xFE ;BYTE DEL PRESCALER DEL PWM

PWM0 EQU 0xFC ;BYTE DONDE SE GUARDA EL DUTY CICLE

IEN0 EQU 0xA8 ;BYTE DE CONFIGURACION DE LAS INTERRUPCIONES

;PUERTOS

LED EQU P2.0 ;PUERTO AL QUE ESTA CONECTADO EL LED

ZUMBADOR EQU P2.1 ;PUERTO AL QUE ESTA CONECTADO EL ZUMBADOR

PLUS EQU P3.0 ;PUERTO AL QUE ESTA CONECTADO EL PULSADOR +

MINUS EQU P3.1 ;PUERTO AL QUE ESTA CONECTADO EL PULSADOR -

SWITCH EQU P3.2 ;PUERTO AL QUE ESTA CONECTADO EL PULSADOR ON/OFF

SENSOR EQU P3.5 ;PUERTO AL QUE ESTA CONECTADO EL SENSOR

;VARIABLES DE ESTADOS

ES1\_CONT\_MS EQU 0x2A ;CONTADOR USADO EN EL ESTADO1 PARA CONTAR LOS TICKS PASADOS

ES1\_CONT\_S EQU 0x2B ;CONTADOR USADO EN EL ESTADO1 PARA CONTAR LOS S PASADOS

ES2\_CONT\_MS EQU 0x2C ;CONTADOR USADO EN EL ESTADO2 PARA CONTAR LOS TICKS PASADOS

ES2\_CONT1\_S EQU 0x2D ;CONTADOR USADO EN EL ESTADO2 PARA CONTAR LOS S PASADOS

;SIN SENSOR!!!

ES2\_CONT2\_S EQU 0x2E ;CONTADOR USADO EN EL ESTADO2 PARA CONTAR LOS S PASADOS

;CON SENSOR!!!

;AJUSTES TIMER

TIMER\_CONT\_1 EQU 125 ;DEFAULT = 125

TIMER\_CONT\_2 EQU 8 ;DEFAULT = 8

;NORMALMENTE EL TIMER ACTIVA LA FLAG "TICK" CADA 100MS

;PARA OBTENER LA FLAG DIEZ VECES MAS RAPIDO ABRIA QUE

;USAR 50 y 2 RESPECTIVAMENTE AUNQUE EXISTEN

;MAS COMBINACIONES COMO 100 Y 1

;============================================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:

;DEBUG

MOV P0, #0

MOV P2, #0

MOV P3, #0

;VALORES GLOBALES

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 ;ACTIVO A NIVEL BAJO!

;TIMER EN MODO 8BIT AUTORECARGA

ORL TMOD, #00000010b

ANL TMOD, #11110010b

CLR TICK

;VARIABLES DE ESTADOS

MOV ES1\_CONT\_MS, #0

MOV ES1\_CONT\_S, #0

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

AJMP ES0\_EV1

AJMP ES0\_EV2

ES0\_EV0:

RET

ES0\_EV1: ;EVENTO VACIO

AJMP ACTIVAR\_IDL

RET

ES0\_EV2: ;EVENTO ESPERA

MOV VALOR\_DISPLAY, #0

ACALL UPDATE\_DISPLAY

ACALL ENCENDER\_PARPADEO

ACALL ENCENDER\_LED

ACALL ENCENDER\_ZUMBADOR

ACALL ENCENDER\_TIMER

MOV ESTADO, #1

RET

ES0\_GEN\_EV:

MOV C, SWITCH

JNC ES0\_SWITCH ;COMPROBAR SI SE HA PULSADO EL BOTON

MOV EVENTO, #1 ;SIEMPRE SE DA EL EVENTO 0 SI NO SE HA PULSADO EL SWITCH

RET

ES0\_SWITCH: ;SI EL SWITCH SE HA PULSADO PASA SE DA EL EVENTO 1

MOV EVENTO, #2

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

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

ACALL ENCENDER\_PWM

MOV ESTADO, #2

RET

ES1\_GEN\_EV:

MOV C, SWITCH

JNC ES1\_SWITCH

MOV C, SENSOR

JC ES1\_SENSOR

MOV C, TICK

JC ES1\_TICK

MOV EVENTO, #0

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

AJMP ES2\_EV2

AJMP ES2\_EV3

ES2\_EV0: ;EVENTO VACIO

RET

ES2\_EV1: ;EVENTO TRANSICION

MOV ES2\_CONT\_MS, #0

MOV ES2\_CONT1\_S, #0

MOV ES2\_CONT2\_S, #0

ACALL ENCENDER\_ZUMBADOR

ACALL APAGAR\_LED

ACALL ENCENDER\_ADC

ACALL APAGAR\_PWM

MOV ESTADO, #3

RET

ES2\_EV2: ;EVENTO PULSAR BOTON MENOS

MOV A, #0

CLR C

SUBB A, VALOR\_DISPLAY

JZ ES2\_NOTHING

DEC VALOR\_DISPLAY

ACALL UPDATE\_DISPLAY

RET

ES2\_EV3: ;EVENTO PULSAR BOTON MAS

MOV A, #10

CLR C

SUBB A, VALOR\_DISPLAY

JZ ES2\_NOTHING

INC VALOR\_DISPLAY

ACALL UPDATE\_DISPLAY

RET

ES2\_GEN\_EV:

