



BILBOKO
INGENIARITZA
ESKOLA
ESCUELA
DE INGENIERÍA
DE BILBAO

Arquitectura de Computadores

Proyecto

Titulación:

Grado en Informática de Gestión y Sistemas de Información

4º curso (1º cuatrimestre)

Fecha: 1-12-2025

López, Aldonza, Iker

Índice

1. Introducción	2
2. Diagramas de estados y flujo	3
2.1. Diagrama de estados	3
2.2. Tablas de estados, eventos y acciones	3
2.3. Diagramas de flujo	6
2.3.1. Secuencia Principal	6
2.3.2. Interrupciones	7
2.3.3. Maq. Eventos 'Calentando Agua'	9
3. Cálculos y comentarios	12
3.1. Cálculo Sobrepeso	12
3.2. Cálculo Temperaturas	12
3.3. Cálculo timer0 1s	13
3.4. Cálculos PWM	14
3.5. Comentarios sobre Acciones Destacables	14
3.5.1. Mapear minutos a Displays	14
3.6. Lógica de Cuenta de Tiempo	15
4. Código	16

1. Introducción

Para este proyecto se requería de codificar una lavadora funcional con varios modos de temperatura, una función de pesado, dos velocidades de lavado que se extienden en el tiempo y la capacidad de iniciar tantos lavados como queramos.

Para ello surgen varios problemas que hemos de resolver con las herramientas disponibles en un procesador 80C552 de Phillips. En cuanto a mediciones de tiempo existen: **1s, 15s, 1min, 10min, 50min**. Todos se medirán con el **timer0** del procesador. Este medirá en su modo 1 generando a las 30,5 aprox. interrupciones un 1s con este axioma podemos generar el resto de tiempos.

Para las mediciones de peso y temperatura se usa el **Conversor Analógico Digital (ADC)** el cual por el canal 0 medirá el peso y por el canal 1 medirá la temperatura.

Para controlar la potencia del motor usaremos el **PulseWidthModulator (PWM)** en dos modos uno que hará girar a 60RPM Y otro que hará girar a 600RPM.

En cuanto a las decisiones de diseño se ha optado siempre por minimizar dentro de lo lógico y razonable el número de estados, aumenta así el número de eventos por estado. Por otro lado los cambios de estado se gestionan desde un generador de eventos por cada propio estado, es decir 9 Estados \rightarrow 9 Generadores de Eventos.

2. Diagramas de estados y flujo

2.1. Diagrama de estados

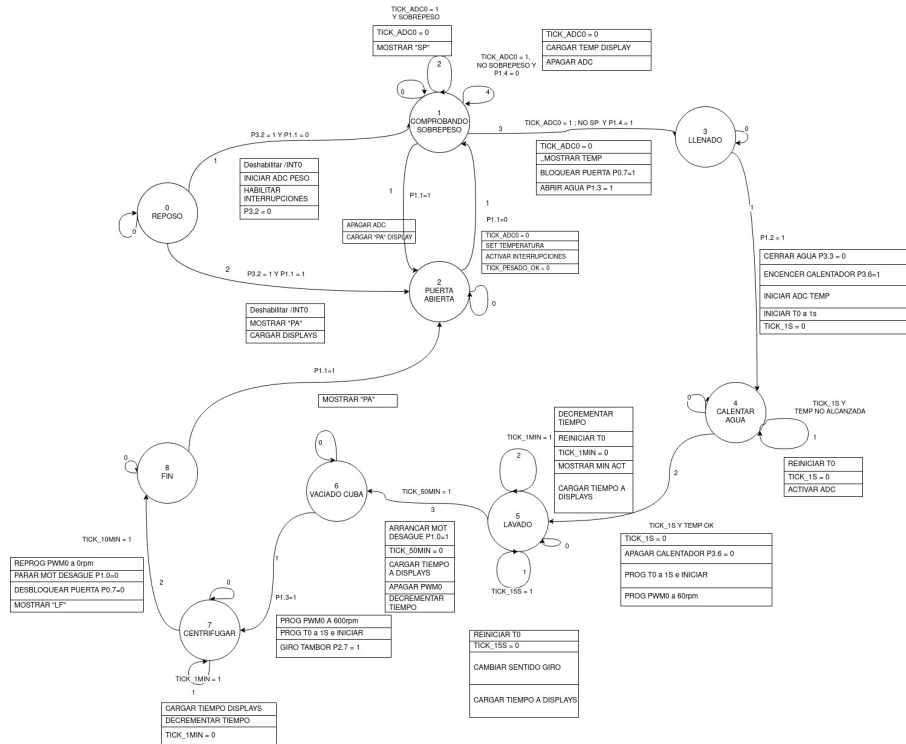


Figura 1: Diagrama de estados de la lavadora.

2.2. Tablas de estados, eventos y acciones

Cuadro 1: Tabla de estados

Nº estado	Nombre del estado
0	Reposo
1	Comprobación sobrepeso
2	Puerta abierta
3	Llenando
4	Calentar cuba
5	Lavado
6	Vaciar agua
7	Centrifugar
8	Fin

Tabla de Eventos

Nº Estado	Nº Evento	Condición	Acciones
0	0	Botón apagado (P3.2 = 1)	A1
0	1	Botón encendido + puerta abierta	Estado \leftarrow 2, mostrar "PA"
0	2	Botón encendido + puerta cerrada	Estado \leftarrow 1, A2
1	0	Sin evento	–
1	1	Puerta abierta	Estado \leftarrow 2, A5
1	2	Sobrepeso detectado	Mostrar "SP", Estado \leftarrow 1
1	3	Peso OK + botón inicio pulsado	A16, A17, Estado \leftarrow 3
1	4	Peso OK + botón inicio no pulsado	A6
2	0	Sin evento	–
2	1	Puerta cerrada	Estado \leftarrow 1, A7, A2
3	0	Sin evento	–
3	1	Sensor de llenado activado	A18, A19, Estado \rightarrow 4
4	0	Sin evento	–
4	1	Agua aún fría	–
4	2	Agua caliente alcanzada	A20, A10, Estado \leftarrow 5
5	0	Sin evento	–
5	1	Tick 15 s	A13, A3, A4
5	2	Tick 1 min	A23, A3, A4
5	3	Tick 50 min	Estado \leftarrow 6, A21
6	0	Sin evento	–
6	1	Sensor vaciado activado	Estado \leftarrow 7, A14
7	0	Sin evento	–
7	1	Tick 10 min	A15, A22, mostrar "LF", Estado \leftarrow 8
7	2	Tick 1 min	A23, A3, A4
8	0	Sin evento	–
8	1	Puerta abierta	Estado \leftarrow 2, mostrar "PA"

*ACCIONES DE **SETB Y CLR** SE MENCIONAN EN LA TABLA ANTERIOR. ESAS ACCIONES NO SON CONSIDERADAS EN ESTA TABLA*

Tabla de Códigos de Acciones

Código	Descripción de la acción
A1	Entrar en modo IDLE (ORL PCON,#01H)
A2	Activar ADC Peso
A3	Cargar Display 0
A4	Cargar Display 1
A5	Apagar el ADC
A6	Cargar la TMP seleccionada en Display 0
A7	Set la TEMPERATURA seleccionada
A8	Activar el ADC Temperatura
A9	Activar Timer 0
A10	Activar PWM 0 a 60RPM
A11	Mapear Tiempo restante a Displays 0 y 1
A12	Apagar el Timer 0
A13	Cambiar el Sentido de Giro del Tambor
A14	Activar el PWM 0 a 600RPM
A15	Apagar el PWM 0
A16	Bloquear puerta
A17	Abrir válvula de llenado
A18	Cerrar válvula de llenado
A19	Encender calentador
A20	Apagar calentador
A21	Encender desagüe
A22	Desbloquear puerta
A23	Decrementar tiempo

