

Arquitectura de Computadores (27698)

Práctica Final de Laboratorio
2024-2025

Departamento:
Tecnología Electrónica

Titulación:
Grado en Informática de Gestión y Sistemas de Información

2º Curso (1º Cuatrimestre)

Fecha: 16-12-2024

Fernández Molano, Iker
Rivera Urretxi, Aitzol

Contenido

Contenido.....	1
1 Introducción.....	2
2 Diagramas.....	5
2.1 Diagrama de Estados.....	5
2.2 Diagrama de Flujos	11
3 Cálculos y Comentarios.....	17
3.1 Timer 0.....	17
3.2 PWM.....	18
3.3 ADC y Sensor de distancia	19
4 Código	20

1 Introducción

Este informe trata sobre el Proyecto Final de la asignatura Arquitectura de Computadores, el cual consistió en diseñar y realizar el código de un lavadero de coches.

El objetivo de esta práctica final fue el desarrollo de software para microcontroladores. Por ello, se utilizó el microcontrolador 80C552 del fabricante Philips Semiconductors. Para el desarrollo del código se empleó el entorno de simulación Keil µVision2.

Para la realización de este proyecto, se utilizaron conceptos como el PWM (Pulse Width Modulation), las interrupciones, un sensor de distancia, el conversor analógico-digital (ADC), el Timer 0, entre otros.

En primer lugar, se realizó el diseño del lavadero. Para ello, se elaboró el diagrama de estados, eventos y acciones correspondiente, acompañado de los diagramas de flujo necesarios para comenzar a plantear el código. Además, durante esta fase inicial, se definieron constantes mediante el uso de directivas EQUs, lo que permitió comprender el dato que se estaba tratando y acceder de manera más sencilla a las diferentes direcciones asignadas.

Cuando se diseñó el diagrama de estados, se intentó dividir las fases del lavado (enjabonado, frotado, etc.) en el mayor número posible de estados, buscando convertir un problema grande en varios más pequeños. Por ello, se obtuvo un diagrama de estados con un total de 22 estados y un número similar de eventos y acciones.

Siguiendo esta misma lógica, al plantear el bucle principal y mientras se desarrollaba el diagrama de flujos correspondiente, se concluyó que sería más sencillo trabajar con el código utilizando un generador de eventos múltiple, tratando de manera independiente los eventos de cada estado.

Continuando con las inicializaciones, los puertos han sido configurados poniendo las entradas a 1 y las salidas a 0, para que, de este modo, no sea posible modificar los pines manualmente. Además, se han inicializado las partes alta y baja del Timer 0, así como su modo, el PWM0 y PWM1, las interrupciones, etc.

Durante el proceso de escritura del código, inicialmente no surgieron problemas. Resultó fácil implementar el Timer 0, gracias a que se había estudiado recientemente en clase y se comprendía claramente su funcionamiento.

La interrupción del Timer es una interrupción por overflow, lo que significa que se direcciona al servicio de interrupción cuando se sobrepasa la cantidad máxima de bits del

timer. El Timer 0 se configuró en el modo 1, es decir, en el modo de 16 bits o 65.536 huecos. Este se utilizó para contar 50 milisegundos; para ello, se llenaron previamente algunos huecos (los cálculos se describirán en el punto 3 del informe). Posteriormente, se obtuvo 1 segundo, ya que 1 segundo equivale a 20 veces 50 milisegundos. Usando este mismo concepto y partiendo del segundo obtenido, se calcularon tiempos de 4, 5, 30 y 60 segundos.

Para asegurar que el código estuviera correctamente organizado en memoria, se utilizaron las directivas ORG, permitiendo asignar direcciones específicas a los servicios de interrupción, como el correspondiente al Timer 0 y al ADC. Este uso garantizó que las rutinas se encontraran en ubicaciones adecuadas y que el mapa de memoria estuviera optimizado según las especificaciones del datasheet del microcontrolador. Como ejemplo, la ISR del Timer 0, que se localiza en la dirección 0x000B; por medio de la directiva ORG 0x000B, la cual está sucedida por las instrucciones que conforman la atención a la interrupción, se trata correctamente la interrupción.

El gran reto al que se enfrentó el proyecto fue en la fase 2 del lavado, es decir, en el frotado de rodillos. Esta fase requería comprender tanto el funcionamiento del PWM como el del conversor analógico-digital (ADC).

Por un lado, el PWM se utilizó para controlar la potencia de giro de los rodillos, tanto horizontales como verticales, del lavadero, así como la potencia de los ventiladores en la última fase, el secado. Mediante el Duty-Cycle proporcionado en el enunciado y los cálculos que se explicarán posteriormente, se consiguió un valor digital correspondiente a la potencia solicitada.

Por otro lado, para controlar la colocación de los rodillos verticales y horizontales, fue imprescindible el uso del ADC para tratar los datos proporcionados por el sensor de distancia. Los datos suministrados por el sensor eran de carácter analógico, por lo que el ADC se encargó de convertirlos a digital para poder tratarlos. La interrupción del ADC se activaba una vez que se completaba la conversión del valor analógico, guardando el valor digital en el registro ADCH, ya que en este caso no se utilizó la parte baja del ADC.