MOV C, SWITCH

JNC ES2\_SWITCH ;SALTO SI SE PULSA EL BOTON ON/OFF

MOV C, MINUS ;SALTO SI SE PULSA EL BOTON -

JC ES2\_MINUS

MOV C, PLUS ;SALTO SI SE PULSA EL BOTON +

JC ES2\_PLUS

MOV C, TICK ;SALTO SI HAN PASADO 100MS

JC ES2\_TICK

MOV EVENTO, #0

RET

ES2\_SWITCH: ;SE HA PULSADO EL BOTON ON/OFF

MOV EVENTO, #1

RET

ES2\_MINUS: ;SE HA PULSADO EL BOTON DE MENOS

MOV EVENTO, #2

RET

ES2\_PLUS: ;SE HA PULSADO EL BOTON DE MAS

MOV EVENTO, #3

RET

ES2\_TICK: ;PASA 100MS

CLR TICK

INC ES2\_CONT\_MS

MOV A, #10

CLR C

SUBB A, ES2\_CONT\_MS

JZ ES2\_1S

RET

ES2\_1S: ;PASA 1S

MOV ES2\_CONT\_MS, #0

ACALL ES2\_SENSOR

RET

ES2\_30S: ;PASAN 30S

MOV ES2\_CONT1\_S, #0

MOV ES2\_CONT2\_S, #0

MOV EVENTO, #1

RET

ES2\_60S: ;PASAN 60S

MOV ES2\_CONT1\_S, #0

MOV ES2\_CONT2\_S, #0

MOV EVENTO, #1

ES2\_SENSOR: ;DICE SI EL RECIPIENTE ESTA O NO

MOV C, SENSOR

JC ES2\_SENSOR\_ON

JNC ES2\_SENSOR\_OFF

ES2\_SENSOR\_ON: ;SI ESTA RECIPIENTE SE CUENTA HASTA 60S

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: ;NO ESTA RECIPIENTE SE CUENTA HASTA 30S

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: ;RESET DE LOS CONTADORES

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

ACALL APAGAR\_TIMER

ACALL APAGAR\_ZUMBADOR

ACALL APAGAR\_DISPLAY

ACALL APAGAR\_ADC

MOV ESTADO, #0

RET

ES3\_GEN\_EV:

MOV C, FADC

JC ES3\_ADC ;EL ADC HA TERMINADO LA CONVERSION

MOV EVENTO, #0

RET

ES3\_ADC: ;UNA VEZ TERMINADO SE LEE EL VALOR Y SE GUARDA EN B PARA LUEGO COMPARARLO

ACALL LEER\_ADC

MOV A, B

CLR C

SUBB A, #20d

JC ES3\_40

MOV A, B

CLR C

SUBB A, #40d

JNC ES3\_80

AJMP ES3\_60

ES3\_40: ;LA TEMPERATURA ES MENOS A 40

MOV EVENTO, #1

RET

ES3\_60: ;LA TEMPERATURA ESTA ENTRE 40 Y 80

MOV VALOR\_DISPLAY, #11

ACALL UPDATE\_DISPLAY

ACALL ENCENDER\_ADC

RET

ES3\_80: ;LA TEMPERATURA ES MAYOR A 80

MOV VALOR\_DISPLAY, #12

ACALL UPDATE\_DISPLAY

ACALL ENCENDER\_ADC

RET

;==============================================LED=============================================;

ENCENDER\_LED: ;ENCIENDE EL LED

SETB LED

RET

APAGAR\_LED: ;APAGA EL LED

CLR LED

RET

;==============================================PWM=============================================;

ENCENDER\_PWM: ;ENCIENDE EL PWM

MOV PWMP, #4d

RET

APAGAR\_PWM: ;APAGA EL PWM

MOV PWMP, #0

MOV PWM0, #255

RET

;============================================DISPLAY===========================================;

ENCENDER\_PARPADEO: ;ACTIVA LA FUNCION DE PARPADEO

MOV CONT\_PARPADEO, #0

SETB PARPADEO ;ACTIVA LA FLAG DE PARPADEO

RET

APAGAR\_PARPADEO: ;DESACTIVA LA FUNCION DE PARPADEO

MOV CONT\_PARPADEO, #0

CLR PARPADEO ;DESACTIVA LA FLAG DE PARPADEO

RET

ESTADO\_PARPADEO: ;NOS DICE SI LA FLAG DE PARPADEO ESTA ACTIVA

MOV C, PARPADEO

JC PARPADEO\_500MS

RET

PARPADEO\_500MS: ;MIRA SI HAN PASADO 500MS DESDE EL ULTIMO PARPADEO

;HAN PASADO 500MS?