2.3. Diagramas de flujo

2.3.1. Secuencia Principal

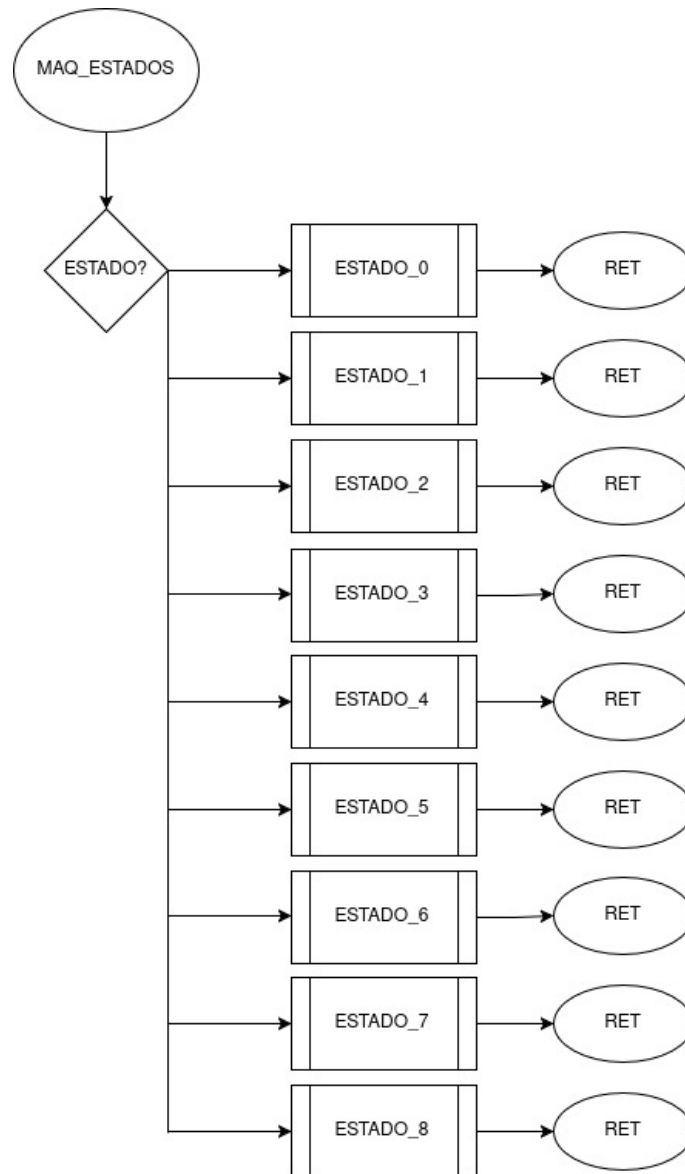


Figura 2: Diagrama bucle principal.

2.3.2. Interrupciones

- Interrupción ADC

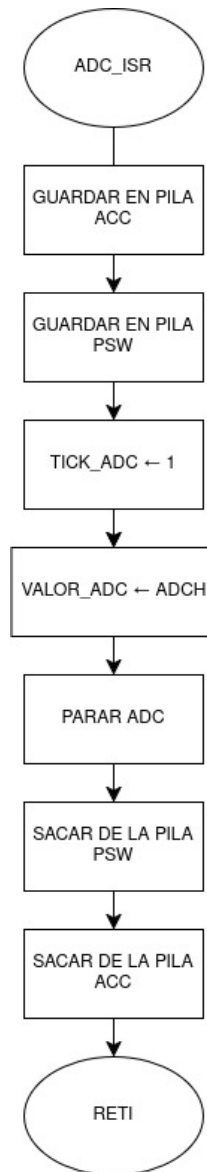
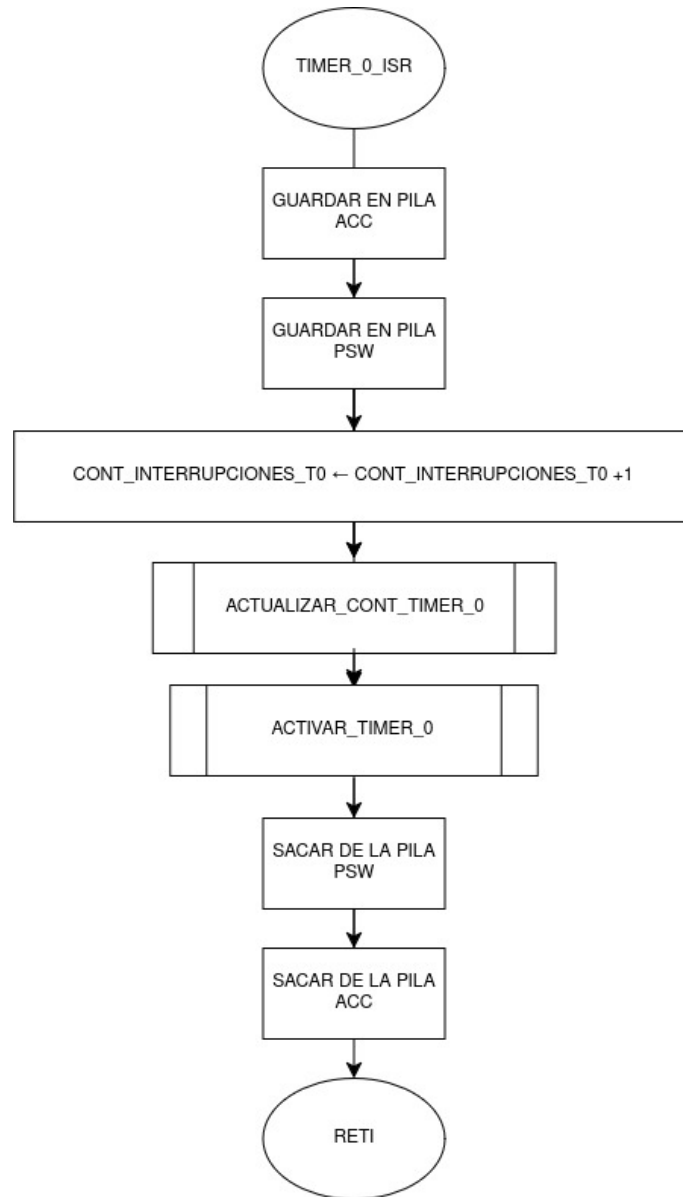


Figura 3: Diagrama Interrupción ADC

■ Interrupción Timer

*Figura 4: Diagrama Interrupción Timer 0.*

2.3.3. Maq. Eventos 'Calentando Agua'

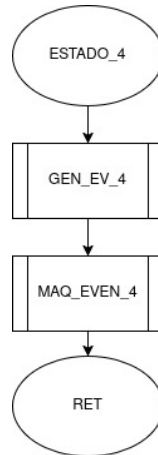


Figura 5: Diagrama ESTADO 4.

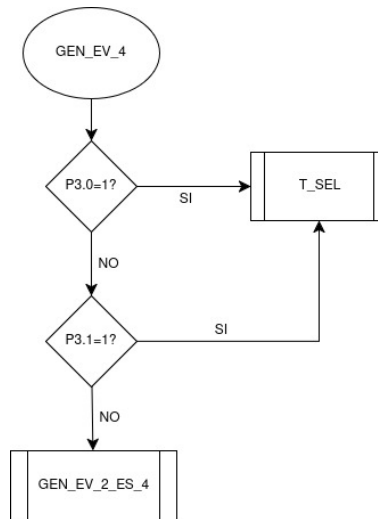


Figura 6: Diagrama Generador de Eventos ESTADO 4.