En el tramo final del lavado, no se encontraron más problemas, ya que, aunque se siguiesen tratando el ADC y el PWM, pues se traía una experiencia previa, gracias a la

fase de frotado anteriormente explicada. De igual manera, el fin del lavado, donde se espera que el coche abandone la plataforma, no ha creado grandes problemas.

En los apartados que suceden a este, se detallará en el diagrama de estados y los diagramas de flujo. Así como, en los cálculos que han sido necesarios durante la implementación del software. Y, finalmente, se mostrará el código resultante.

2 Diagramas

2.1 Diagrama de Estados

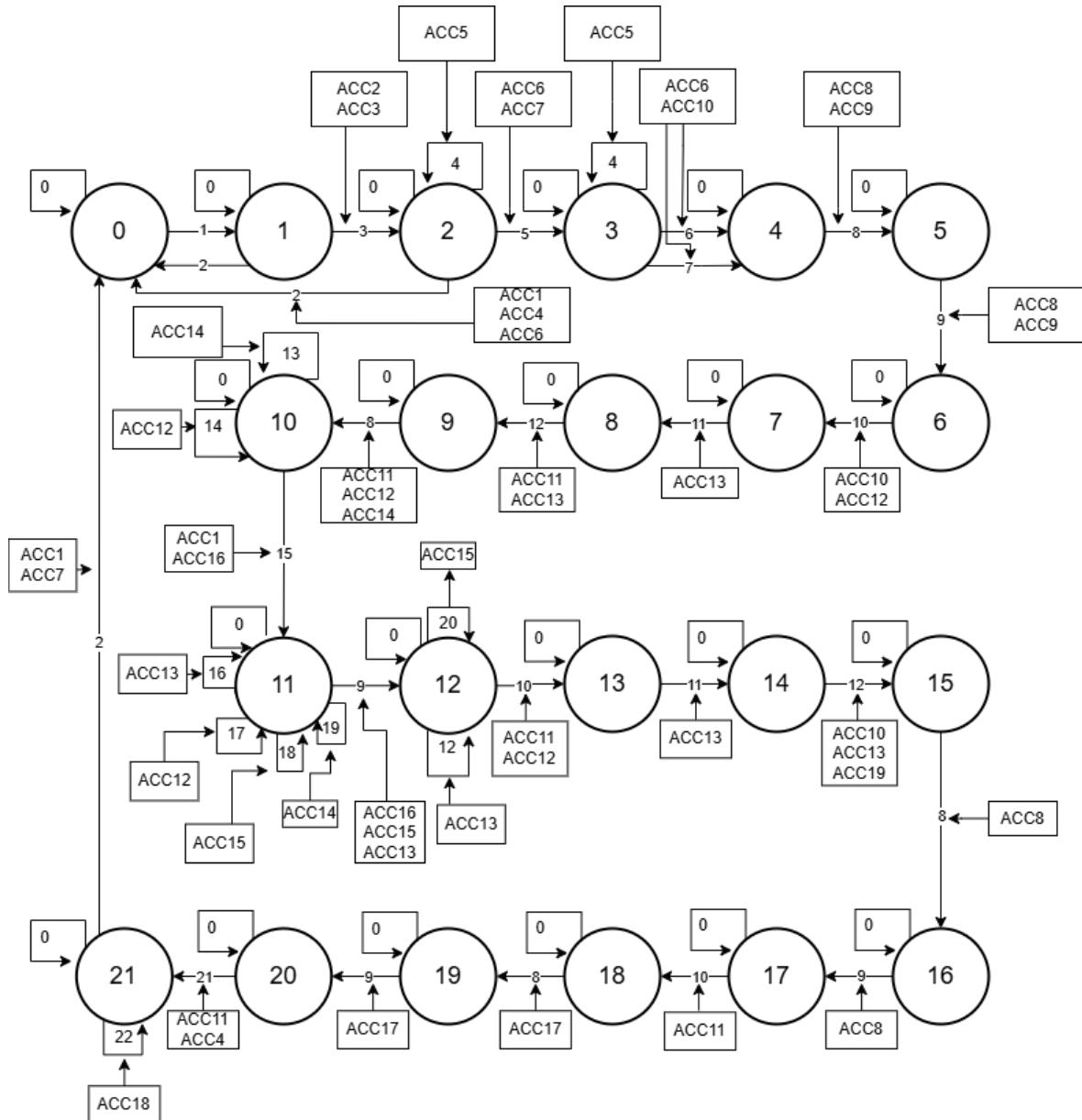


Figura 2.1.1.: Diagrama de estados

Nº DE ESTADO	NOMBRE	DESCRIPCIÓN
0	ESPERA_COCHE	El puente de lavado se encuentra esperando a un nuevo coche.
1	NUEVO_COCHE	Ha llegado un nuevo coche al autolavado, si se retira, volveremos al estado 0.
2	ESPERA_FICHA	Una vez pasados 5 segundos, el coche está correctamente colocado. Se espera la ficha, si en 30 segundos no se ha insertado, se activa la alarma.
3	INSERTADA_FICHA	Se ha insertado la ficha, encendiendo el led del modo seleccionado. Se espera a que se pulse el botón de inicio, si no se pulsa en 30 segundos se activa la alarma, después de 60 segundos iniciará el proceso automáticamente.
4	COLOCADO_COCHE1	Han pasado 60 segundos o se ha pulsado el botón de inicio. El puente se mueve hasta detectar el vehículo.
5	ENJABONADO	Una vez detectado el vehículo, mientras el puente se mueve de delante hacia atrás, se estará rociando agua y jabón. Esto se hará hasta que se deje de detectar el coche.
6	COLOCADO_COCHE2	Se ha dejado de detectar el coche, las electroválvulas de agua y jabón se han apagado, pero el puente sigue en movimiento, hasta alcanzar la velocidad de seguridad de 40cm.
7	ROD_HACIA_DENTRO_DETRES	Se ha alcanzado la distancia de seguridad, el puente no se mueve. Comienza el baile de los rodillos verticales en la parte trasera del coche, estos comienzan a moverse hacia dentro, hasta llegar al centro.
8	ROD_HACIA_FUERA_DETRES	Una vez llegados los rodillos al centro, comienzan a salir hacia fuera, hasta llegar al borde y terminando el baile.
9	COL_PUENTE_SOBRE_COCHE	Una vez terminado el baile, el puente se mueve de atrás hacia delante, hasta detectar el coche.
10	ROD_VDENTRO_HABAJO	Una vez detectado el coche, el puente se para. Los rodillos verticales apuntan hacia dentro y los horizontales hacia abajo hasta quedar ambos a 40cm de distancia del vehículo.
11	ROD_HV_DIST_SEGURIDAD	Cuando los rodillos se encuentran a una distancia de 40cm, el puente empieza a moverse de atrás hacia delante. Los rodillos, girando, siempre mantiene un rango de distancia entre los 37.5 y 42.5 cm de distancia del coche, acercándose o alejándose si es

		necesario. Esto se ocurre hasta dejar de detectar el coche.
12	ROD_HVPOSORIG	Se ha dejado de detectar el coche, el puente sigue avanzando a la distancia de seguridad de 40cm. Mientras tanto, el rodillo vertical sube a su posición original y los verticales hacia su posición en el borde, para prepararse para el segundo baile.
13	ROD_HACIA_DENTRO_DELANTE	Una vez lograda la posición de seguridad y que todos los rodillos se encuentren en su posición original, los rodillos verticales, apuntan hacia dentro, hasta llegar al centro.