INC CONT\_PARPADEO

MOV A, #5

CLR C

SUBB A, CONT\_PARPADEO

JZ UPDATE\_PARPADEO

RET

UPDATE\_PARPADEO: ;SI HAN PASADO 500MS CAMBIA EL ESTADO DEL DISPLAY

MOV CONT\_PARPADEO, #0

MOV A, P0

ANL A, #01111111b

JNZ APAGAR\_DISPLAY

JZ ENCENDER\_DISPLAY

RET

APAGAR\_DISPLAY: ;APAGA EL DISPLAY

ANL P0, #0000000b

RET

ENCENDER\_DISPLAY: ;ENCIENDE EL DISPLAY CARGANDO EL VALOR DE LA VARIABLE

;VALOR DISPLAY

ACALL UPDATE\_DISPLAY

RET

UPDATE\_DISPLAY: ;ACTUALIZA EL VALOR DEL DISPLAY Y DEL PWM!!!

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, #229 ;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, #76 ;70%

RET

DISPLAY\_8:

ORL P0, #01111111b

MOV PWM0, #51 ;80%

RET

DISPLAY\_9:

ORL P0, #01101111b

MOV PWM0, #25 ;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: ;ENCIENDE EL ZUMBADO A NIVEL BAJO!

MOV CONT\_ZUMB, #0

CLR ZUMBADOR

RET

APAGAR\_ZUMBADOR: ;APAGA EL ZUMBADOR

MOV CONT\_ZUMB, #0

SETB ZUMBADOR

RET

ZUMBADOR\_ESTADO: ;NOS DICE SI EL ZUMBADOR ESTA ENCENDIDO O NO

MOV C, ZUMBADOR

JNC ZUMBADOR\_200MS

RET

ZUMBADOR\_200MS: ;COMPRUEBA SI HAN PASADO 200MS DESDE ENCENDER EL ZUMBADOR

INC CONT\_ZUMB

MOV A, #2

CLR C

SUBB A, CONT\_ZUMB

JZ APAGAR\_ZUMBADOR

RET

;==============================================IDL=============================================;

ACTIVAR\_IDL: ;ACTIVA EL MODO IDL Y LAS INTERRUPCIONES EXTERNAS

ORL IEN0, #10000001b

ORL PCON, #00000001b

RET

DESACTIVAR\_IDL: ;DESACTIVA TANTO EL IDL COMO LAS INTERRUPCIONES EXTERNAS

PUSH PSW

PUSH ACC

ANL IEN0, #00000000b

ANL PCON, #00000000b

POP ACC

POP PSW

RET

;==============================================ADC=============================================;

ENCENDER\_ADC: ;ENCIENDE EL ADC Y LO CONFIGURA

ORL IEN0,#11000000b

ORL ADCON, #00001000b

RET

APAGAR\_ADC: ;APAGA EL ADC

ANL IEN0, #00000000b

RET

LEER\_ADC: ;LEE EL VALOR DEL ADC Y LO GUARDA EN B

CLR FADC

MOV B, ADCH

RET

;=============================================TIMER============================================;

ENCENDER\_TIMER: ;ENCIENDE EL TIMER Y LE CONFIGURA EL VALOR DE AUTORECARGA

ORL IEN0, #10000010b

MOV TH0, #56

MOV TL0, #56

MOV T\_CONT1, #0

MOV T\_CONT2, #0

SETB TR0

RET

APAGAR\_TIMER: ;APAGA EL TIMER

ANL IEN0, #00000000b

CLR TR0

RET

TIMER\_FUNC: ;FUNCION DEL TIMER

PUSH PSW

PUSH ACC

INC T\_CONT1

MOV A, TIMER\_CONT\_1

CLR C

SUBB A, T\_CONT1

JNZ CONDTIMER

MOV T\_CONT1, #0

INC T\_CONT2

MOV A, TIMER\_CONT\_2

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

RET

CONDTIMER: ;SI NO SE CUMPLE UN CONTADOR DEL TIMER

POP ACC

POP PSW

RET

;=========================================INTERRUPCIONES=======================================;

ORG 0x03

INTERRUPCION\_EXTERNA:

ACALL DESACTIVAR\_IDL

RETI

ORG 0x0B

INTERRUPCION\_TIMER:

ACALL TIMER\_FUNC

RETI

ORG 0x53

INTERRUPCION\_ADC:

PUSH PSW

PUSH ACC

SETB FADC

POP ACC

POP PSW

RETI

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

END