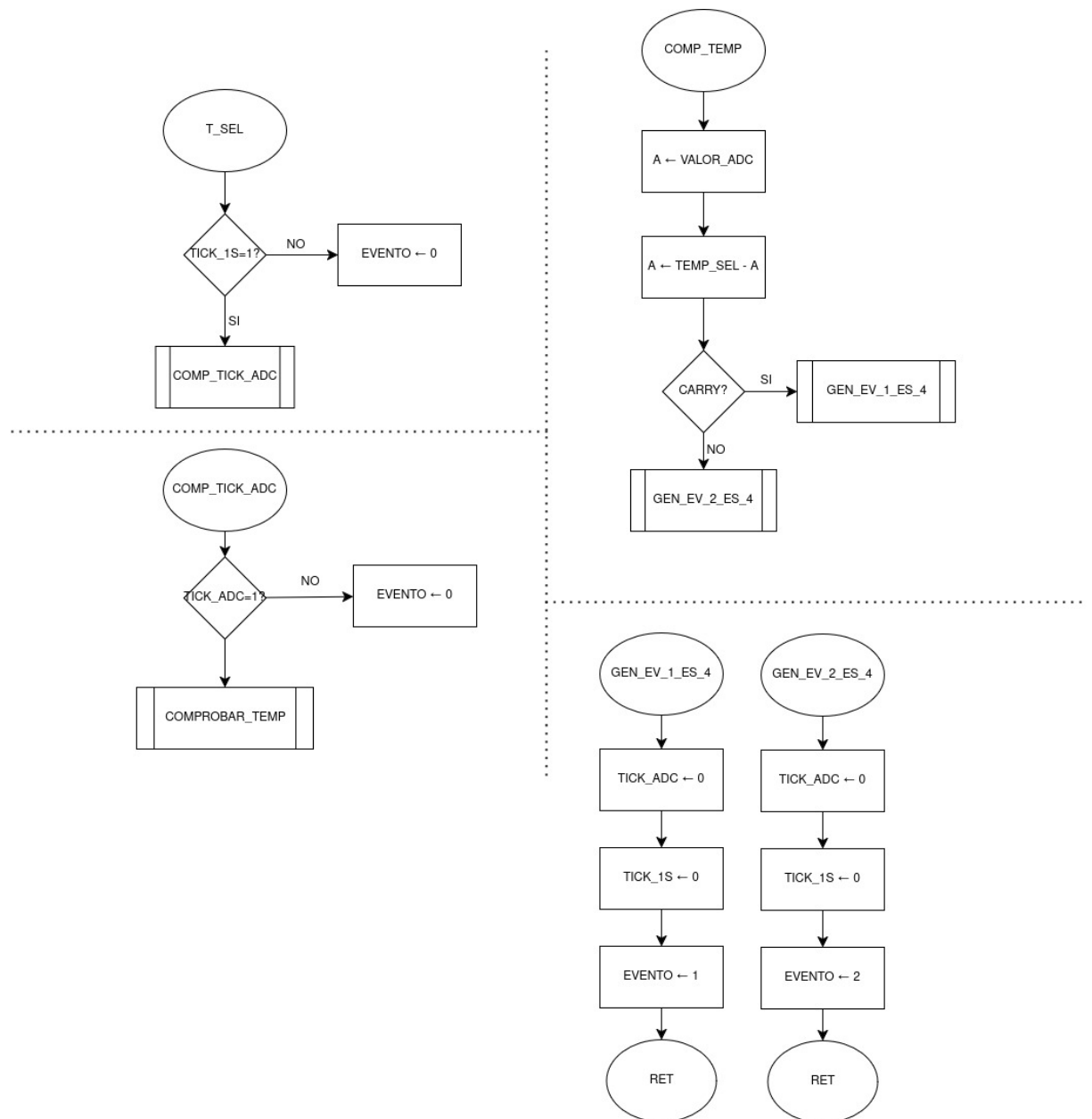
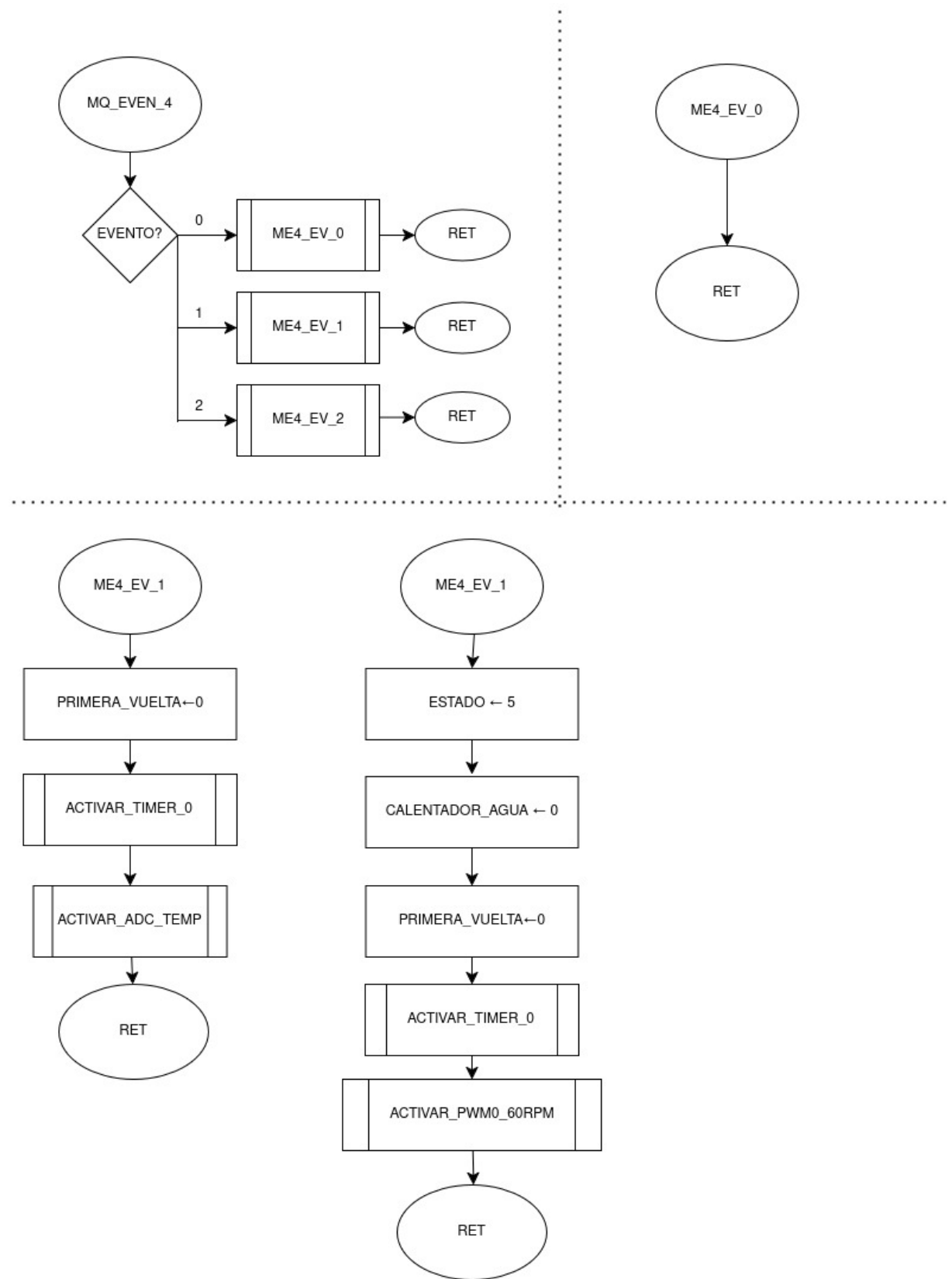


Figura 7: Diagrama lógica Generador de Eventos 4.

*Figura 8: Diagrama Máquina de Eventos 4.*

3. Cálculos y comentarios

3.1. Cálculo Sobrepeno

Para calcular el valor a partir del cual mediremos sobre peso debemos tener en cuenta lo siguiente. Nuestro V_{\min} es 0V y nuestro V_{\max} son 5V. Por tanto:

$$V_D = 2^n \frac{V_x - V_{\min}}{V_{\max} - V_{\min}} \quad (1)$$

De esta forma sabemos que nuestro V_x es 4,5V según lo especificado.

$$V_D = 2^8 \cdot \frac{4,5 - 0}{5 - 0} = 230,4 \approx 230_d \quad (2)$$

O lo que es lo mismo **0xE6** en Hexadecimal

3.2. Cálculo Temperaturas

Para el calculo de las temperaturas tenemos que 0°C son 0V y 100°C son 5V por ello para calcular las 3 temperaturas tenemos:

- 40°C son 2V
- 60°C son 3V
- 80°C son 4V

De esta manera podemos introducir en la ecuación:

$$V_D = 2^n \frac{V_x - V_{\min}}{V_{\max} - V_{\min}} \quad (3)$$

Por tanto:

- 40°C / 2V

$$V_D = 2^8 \cdot \frac{2 - 0}{5 - 0} = 102,4 \approx 102_d = 0xCD \quad (4)$$

- 60°C / 3V

$$V_D = 2^8 \cdot \frac{3 - 0}{5 - 0} = 153,6 \approx 154_d = 0x9A \quad (5)$$

- 80°C / 4V

$$V_D = 2^8 \cdot \frac{4 - 0}{5 - 0} = 204,8 \approx 205_d = 0x66 \quad (6)$$

3.3. Cálculo timer0 1s

Para calcular un segundo del timer usaremos el **Timer0 en modo 1**. Esto significa que el timer contará desde *0-65535d* y cuando llegue a este ultimo numero hará saltar una interrupción.

Hay que mencionar que contamos con un preescaler el cual reduce la frecuencia del tick del procesador.

Calculamos la frecuencia como:

$$f_{\text{tick}} = \frac{f_{\text{Osc}}}{\text{Prescaler}} = \frac{24 \cdot 10^6}{12} = 2 \cdot 10^6$$

$$T_{\text{tick}} = \frac{1}{f_{\text{tick}}} = \frac{1}{2 \cdot 10^6} = 0,5 \cdot 10^{-6}$$

Así el numero de ticks para 1s viene a ser:

$$n\text{Ticks} = \frac{1s}{0,5 \cdot 10^{-6} \frac{s}{\text{tick}}} = 2 \cdot 10^6 \text{ticks} = 1s$$

Sabemos que cada interrupcion cuenta 65536 ticks, de esta manera:

$$\text{numInterrupciones} = \frac{2 \cdot 10^6}{65536} = 30,5175$$

Por ello hemos de contar 30 Interrupciones y 0,5175. Esta vuelta NO-completa la calcularemos la primera y para ello iniciaremos el timer 0 con los valores que hagan contar solo 0,5175 de 65536.

$$0,5175 \cdot 65536 = 33920; 65535 - 33920 = 31615$$

Estrategia para contar 1s:

- 30 veces x 0-65535 Iniciamos TH0 a 0x00 y TL0 a 0x00
- 1 x vez 31615-65535 Iniciaremos TH0 a 0x7B y TL0 a 0x80

3.4. Cálculos PWM

Para calcular los valores a pasar al PWM0 debemos saber que la f_{osc} es 24MHz y que no tenemos Preescaler.