14	ROD_HACIA_FUERA_DELANTE	Una vez llegado al centro, los rodillos verticales empiezan a salir hacia fuera, hasta llegar a la posición original, el borde.
15	COLOCADO_COCHE3	Una vez que los rodillos verticales están colocados en el borde, el puente comienza nuevamente a moverse de delante hacia atrás, hasta detectar el vehículo.
16	ACLARADO	Una vez detectado el vehículo, activa las electroválvulas de agua, para así aclarar el coche, hasta que deja de detectar el coche.
17	FIN_ACLARADO	Una vez se deja de detectar el coche, se apagan las electroválvulas de agua. El puente sigue moviéndose, hasta alcanzar la distancia de seguridad nuevamente.
18	COLOCADO_COCHE4	Una vez alcanzada la distancia de seguridad, el movimiento del puente cambia de sentido de atrás hacia delante.
19	SECADO	Una vez detectado el coche, activa los ventiladores, comenzando con el secado, hasta dejar de detectar el coche.
20	FIN_SECADO	Tras dejar de detectar el coche, se desactivan los ventiladores. El puente avanza hasta llegar a su posición inicial.
21	FINAL_LAVADO	Con el puente ya colocado en su posición original, el semáforo rojo se apaga, indicando que se ha acabado con el lavado. El semáforo verde comenzará a encenderse y apagarse cada segundo. Cuando el cliente retire el coche, se volverá al estado inicial, apagando el led del modo y encendiendo el semáforo verde.

Nº DE EVENTO	BIT	DESCRIPCIÓN
0	Null	Evento nulo.
1	S_PLAT = 1	Un coche se coloca en la plataforma.
2	S_PLAT = 0	La plataforma se encuentra sin coche.
3	TICK_5 = 1	Han pasado 5 segundos desde que se ha activado el timer.
4	TICK_30 = 1	Han pasado 30 segundos desde que se ha activado el timer.
5	S_FICHA = 1	Se ha insertado una ficha.
6	TICK_60 = 1	Han pasado 60 segundos desde que se ha activado el timer.
7	P_START = 1	Se ha pulsado el botón START, para empezar el lavado.
8	S_CAR = 1	El coche se encuentra bajo el puente y es detectado.
9	S_CAR = 0	El coche no se encuentra bajo el puente, por lo que no se detecta
10	TICK_4 = 1	Han pasado 4 segundos desde que se ha activado el timer.
11	FC_RV_CENTRO = 1	El sensor de fin de carrera de los rodillos verticales se encuentra en su posición más interna al puente.
12	FC_RV_BORDE = 1	El sensor de fin de carrera de los rodillos verticales se encuentra en su posición más exterior.
13	DIST_RH_40 = 1	El rodillo horizontal se encuentra a 40cm del vehículo.
14	DIST_RV_40 = 1	Los rodillos verticales se encuentran a 40cm del vehículo.
15	DIST_RH_40 = 1 & DIST_RV_40 = 1	Los rodillos verticales y el horizontal se encuentran a 40cm del vehículo.
16	DIST_RH < 37.5cm	El rodillo horizontal está a menos de 37.5cm de distancia del vehículo.
17	DIST_RH > 42.5cm	El rodillo horizontal está a más de 42.5cm de distancia del vehículo
18	DIST_RV < 37.5cm	Los rodillos verticales están a menos de 37.5cm de distancia del vehículo.
19	DIST_RV > 42.5cm	Los rodillos verticales están a más de 42.5cm de distancia del vehículo.
20	FC_RH_TOP = 1	El sensor de fin de carrera del rodillo horizontal está en su parte más alta.
21	BPOS_START = 1	Fin de carrera, posición de reposo del puente de lavado.
22	TICK_1 = 1	Ha pasado 1 segundo desde que se ha activado el timer.

Nº DE ACCIÓN	NOMBRE DE ACCIÓN	BIT	DESCRIPCIÓN
ACC1	-	SEM_VERDE = 1	Se enciende el semáforo verde.
ACC2	-	SEM_VERDE = 0	Se apaga el semáforo verde.
ACC3	-	SEM_ROJO = 1	Se enciende el semáforo rojo.
ACC4	-	SEM_ROJO = 0	Se apaga el semáforo rojo.
ACC5	-	AL_START = 1	Se enciende la alarma.
ACC6	-	AL_START = 0	Se apaga la alarma.
ACC7	LED_MODO_ ONOFF	Si LED_LNOR = 0 & LED_LINT = 0 -> LED_LNOR/LINT = 1 Si LED_LNOR/LINT=1 -> LED_LNOR/LINT = 0	Si el led de modo está apagado, activa el led correspondiente al modo. Si uno de los leds de modo está activado, lo apaga.
ACC8	EV_AGUA_ ONOFF	Si EV_AGUA= 1 -> EV_AGUA = 0 Si EV_AGUA = 0 -> EV_AGUA = 1	Si la electroválvula de agua está encendida se apaga. Si está apagada, se enciende.
ACC9	EV_JABON_ ONOFF	Si EV_JABON = 1 -> EV_JABON = 0 Si EV_JABON = 0 -> EV_JABON = 1	Si la electroválvula de jabón está encendida se apaga. Si está apagada, se enciende.
ACC10	MOV_FRONT_ TO_BACK_ ONOFF	Si BMOV_FRONT = 1 -> BMOV_FRONT = 0 Si BMOV_FRONT = 0 -> BMOV_BACK = 0 & BMOV_FRONT = 1	Si el puente está yendo de delante hacia atrás, lo para. Si está yendo de atrás hacia delante o está quieto, cambia el sentido o enciende el movimiento del puente de delante hacia atrás.
ACC11	MOV_BACK_TO _FRONT_ONOFF	Si BMOV_BACK = 1 -> BMOV_BACK = 0 Si BMOV_BACK = 0 -> BMOV_FRONT = 0 & BMOV_BACK = 1	Si el puente está yendo de atrás hacia delante, lo para. Si está yendo de delante hacia atrás o está quieto, cambia el sentido o enciende el movimiento del puente de atrás hacia delante.
ACC12	ROD_VERTI _DENTRO_ ONOFF	Si RV_OUTTO = 1 -> RV_OUTTO = 0 Si RV_OUTTO = 0 -> RV_INTO = 0 & RV_OUTTO = 1	Si los rodillos verticales no están yendo hacia fuera, empiezan a ir hacia fuera, apagando el movimiento hacia dentro por si estuviesen haciendo ese movimiento. Si estaban yendo hacia dentro, deja de ir hacia dentro.
ACC13	ROD_VERTI _FUERA_ONOFF	Si RV_INTO = 1 -> RV_INTO = 0 Si RV_INTO = 0 -> RV_OUTTO= 0 & RV_INTO = 1	Si los rodillos verticales no están yendo hacia fuera, empiezan a ir hacia fuera, apagando el movimiento hacia dentro por si estuviesen haciendo ese movimiento. Si estaban yendo hacia

			fuera, deja de ir hacia fuera.
ACC14	ROD_HOR_ABAJO_ONOFF	Si RH_DWNTO = 1 -> RH_DWNTO= 0 Si RH_DWNTO = 0 -> RH_UPTO = 0 & RH_DWNTO = 1	Si el rodillo horizontal no está yendo hacia abajo, empieza a ir hacia abajo, apagando el movimiento hacia arriba por si estuviese haciendo ese movimiento. Si estaba yendo hacia abajo, deja de ir hacia abajo.
ACC15	ROD_HOR_ARRIBA_ONOFF	Si RH_UPTO = 1 -> RH_UPTO = 0 Si RH_UPTO = 0 -> RH_DWNTO = 0 & RH_UPTO = 1	Si el rodillo horizontal no está yendo hacia arriba, empieza a ir hacia arriba, apagando el movimiento hacia abajo por si estuviese haciendo ese movimiento. Si estaba yendo hacia arriba, deja de ir hacia arriba.
ACC16	OFF_TODOS_MOV_ROD_HV	RV_INTO = 0 RV_OUTTO = 0 RH_DWNTO = 0 RH_UPTO = 0	Desactiva todos los movimientos de los rodillos, los deja quietos, aunque no afecta en la velocidad de giro.
ACC17	VENT_ONOFF	Si PWM1 = 0xFF -> PWM1 = 0x80/0x26 Si PWM= 0x80/0x26 -> PWM1 = 0xFF	Si los ventiladores están encendidos, los apaga. Si están apagados, los enciende con la potencia correspondiente a su modo.
ACC18	SEM_VERDE_ONOFF	Si SEM_VERDE = 1 -> SEM_VERDE = 0 Si SEM_VERDE = 0 -> SEM_VERDE = 1	Si el semáforo verde se encuentra encendido, lo apaga. Si se encuentra apagado, lo enciende.
ACC19	-	PWM0 = 0xFF	Apaga la rotación de los rodillos.