De esta manera sabemos que:

$$f_{PWM} = \frac{f_{osc}}{2 * (1 + PWMP) * 255} = \frac{24 * 10^6}{2 * (1 + 0) * 255} = 0,04706 * 10^6$$

PWMP es 0 ya que no usamos preescaler. Modo a 60RPM equivale a 90 % de la frecuencia, y otro modo a 600RPM a 10 %. Por otro lado, existe el modo Apagado en el cual el ciclo es del 100 %.

$$Valor = 255 * (1 - Ciclo)$$

- MODO 60RPM:

$$PWM0 = 255(1 - 0,9) = 25,5 \approx 26_d = 0x1A$$

- MODO 600RPM:

$$PWM0 = 255(1 - 0,1) = 229,5 \approx 230_d = 0xE6$$

- MODO 0RPM:

$$PWM0 = 255(1 - 1) = 0_d = 0x00$$

3.5. Comentarios sobre Acciones Destacables

3.5.1. Mapear minutos a Displays

Para mapear los minutos restantes a los displays hemos usado dos elementos principales del procesador

- DIV A,B

Para esto he aprovechado que la operacion de división, una de las mas costosas del juego de instrucciones, cuando se realiza guarda en **A el COCIENTE y en B el RESTO**. Esto es perfecto para despues realizar un MOV DECENAS, A y un MOV UNIDADES, B y tener un desglose perfecto del tiempo restante. En este caso se obtiene realizando la division de TIEMPO_RESTANTE y 0x0A Asi dividiendo entre 10 podemos separar esto de la manera que se quiere.

- 2 Tablas 0-9

Una vez extraidos las decenas y unidades con una tabla de 10 valores que en base al valor de la decena/undidad salta a la subrutina concreta de ese numero y lo mueve al display0/1 respectivamente.

3.6. Lógica de Cuenta de Tiempo

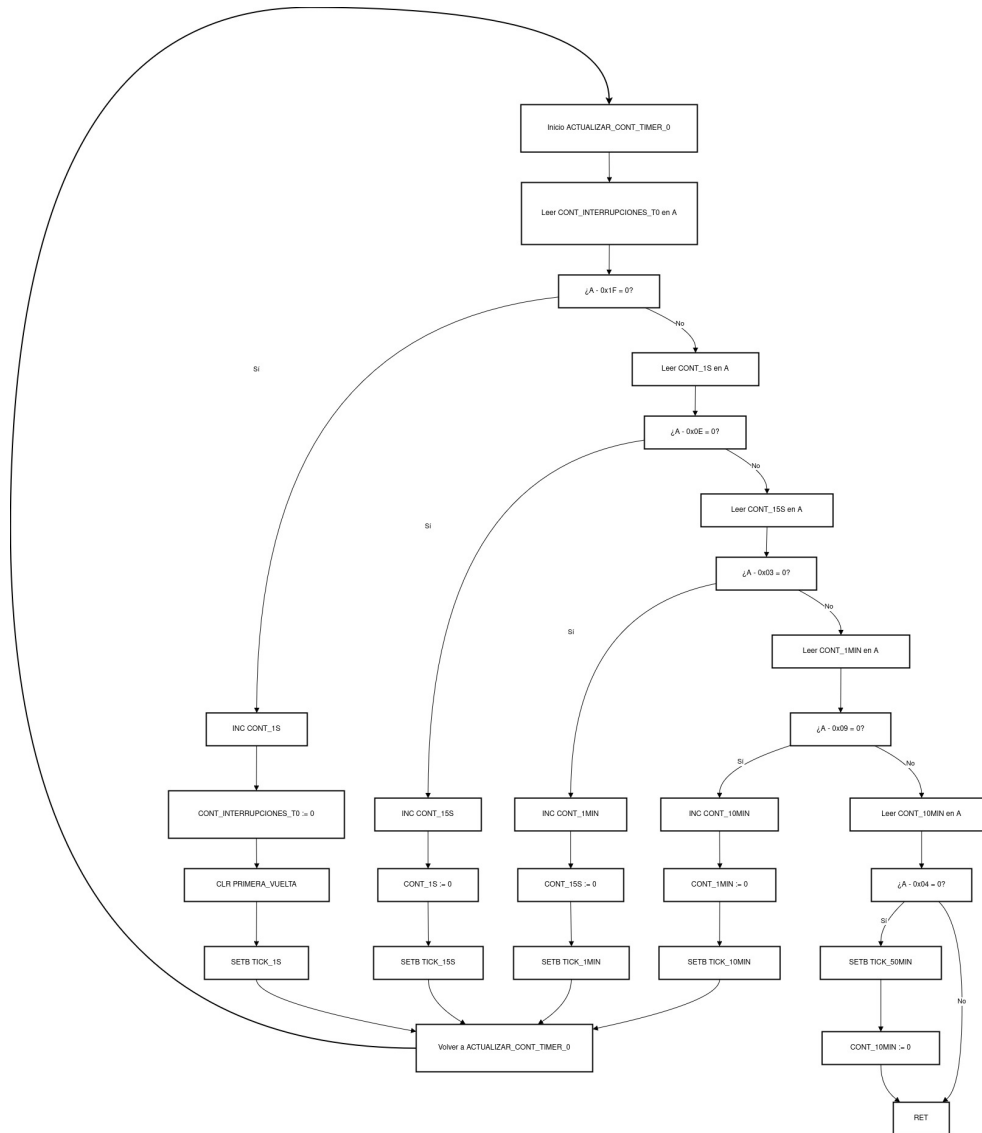


Figura 9: Diagrama Lógica Cuentas de Tiempo

Para contar el tiempo, o settear los ticks mejor dicho, en cada ocasion se comprueban todos los ticks recursivamente. Aprovechamos que 1s compone 15 veces 15s, 15s compone 4 veces 1 MIN, 1 MIN compone 10 veces 10 MIN y 10 MIN compone 5 veces a 50 MIN. Por ello esta forma recursiva es la que he elegido para comprobar el tiempo.

4. Código

```

1 ;=====VARIABLES=====
2
3 ESTADO EQU 0x20
4 EVENTO EQU 0x21
5 DISPLAY_1 EQU 0x25
6 DISPLAY_0 EQU 0x26
7
8 ;-----CUENTAS Y VALORES-----
9
10 TIEMPO_RESTANTE_MINS EQU 0x22
11 DECENAS EQU 0x28
12 UNIDADES EQU 0x29
13 TEMP_SEL EQU 0x2A
14
15 PRIMERA_VUELTA EQU 0x27.5
16 CONT_INTERRUPCIONES_T0 EQU 0x2B
17 CONT_1S EQU 0x2C
18 CONT_15S EQU 0x2D
19 CONT_1MIN EQU 0x2E
20 CONT_10MIN EQU 0x2F
21
22 VALOR_ADC EQU 0x30
23 ;-----TICKS-----
24
25 TICK_INT_0 EQU 0x23.0
26 TICK_ADC EQU 0x23.1
27 TICK_TIMER_0 EQU 0x23.3
28 TICK_PESADO_OK EQU 0x23.4
29 TICK_1S EQU 0x27.0
30 TICK_15S EQU 0x27.1
31 TICK_1MIN EQU 0x27.2
32 TICK_50MIN EQU 0x27.3
33 TICK_10MIN EQU 0x27.4
34
35 ;-----SENSORES Y ACTIVACIONES-----
36
37 BLOQUEO_PUERTA EQU P0.7
38 MOTOR_DESAGUE EQU P1.0
39 PUERTA_ABIERTA EQU P1.1
40 SENSOR_LLENADO EQU P1.2
41 SENSOR_VACIADO EQU P1.3
42 BOTON_INICIO EQU P1.4
43 BIT1_TEMP EQU P3.1
44 BIT0_TEMP EQU P3.0
45 VALV_ENTRADA_AGUA EQU P3.3
46 CALENTADOR_AGUA EQU P3.6
47 SENTIDO_GIRO_MOTOR EQU P2.7
48
49 ;-----REGISTROS IMPORTANTES-----
50
51 ADCON EQU 0xC5
52 ADCH EQU 0xC6
53 PWMP EQU 0xFE
54 PWM0 EQU 0xFC
55
56 ;=====INCIO=====
57
58 ORG 0x0000
59 AJMP INICIO
60 ORG 0x0003
61 AJMP INT0_ISR
62 ORG 0x000B
63 AJMP TIMER_0_ISR
64 ORG 0x0053
65 AJMP ADC_ISR

```

```

66 | ORG 0X007b
67 |
68 | INICIO:
69 |     ACALL INICIALIZACIONES
70 |
71 | MAIN:                ; LAZO PRINCIPAL
72 |     ACALL MAQ_ESTADOS
73 |     AJMP MAIN
74 |
75 | MAQ_ESTADOS:
76 |     MOV A, ESTADO
77 |     RL A
78 |     MOV DPTR, #LISTA_EST
79 |     JMP @A+DPTR
80 | LISTA_EST:
81 |     AJMP ESTADO_0 ; REPOSO
82 |     AJMP ESTADO_1 ; COMPROBACION SOBREPESO
83 |     AJMP ESTADO_2 ; PUERTA ABIERTA
84 |     AJMP ESTADO_3 ; LLENANDO
85 |     AJMP ESTADO_4 ; CALENTAR CUBA
86 |     AJMP ESTADO_5 ; LAVADO
87 |     AJMP ESTADO_6 ; VACIAR AGUA
88 |     AJMP ESTADO_7 ; CENTRIFUGAR
89 |     AJMP ESTADO_8 ; FIN
90 |
91 | ;=====INICIALIZACIONES=====
92 |
93 | INICIALIZACIONES:
94 |
95 |     MOV ESTADO, #0
96 |     MOV EVENTO, #0
97 |     MOV DISPLAY_1, #0x05
98 |     MOV DISPLAY_0, #0x00
99 |     MOV TIEMPO_RESTANTE_MINS, #0x3C ; INICIAMOS LOS MINUTOS RESTANTES DE LAVADO A 60h IREMOS RESTANDO DE UNO
100 |     EN UNO
101 |     MOV PWMP, #0x80
102 |     CLR IE.7
103 |     SETB P3.2          ; DESACTIVAMOS A NIVEL BAJO LA INTO
104 |     SETB IE.7          ; ACTIVAMOS EA* TODAS LAS INTERRUPCIONES
105 |     SETB IE.0          ; ACTIVAMOS LA INTO
106 |     CLR TICK_INT_0      ; TICK A NIVEL ALTO POR ELLO SE INICIA A 0
107 |     CLR TICK_ADC        ; TICK A NIVEL ALTO INICIAMOS A 0
108 |     CLR TICK_1S
109 |     CLR TICK_15S
110 |     CLR TICK_1MIN
111 |     CLR TICK_10MIN
112 |     CLR TICK_50MIN
113 |     CLR TICK_PESADO_OK
114 |     CLR PRIMERA_VUELTA
115 |
116 | ; ENTRADAS HACEMOS SETB Y ANTES DE LA SIMULACION MARCAMOS EL !! PIN !! AL ESTADO QUE NECESITEMOS INICIAL
117 | SETB P1.1
118 | SETB P1.2
119 | SETB P1.3
120 | SETB P1.4
121 | SETB P3.0
122 | SETB P3.1
123 |
124 | ; SALIDAS HACEMOS CLR PORQUE NO NOS ENCARGAREMOS MANUALMENTE DE HACER NADA ES EL PROPIO PROGRAMA EL QUE
125 |     LOS ACTIVA
126 | CLR P1.0
127 | CLR P3.3
128 | CLR P3.6
129 | ANL P0, #0x00          ; ASIGNAMOS P0 COMO OUTPUT
130 | ANL P2, #0x00          ; ASIGNAMOS P2 COMO OUTPUT
131 |
132 | CLR P1.5
133 | CLR P1.6

```

```

132     CLR P1.7
133     CLR P3.4
134     CLR P3.5
135     CLR P3.7
136
137     RET
138 ;=====
139
140 ;LLAMAMOS AL GENERADOR DE EVENTOS PARA QUE ASI NOS DIGA A CUAL DEBEMOS IR EN EL ESTADO DEL SISTEMA QUE NOS
    ENCONTRAMOS
141
142 ESTADO_0:
143     ACALL GEN_EV_0
144     ACALL MQ_EVEN_0      ;REPOSO
145     RET
146
147 ESTADO_1:
148     ACALL GEN_EV_1
149     ACALL MQ_EVEN_1      ;COMPROBANDO SP
150     RET
151
152 ESTADO_2:
153     ACALL GEN_EV_2
154     ACALL MQ_EVEN_2
155     RET
156
157 ESTADO_3:
158     ACALL GEN_EV_3
159     ACALL MQ_EVEN_3
160     RET
161
162 ESTADO_4:
163     ACALL GEN_EV_4
164     ACALL MQ_EVEN_4
165     RET
166
167 ESTADO_5:
168     ACALL GEN_EV_5
169     ACALL MQ_EVEN_5
170     RET
171
172 ESTADO_6:
173     ACALL GEN_EV_6
174     ACALL MQ_EVEN_6
175     RET
176
177 ESTADO_7:
178     ACALL GEN_EV_7
179     ACALL MQ_EVEN_7
180     RET
181
182 ESTADO_8:
183     ACALL GEN_EV_8
184     ACALL MQ_EVEN_8
185     RET
186
187 ;=====ESTADO 0 - REPOSO=====
188 MQ_EVEN_0:
189
190     MOV A, EVENTO
191     RL A
192     MOV DPTR, #LIST_EVEN_MQEV_0
193     JMP @A+DPTR
194 LIST_EVEN_MQEV_0:
195     AJMP ME0_EV_0 ; Evento 0
196     AJMP ME0_EV_1 ; Evento 1
197     AJMP ME0_EV_2 ; Evento 2
198 ME0_EV_0:

```

```

199 ORL PCON, #01H      ;PONEMOS EL IDLE BIT 0 DE PCON NO DEBEMOS ACCEDER EN FORMA DE BIT HACEMOS UN ORL
200 RET
201 ME0_EV_1:
202     ;---      ;ENCENDIO Y PUERTA ABIERTA
203     CLR IE.0
204     MOV DISPLAY_0, #77H ;A
205     MOV DISPLAY_1, #73H ;P
206     ACALL CARGAR_DISPLAY_0
207     ACALL CARGAR_DISPLAY_1
208     MOV ESTADO, #02H
209     RET
210 ME0_EV_2:
211     ;---      ;ENCENDIO Y PUERTA CERRADA
212     CLR IE.0
213     SETB IE.7
214     SETB IE.6
215     ACALL ACTIVAR_ADC_PESO
216     CLR P3.2
217     MOV ESTADO, #01H
218     RET
219
220 ;=====ESTADO 1 - COMPROBANDO SP=====
221 MQ_EVEN_1:
222     MOV A, EVENTO
223     RL A
224     MOV DPTR, #LIST_EVEN_MQEV_1
225     JMP @A+DPTR
226 LIST_EVEN_MQEV_1:
227     AJMP ME1_EV_0 ; Evento 0
228     AJMP ME1_EV_1 ; Se abre la puerta
229     AJMP ME1_EV_2 ; salta adc y sobre peso
230     AJMP ME1_EV_3 ; salta adc, peso ok y pl.4
231     AJMP ME1_EV_4 ; salta adc, peso ok y no pl.4
232 ME1_EV_0:
233     RET
234
235 ME1_EV_1:      ;PUERTA ABIERTA
236     ACALL APAGAR_ADC
237     MOV DISPLAY_0, #77H ;A
238     MOV DISPLAY_1, #73H ;P
239     MOV ESTADO, #02H
240     ACALL CARGAR_DISPLAY_0
241     ACALL CARGAR_DISPLAY_1
242     RET
243 ME1_EV_2:      ;HAY SOBREPESO
244     ACALL APAGAR_ADC
245     MOV DISPLAY_0, #73H ;P
246     MOV DISPLAY_1, #6DH ;S
247     ACALL CARGAR_DISPLAY_0
248     ACALL CARGAR_DISPLAY_1
249     RET
250 ME1_EV_3:      ;NO SOBREPESO Y ENCENDIDO
251     ACALL APAGAR_ADC
252     MOV DISPLAY_1, #78H ;t
253     SETB BLOQUEO_PUERTA
254     ACALL CARGAR_TEMP_DSP_0 ;Esta subrutina carga en DISPLAY_1 el valor de la temperatura actual
255     ACALL CARGAR_DISPLAY_0
256     ACALL CARGAR_DISPLAY_1
257     SETB VALV_ENTRADA_AGUA
258     MOV ESTADO, #03H
259     RET
260
261 ME1_EV_4:      ;NO SOBREPESO Y NO IR A LLENADO, ESPERAMOS POR EL BOTON_INICIO
262     ACALL APAGAR_ADC
263     MOV DISPLAY_1, #78H ;t
264     ACALL CARGAR_TEMP_DSP_0 ;Esta subrutina carga en DISPLAY_0 el valor de la temperatura actual
265     ACALL CARGAR_DISPLAY_0
266     ACALL CARGAR_DISPLAY_1