2.2 Diagrama de Flujos



Figura 2.2.1: Diagrama de flujos principal

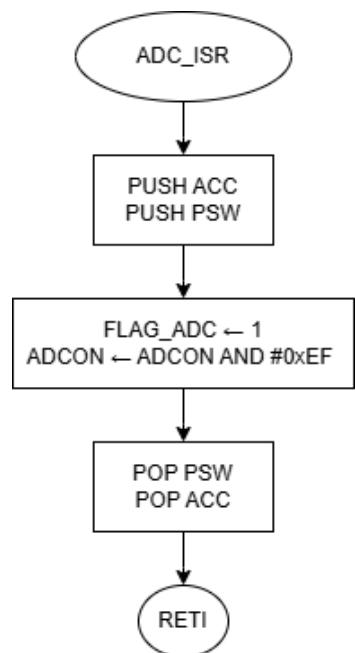


Figura 2.2.2: Diagrama de flujos ISR del ADC

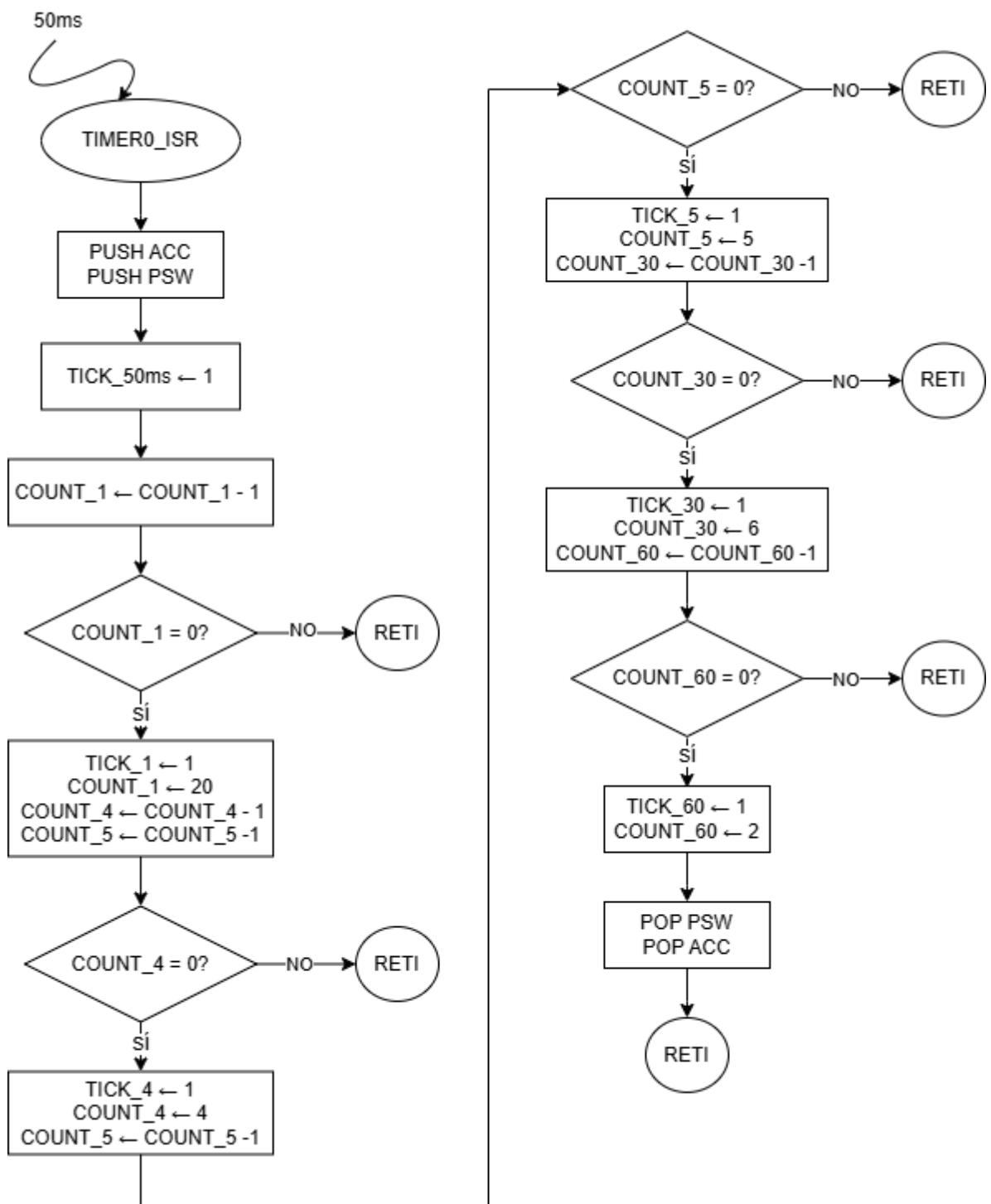


Figura 2.2.3: Diagrama de flujos ISR del Timer 0

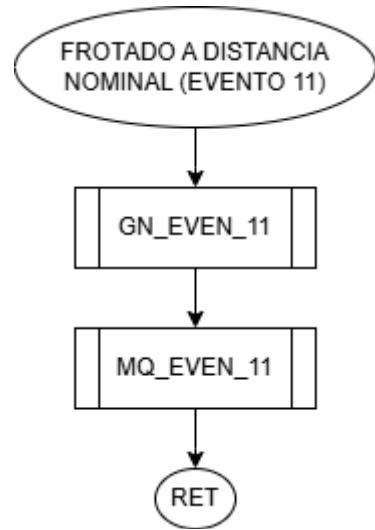


Figura 2.2.4: Diagrama de flujos del estado 11, frotado a distancia nominal

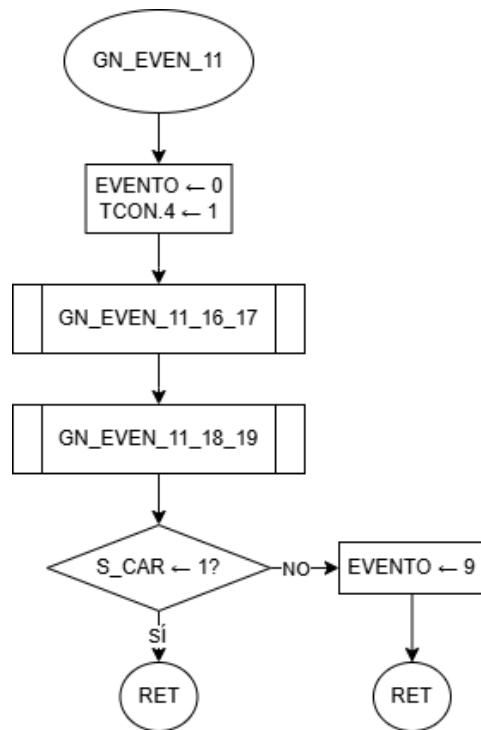


Figura 2.2.4.1: Diagrama de flujos del generador de eventos del estado 11

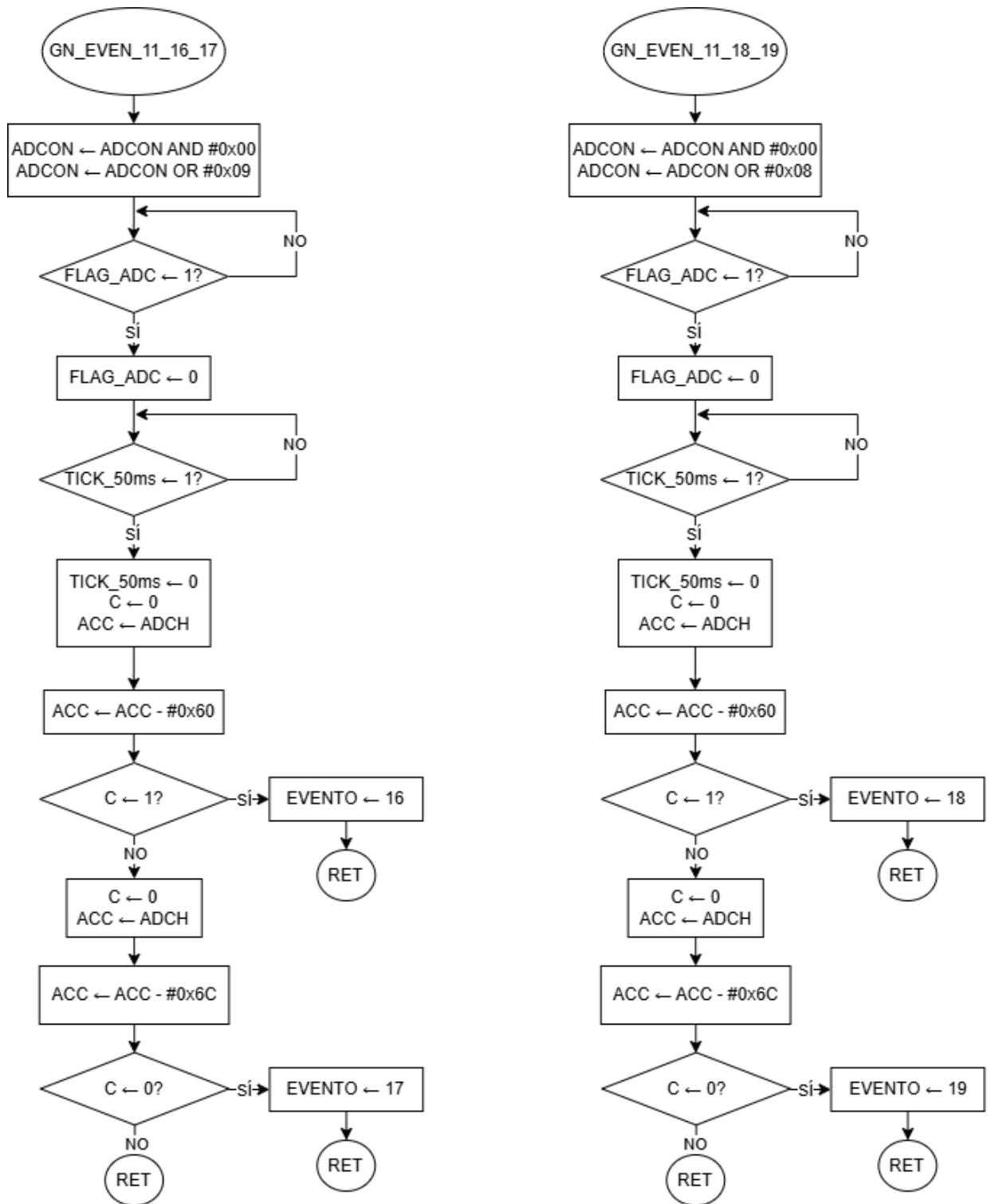


Figura 2.2.4.2: Diagrama de flujos del generador de eventos del estado 11, eventos 16, 17, 18 y 19.