```

```

267     RET
268 ;=====ESTADO 2 - PUERTA ABIERTA=====
269
270 MQ_EVEN_2:
271     MOV A, EVENTO
272     RL A
273     MOV DPTR, #LIST_EVEN_MQEV_2
274     JMP @A+DPTR
275 LIST_EVEN_MQEV_2:
276     AJMP ME2_EV_0 ; Evento 0
277     AJMP ME2_EV_1 ; Descripción Maq. Event. 0. Evento 1
278 ME2_EV_0:
279     RET
280 ME2_EV_1: ;PUERTA CERRADA
281     MOV ESTADO, #01H
282     ACALL SET_TEMP_SELECT
283     ACALL ACTIVAR_ADC_PESO
284     SETB IE.7
285     SETB IE.6
286     CLR TICK_ADC
287     CLR TICK_PESADO_OK
288     RET
289
290 ;=====ESTADO 3 - LLENADO=====
291
292 MQ_EVEN_3:
293     MOV A, EVENTO
294     RL A
295     MOV DPTR, #LIST_EVEN_MQEV_3
296     JMP @A+DPTR
297 LIST_EVEN_MQEV_3:
298     AJMP ME3_EV_0 ; Evento 0
299     AJMP ME3_EV_1
300 ME3_EV_0:
301     RET
302 ME3_EV_1: ;CUBA LLENA
303     MOV ESTADO, #04H
304     SETB CALENTADOR_AGUA
305     ACALL ACTIVAR_ADC_TEMP
306     ACALL ACTIVAR_TIMER_0
307     CLR TICK_1S
308     CLR VALV_ENTRADA_AGUA
309     ACALL SET_TEMP_SELECT
310     RET
311
312 ;=====ESTADO 4 - CALENTAMIENTO CUBA=====
313
314 MQ_EVEN_4:
315     MOV A, EVENTO
316     RL A
317     MOV DPTR, #LIST_EVEN_MQEV_4
318     JMP @A+DPTR
319 LIST_EVEN_MQEV_4:
320     AJMP ME4_EV_0 ; Evento 0
321     AJMP ME4_EV_1
322     AJMP ME4_EV_2
323 ME4_EV_0:
324     RET
325 ME4_EV_1: ; NO CALIENTE
326     CLR PRIMERA_VUELTA
327     ACALL ACTIVAR_TIMER_0
328     ACALL ACTIVAR_ADC_TEMP
329     RET
330 ME4_EV_2: ; AGUA CALIENTE
331     MOV ESTADO, #05H
332     CLR CALENTADOR_AGUA
333     CLR PRIMERA_VUELTA
334     ACALL ACTIVAR_TIMER_0

```

```

335     ACALL ACTIVAR_PWM0_60RPM
336     RET
337
338 ;=====ESTADO 5 - LAVADO=====
339
340 MQ_EVENT_5:
341     MOV A, EVENTO
342     RL A
343     MOV DPTR, #LIST_EVENT_MQEV_5
344     JMP @A+DPTR
345 LIST_EVENT_MQEV_5:
346     AJMP ME5_EV_0 ; Evento 0
347     AJMP ME5_EV_1
348     AJMP ME5_EV_2
349     AJMP ME5_EV_3
350 ME5_EV_0:
351     RET
352 ME5_EV_1: ; PASA 15S
353     ACALL CAMBIAR_SENTIDO_GIRO
354     CLR PRIMERA_VUELTA
355     ACALL ACTIVAR_TIMER_0
356     MAPEAR_TIEMPO_D1_D0 ;ACTUALIZAMOS PARA PODER MODIFICAR EL PUERTO 2.7
357     ACALL CARGAR_DISPLAY_0
358     ACALL CARGAR_DISPLAY_1
359     RET
360 ME5_EV_2: ; PASAN 1 MIN
361     DEC TIEMPO_REstante_MINS
362     ACALL MAPEAR_TIEMPO_D1_D0 ;GUARDA EN DSP1 Y DSP0 LOS VALORES DEL TIEMPO
363     ACALL CARGAR_DISPLAY_0
364     ACALL CARGAR_DISPLAY_1
365     CLR PRIMERA_VUELTA
366     ACALL ACTIVAR_TIMER_0 ;VOLVEMOS A ACTIVAR LA CUENTA DE 15S
367     RET
368 ME5_EV_3: ; PASAN 50 MIN
369     DEC TIEMPO_REstante_MINS
370     ACALL MAPEAR_TIEMPO_D1_D0 ;GUARDA EN DSP1 Y DSP0 LOS VALORES DEL TIEMPO
371     ACALL CARGAR_DISPLAY_0
372     ACALL CARGAR_DISPLAY_1
373     ACALL APAGAR_TIMER
374     ACALL APAGAR_PWM0
375     MOV ESTADO, #06H
376     SETB MOTOR_DESAGUE
377     CLR PRIMERA_VUELTA
378     RET
379
380 ;=====ESTADO 6 - VACIAR AGUA=====
381
382 MQ_EVENT_6:
383     MOV A, EVENTO
384     RL A
385     MOV DPTR, #LIST_EVENT_MQEV_6
386     JMP @A+DPTR
387 LIST_EVENT_MQEV_6:
388     AJMP ME6_EV_0 ; Evento 0
389     AJMP ME6_EV_1
390 ME6_EV_0:
391     RET
392 ME6_EV_1: ; CUBA VACIA
393     ACALL ACTIVAR_PWM0_600RPM
394     ACALL ACTIVAR_TIMER_0
395     SETB SENTIDO_GIRO_MOTOR
396     MAPEAR_TIEMPO_D1_D0
397     ACALL CARGAR_DISPLAY_1
398     ACALL CARGAR_DISPLAY_0
399     MOV ESTADO, #07H
400     RET
401
402 ;=====ESTADO 7 - CENTRIFUGAR=====

```

```

403
404 MQ_EVEN_7:
405     MOV A, EVENTO
406     RL A
407     MOV DPTR, #LIST_EVEN_MQEV_7
408     JMP @A+DPTR
409 LIST_EVEN_MQEV_7:
410     AJMP ME7_EV_0 ; Evento 0
411     AJMP ME7_EV_1 ; PASAN LOS 10MIN ACABAMOS EL CENTRIF.
412     AJMP ME7_EV_2 ; ACTUALIZAR TIEMPO Y NO 10 MINS DE CENTRIF.
413 ME7_EV_0:
414     RET
415 ME7_EV_1: ; PASAN 10 MIN
416     MOV ESTADO, #08H
417     ACALL APAGAR_PWM0
418     CLR BLOQUEO_PUERTA
419     MOV DISPLAY_1, #0x38 ;L
420     MOV DISPLAY_0, #0x71 ;F
421     ACALL CARGAR_DISPLAY_0
422     ACALL CARGAR_DISPLAY_1
423     CLR MOTOR_DESAGUE
424     RET
425
426 ME7_EV_2: ; PASA 1 MIN
427     DEC TIEMPO_REstante_MINS
428     ACALL MAPEAR_TIEMPO_D1_D0 ;GUARDA EN DSP1 Y DSP0 LOS VALORES DEL TIEMPO
429     ACALL CARGAR_DISPLAY_0
430     ACALL CARGAR_DISPLAY_1
431     RET
432
433 ;=====ESTADO 8 - FIN=====
434
435 MQ_EVEN_8:
436     MOV A, EVENTO
437     RL A
438     MOV DPTR, #LIST_EVEN_MQEV_8
439     JMP @A+DPTR
440 LIST_EVEN_MQEV_8:
441     AJMP ME8_EV_0 ; Evento 0
442     AJMP ME8_EV_1
443 ME8_EV_0:
444     RET
445 ME8_EV_1: ; PUERTA ABIERTA
446     MOV DISPLAY_0, #77H ; A
447     MOV DISPLAY_1, #73H ; P
448     MOV ESTADO, #02H
449     ACALL CARGAR_DISPLAY_0
450     ACALL CARGAR_DISPLAY_1
451     RET
452
453 ;=====GENERADORES DE EVENTOS=====
454
455 ;-----0-----
456
457 GEN_EV_0:
458     ;CONDICIONES ESTADO 0
459     JNB P3.2, COMPROBAR_PUERTA ;SI NO ESTA EL BOTON NI NOS MOLESTAMOS EN VER EL RESTO DE CONDICIONES
460     MOV EVENTO, #00H ;ESTAMOS SEGURO EN ESTADO 0 SI NO ESTA P3.2
461     RET
462
463 COMPROBAR_PUERTA:
464     JB PUERTA_ABIERTA, GEN_EV_1_ES_0
465     AJMP GEN_EV_2_ES_0
466
467 GEN_EV_1_ES_0:
468     ANL PCON, #0xFE ;QUITAMOS EL IDLE MODIFICANDO EL BIT 0 DEL PCON
469     MOV EVENTO, #01H ;EN EL CASO DE QUE SI ESTE LA PUERTA ABIERTA EVENTO 1 ESTADO 0
470     RET

```

```

471
472 GEN_EV_2_ES_0:
473     ANL PCON, #0xFE           ;QUITAMOS EL IDLE MODIFICANDO EL BIT 0 DEL PCON
474     MOV EVENTO, #02H         ;EN EL CASO DE QUE SI ESTE LA PUERTA ABIERTA EVENTO 2 ESTADO 0
475     RET
476
477
478
479 ;-----1-----
480
481 GEN_EV_1:
482     ;CONDICIONES ESTADO 1
483     JB PUERTA_ABIERTA, GEN_EV_1_ES_1 ;SI PUERTA ABIERTA NOS VAMOS A PUERTA ABIERTA
484     JB TICK_PESADO_OK, COMPROBAR_BTN_INICIO ;SI HEMOS PESADO YA COMPROBAMOS SI HEMOS PULSADO INICIO
485     JB TICK_ADC, HAY_SP         ;SI HAY TICK DEL ADC VAMOS A COMPROBAR EL SP
486     MOV EVENTO, #00H
487     RET
488
489 COMPROBAR_BTN_INICIO:
490     JB BOTON_INICIO, GEN_EV_3_ES_1
491     MOV EVENTO, #00H
492     RET
493
494 GEN_EV_1_ES_1:           ;PUERTA ABIERTA
495     CLR TICK_ADC
496     CLR TICK_PESADO_OK
497     MOV EVENTO, #01H
498     RET
499 GEN_EV_2_ES_1:           ;SOBREPESO
500     CLR TICK_ADC
501     MOV EVENTO, #02H
502     RET
503 GEN_EV_3_ES_1:           ;PESO OK E INICIO AVANZAMOS DE ESTADO
504     CLR TICK_ADC
505     MOV EVENTO, #03H
506     RET
507 GEN_EV_4_ES_1:           ;PESO OK NO INICIO NOS QUEDAMOS ESPERANDO A QUE SE PULSE INICIO
508     CLR TICK_ADC
509     SETB TICK_PESADO_OK
510     MOV EVENTO, #04H
511     RET
512
513 HAY_SP:
514     MOV A, VALOR_ADC
515     SUBB A, #0xE6; <- VALOR SP EQUIVALE A 4.5 VOLTIOS 230d
516     JNC GEN_EV_2_ES_1 ;SI EL RESULTADO ES NEGATIVO (HAY CARRY) VALOR SP > ADCH PESO OK
517     JNB BOTON_INICIO, GEN_EV_4_ES_1 ;SI NO SI ESTA INICIO PULSADO PASAMOS AL EVENTO 4
518     AJMP GEN_EV_3_ES_1
519     RET
520
521 ;-----2-----
522
523 GEN_EV_2:
524     ;CONDICIONES ESTADO 0
525     JNB PUERTA_ABIERTA, GEN_EV_1_ES_2
526     MOV EVENTO, #00H
527     RET
528
529 GEN_EV_1_ES_2:
530     MOV EVENTO, #01H
531     RET
532
533
534 ;-----3-----
535
536 GEN_EV_3:
537     ;CONDICIONES ESTADO 3
538     JB SENSOR_LLENADO, GEN_EV_1_ES_3

```



```

539     MOV EVENTO, #00H
540     RET
541
542 GEN_EV_1_ES_3:
543     MOV EVENTO, #01H
544     RET
545
546 ;-----4-----
547
548 GEN_EV_4:
549     ;CONDICIONES ESTADO 44
550
551     JB P3.0, T_SEL                ; P3.1 = 0 Y P3.0 = 0 NI SIQUERA MEDIMOS DIRECTAMENTE A
552     JB P3.1, T_SEL                ; TEMPERATURA ALCANZADA
553     AJMP GEN_EV_2_ES_4
554     T_SEL:
555     JB TICK_1S, COMP_TICK_ADC
556     MOV EVENTO, #00H
557     RET
558
559 COMP_TICK_ADC:
560     JB TICK_ADC, COMPROBAR_TEMP
561     MOV EVENTO, #00H
562     RET
563
564 COMPROBAR_TEMP:
565     JB P3.1, COMP_T3_T2_T1
566     JB P3.0, COMP_T3_T2_T1
567     AJMP GEN_EV_2_ES_4            ;SI TEMP1 - 0 Y TEMP0 - 0 ENTONCES T0 Y COMO ESTE ESTA OK
568     RET
569
570     COMP_T3_T2_T1:                ;SOLO SET TEMP_SEL SI T1,T2,T3 YA QUE T0 USA AGUA TAL CUAL
571     MOV A, VALOR_ADC
572     SUBB A, TEMP_SEL              ; SUSTITUIR EL VALOR DE 00H POR TEMP MINA ALCANZAR
573     JC GEN_EV_1_ES_4
574     AJMP GEN_EV_2_ES_4
575     RET
576
577 GEN_EV_1_ES_4:
578     CLR TICK_ADC                  ; NO CALIENTE
579     CLR TICK_1S
580     MOV EVENTO, #01H
581     RET
582
583 GEN_EV_2_ES_4:
584     CLR TICK_ADC                  ; SI CALIENTE
585     CLR TICK_1S
586     MOV EVENTO, #02H
587     RET
588
589 ;-----5-----
590
591 GEN_EV_5:
592     ;CONDICIONES ESTADO 0
593     JB TICK_50MIN, GEN_EV_3_ES_5
594     JB TICK_1MIN, GEN_EV_2_ES_5
595     JB TICK_15S, GEN_EV_1_ES_5
596     MOV EVENTO, #00H
597     RET
598
599 GEN_EV_1_ES_5: ;15 SEG
600     CLR TICK_15S
601     MOV EVENTO, #01H
602     RET
603
604 GEN_EV_2_ES_5: ;1MIN
605     CLR TICK_1MIN
606     MOV EVENTO, #02H
607     RET

```

```
607 GEN_EV_3_ES_5:  ;50MIN
608     CLR TICK_50MIN
609     CLR TICK_10MIN
610     CLR TICK_1MIN
611     CLR TICK_15S
612     MOV EVENTO,#03H
613     RET
614
615 ;-----6-----
616
617 GEN_EV_6:
618     ;CONDICIONES ESTADO 0
619     JB SENSOR_VACIADO, GEN_EV_1_ES_6
620     MOV EVENTO, #00H
621     RET
622
623 GEN_EV_1_ES_6:
624     MOV EVENTO, #01H
625     RET
626 ;-----7-----
627 GEN_EV_7:
628     ;CONDICIONES ESTADO 7
629     JB TICK_10MIN, GEN_EV_1_ES_7
630     JB TICK_1MIN, GEN_EV_2_ES_7
631     MOV EVENTO, #00H
632     RET
633
634 GEN_EV_1_ES_7:
635     CLR TICK_10MIN
636     MOV EVENTO, #01H
637     RET
638
639 GEN_EV_2_ES_7:
640     CLR TICK_1MIN
641     MOV EVENTO, #02H
642     RET
643 ;-----8-----
644 GEN_EV_8:
645     ;CONDICIONES ESTADO 0
646     JB PUERTA_ABIERTA, GEN_EV_1_ES_8
647     MOV EVENTO, #00H
648     RET
649
650 GEN_EV_1_ES_8:
651     MOV EVENTO, #01H
652     RET
653
654 ;-----
655 ;=====ACCIONES=====
656
657 ACTIVAR_ADC_PESO:                ;SELECCIONAMOS EL CANAL 0 PARA PESO
658     ANL ADCON,#0xE8
659     ORL ADCON,#0x08
660     RET
661
662 ACTIVAR_ADC_TEMP:
663     SETB IE.6                    ;SELECCIONAMOS EL CANAL 1 PARA TEMP
664     ANL ADCON, #0xE6
665     ORL ADCON, #0x09
666     RET
667 APAGAR_ADC:
668     CLR IE.6
669     ANL ADCON, #0xE0
670     RET
671
672
673 ACTIVAR_TIMER_0:
674     JNB PRIMERA_VUELTA, ACTIVAR_TIMER_0_PRIMERA_VUELTA
```

```
675     AJMP ACTIVAR_TIMER_0_FULL
676     RET
677
678 ACTIVAR_TIMER_0_PRIMERA_VUELTA:
679     MOV TMOD, #0x01
680     MOV TH0, #0x7B
681     MOV TL0, #0x80
682     SETB IE.7
683     SETB IE.1
684     SETB PRIMERA_VUELTA
685     SETB TR0
686     RET
687
688 ACTIVAR_TIMER_0_FULL:
689     MOV TMOD, #0x01
690     MOV TH0, #0x00
691     MOV TL0, #0x00
692     SETB IE.7
693     SETB IE.1
694     SETB TR0
695     RET
696
697 ACTUALIZAR_CONT_TIMER_0:
698
699     MOV A, CONT_INTERRUPCIONES_T0
700     SUBB A, #0x1F
701     JZ PASAN_1SEG
702
703     MOV A, CONT_1S
704     SUBB A, #0x0E
705     JZ PASAN_15SEG
706
707     MOV A, CONT_15S
708     SUBB A, #0x03
709     JZ PASAN_1MIN
710
711     MOV A, CONT_1MIN
712     SUBB A, #0x09
713     JZ PASAN_10MIN
714
715     MOV A, CONT_10MIN
716     SUBB A, #0x04
717     JZ PASAN_50MIN
718
719
720     RET                                ;escapar del act tiempos
721
722 PASAN_1SEG:
723     INC CONT_1S
724     MOV CONT_INTERRUPCIONES_T0, #00H
725     CLR PRIMERA_VUELTA                ;CUANDO SE VUELVA A ACTIVAR T0 LO HARA DE FORMA
726     SETB TICK_1S
727     AJMP ACTUALIZAR_CONT_TIMER_0
728     RET
729
730 PASAN_15SEG:
731     INC CONT_15S
732     MOV CONT_1S, #00H
733     SETB TICK_15S
734     AJMP ACTUALIZAR_CONT_TIMER_0
735     RET
736
737 PASAN_1MIN:
738     INC CONT_1MIN
739     MOV CONT_15S, #00H
740     SETB TICK_1MIN
741     AJMP ACTUALIZAR_CONT_TIMER_0
742     RET
743
744 PASAN_10MIN:
745     INC CONT_10MIN
```

```
743     MOV CONT_1MIN, #00H
744     SETB TICK_10MIN
745     AJMP ACTUALIZAR_CONT_TIMER_0
746     RET
747 PASAN_50MIN:
748     SETB TICK_50MIN
749     MOV CONT_10MIN, #00H
750     RET
751
752
753
754     ;SI CONT_INT_TO = 31 ENTONCES tick_1s
755     ;si tick_1s inc cont 1s
756     ;si cont1_1s = 15 ENTONCES tick_15s
757     ;si tick 15s inc cont_15s
758     ;si cont_15s = 4 entonces tick_1min
759     ;si tick_1min inc cont_1min
760     ;si cont_1min = 10 ENTONCES tick_10min
761     ;si tick_10min inc cont_10min
762     ;si cont_10min = 5 ENTONCES tick_50min
763
764
765     RET
766 APAGAR_TIMER:
767     RET
768
769 ACTIVAR_PWM0_60RPM:
770     MOV PWM0, #0x1A
771     RET
772
773 ACTIVAR_PWM0_600RPM:
774     MOV PWM0, #0xE6
775     RET
776
777 APAGAR_PWM0:
778     MOV PWM0, 0x00
779     RET
780
781 CARGAR_DISPLAY_1:
782     MOV A, #0x80
783     ANL P0, A           ; 1000 0000 -> nos settea todo a 0 menos el .7
784     MOV A, DISPLAY_1
785     ORL P0, A           ; dsp1 siempre tiene un 0 en .7 por ello no modifica p0.7
786
787     RET
788
789 CARGAR_DISPLAY_0:
790     MOV A, #0x80
791     ANL P2, A           ; 1000 0000 -> nos settea todo a 0 menos el .7
792     MOV A, DISPLAY_0
793     ORL P2, A           ; dsp1 siempre tiene un 0 en .7 por ello no modifica p0.7
794     RET
795
796
797 SET_TEMP_SELECT:
798     JB P3.1, SET_T_ALTAS
799     AJMP SET_T_BAJAS
800     RET
801
802 SET_T_ALTAS:
803     JB P3.0, SET_T_T3
804     AJMP SET_T_T2
805     RET
806 SET_T_BAJAS:
807     JB P3.0, SET_T_T1
808     RET
809
810 SET_T_T3:
```

```
811         MOV TEMP_SEL, #0xCD
812         RET
813     SET_T_T2:
814         MOV TEMP_SEL, #0x9A
815         RET
816     SET_T_T1:
817         MOV TEMP_SEL, #0x66
818         RET
819
820 CARGAR_TEMP_DSP_0:
821     JNB BIT1_TEMP, COMPROBAR_B0T_B1_0
822     AJMP COMPROBAR_B0T_B1_1
823     RET
824
825     COMPROBAR_B0T_B1_0:
826         JB BIT0_TEMP, PINTAR_D0_T1
827         AJMP PINTAR_D0_T0
828         RET
829     COMPROBAR_B0T_B1_1:
830         JB BIT0_TEMP, PINTAR_D0_T3
831         AJMP PINTAR_D0_T2
832         RET
833
834     PINTAR_D0_T0:
835         MOV DISPLAY_0, #0x3F    ;0
836         RET
837     PINTAR_D0_T1:
838         MOV DISPLAY_0, #0x06    ;1
839         RET
840     PINTAR_D0_T2:
841         MOV DISPLAY_0, #0x5B    ;2
842         RET
843     PINTAR_D0_T3:
844         MOV DISPLAY_0, #0x4F    ;3
845         RET
846
847     MAPEAR_TIEMPO_D1_D0:        ;guardamos los valores de t restante por decenas y unidades en dos dir.
848         MOV A, TIEMPO_REstante_MINS
849         MOV B, #0AH
850         DIV AB                  ; APROVECHAMOS QUE ADCH/10 COCIENTE -> A (decenas) Y RESTO -> B (unidades)
851         MOV DECENAS, A
852         MOV UNIDADES, B
853
854         ACALL MAP_DEC_D1
855         ACALL MAP_UNI_D0
856
857     RET
858
859     MAP_DEC_D1:
860         MOV A, DECENAS
861         RL A
862         MOV DPTR, #LISTA_DECENAS
863         JMP @A+DPTR
864     LISTA_DECENAS:
865         AJMP CD0
866         AJMP CD1
867         AJMP CD2
868         AJMP CD3
869         AJMP CD4
870         AJMP CD5
871         AJMP CD6
872         AJMP CD7
873         AJMP CD8
874         AJMP CD9
875
876     CD0:
877         MOV DISPLAY_1, #0x3F
878         RET
```

```
879 | CD1:
880 |     MOV DISPLAY_1, #0x06
881 |     RET
882 | CD2:
883 |     MOV DISPLAY_1, #0x5B
884 |     RET
885 | CD3:
886 |     MOV DISPLAY_1, #0x4F
887 |     RET
888 | CD4:
889 |     MOV DISPLAY_1, #0x66
890 |     RET
891 | CD5:
892 |     MOV DISPLAY_1, #0x6D
893 |     RET
894 | CD6:
895 |     MOV DISPLAY_1, #0x7D
896 |     RET
897 | CD7:
898 |     MOV DISPLAY_1, #0x07
899 |     RET
900 | CD8:
901 |     MOV DISPLAY_1, #0x7F
902 |     RET
903 | CD9:
904 |     MOV DISPLAY_1, #0x6F
905 |     RET
906 |
907 | MAP_UNI_D0:
908 |     MOV A, UNIDADES
909 |     RL A
910 |     MOV DPTR, #LISTA_UNIDADES
911 |     JMP @A+DPTR
912 | LISTA_UNIDADES:
913 |     AJMP CU0
914 |     AJMP CU1
915 |     AJMP CU2
916 |     AJMP CU3
917 |     AJMP CU4
918 |     AJMP CU5
919 |     AJMP CU6
920 |     AJMP CU7
921 |     AJMP CU8
922 |     AJMP CU9
923 |
924 | CU0:
925 |     MOV DISPLAY_0, #0x3F
926 |     RET
927 | CU1:
928 |     MOV DISPLAY_0, #0x06
929 |     RET
930 | CU2:
931 |     MOV DISPLAY_0, #0x5B
932 |     RET
933 | CU3:
934 |     MOV DISPLAY_0, #0x4F
935 |     RET
936 | CU4:
937 |     MOV DISPLAY_0, #0x66
938 |     RET
939 | CU5:
940 |     MOV DISPLAY_0, #0x6D
941 |     RET
942 | CU6:
943 |     MOV DISPLAY_0, #0x7D
944 |     RET
945 | CU7:
946 |     MOV DISPLAY_0, #0x07
```

```
947     RET
948 CU8:
949     MOV DISPLAY_0, #0x7F
950     RET
951 CU9:
952     MOV DISPLAY_0, #0x6F
953     RET
954
955 CAMBIAR_SENTIDO_GIRO:
956     JB SENTIDO_GIRO_MOTOR, FALSE_SG      ;SI SENT 1 -> VAMOS A SET0
957     AJMP TRUE_SG                        ;SI SENT 0 -> SET1
958     RET
959     FALSE_SG:
960         CLR SENTIDO_GIRO_MOTOR
961         RET
962     TRUE_SG:
963         SETB SENTIDO_GIRO_MOTOR
964         RET
965 ;=====INTERRUPCIONES=====
966
967 INTO_ISR:
968     PUSH ACC
969     PUSH PSW
970     SETB TICK_INT_0
971     CLR IE.0
972     POP PSW
973     POP ACC
974     RETI
975
976 ADC_ISR:
977     PUSH ACC
978     PUSH PSW
979     SETB TICK_ADC
980     MOV VALOR_ADC, ADCH
981     ANL ADCON, #0xE1      ;PARAS EL ADC
982     POP PSW
983     POP ACC
984     RETI
985
986 TIMER_0_ISR:
987     PUSH ACC
988     PUSH PSW
989     INC CONT_INTERRUPCIONES_T0
990     ACALL ACTUALIZAR_CONT_TIMER_0
991     ACALL ACTIVAR_TIMER_0
992     POP PSW
993     POP ACC
994     RETI
995
996 ;=====FIN=====
997
998 END
```