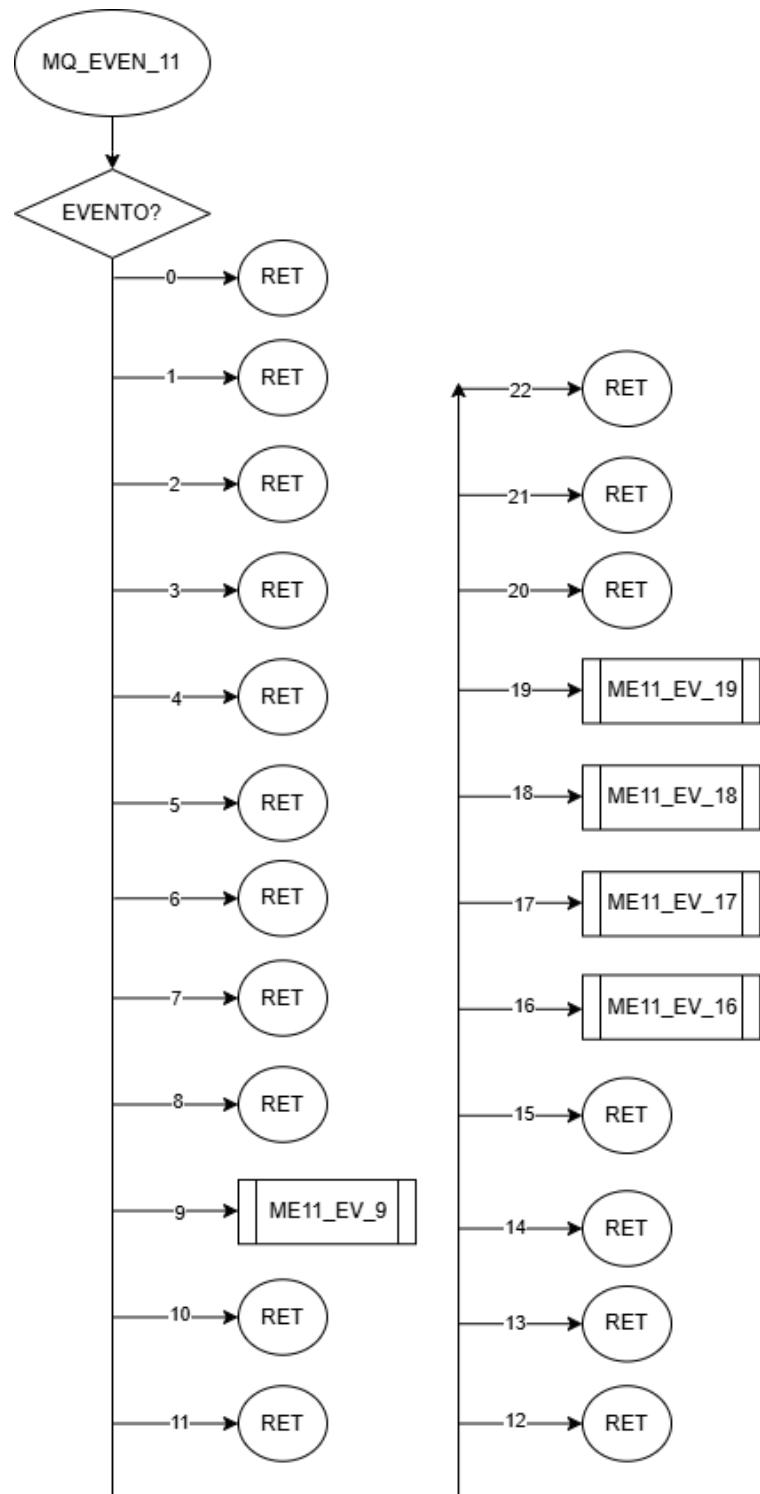


Figura 2.2.4.3: Diagrama de flujos de la máquina de eventos del estado 11

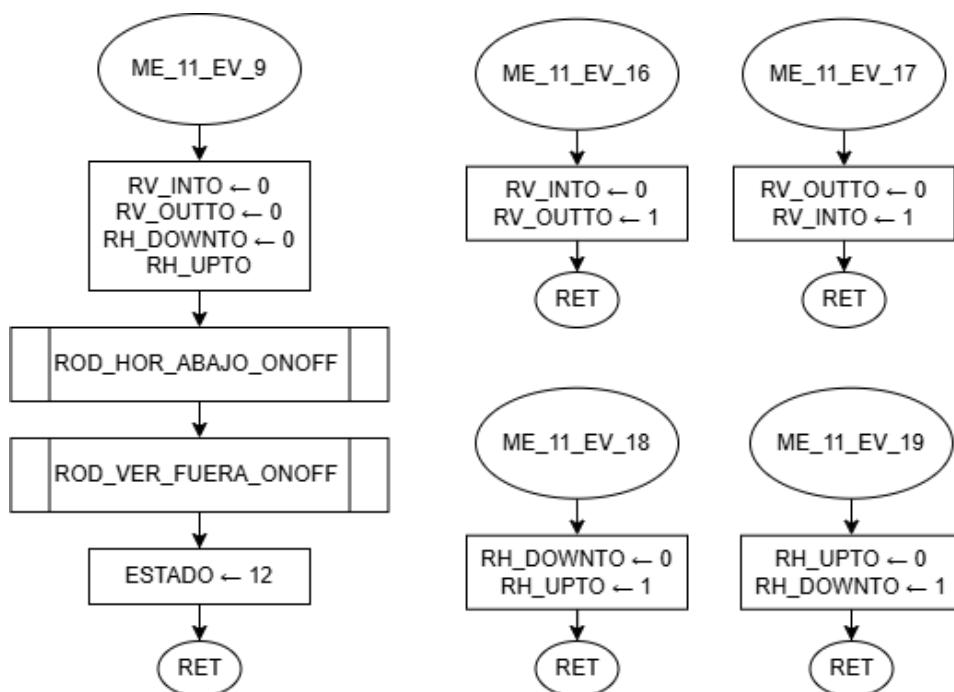


Figura 2.2.4.4: Diagrama de flujos de las máquinas de eventos 9, 16, 17, 18 y 19

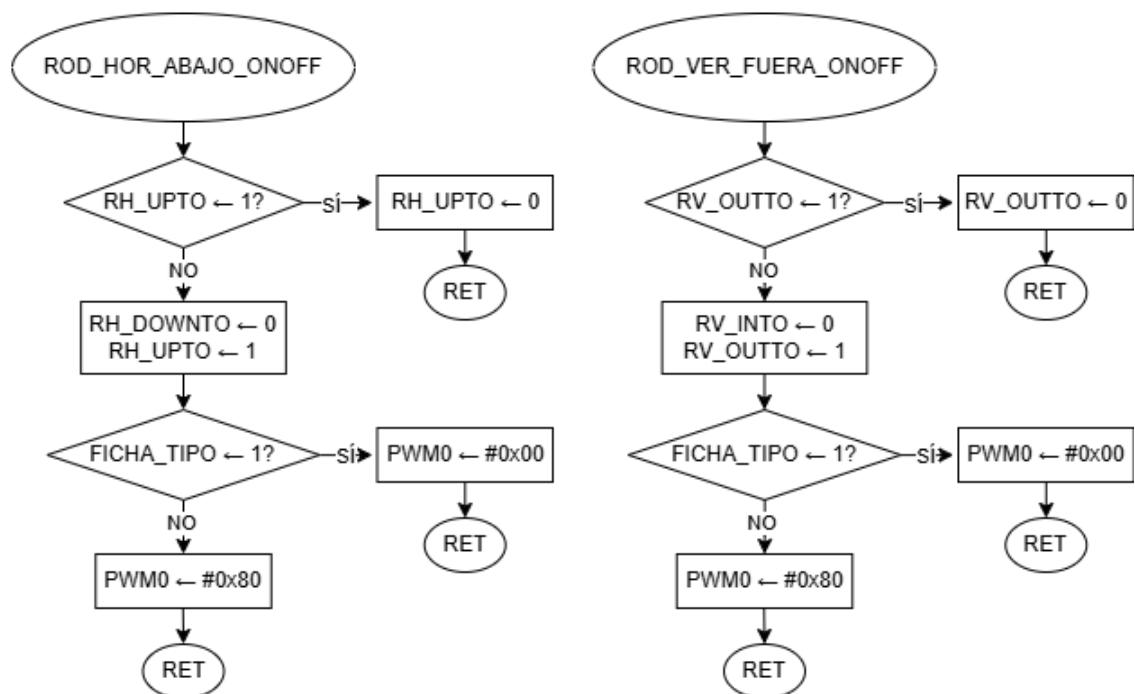


Figura 2.2.4.5: Diagrama de flujos de las acciones 13 y 14

3 Cálculos y Comentarios

3.1 Timer 0

El Timer 0, utilizado para hacer determinados conteos dentro del programa, está configurado en modo 1 (Figura 3.1.), lo que significa que tiene 16 bits o lo que es lo mismo 65.536 huecos. Estos han sido los cálculos necesarios para su implementación.

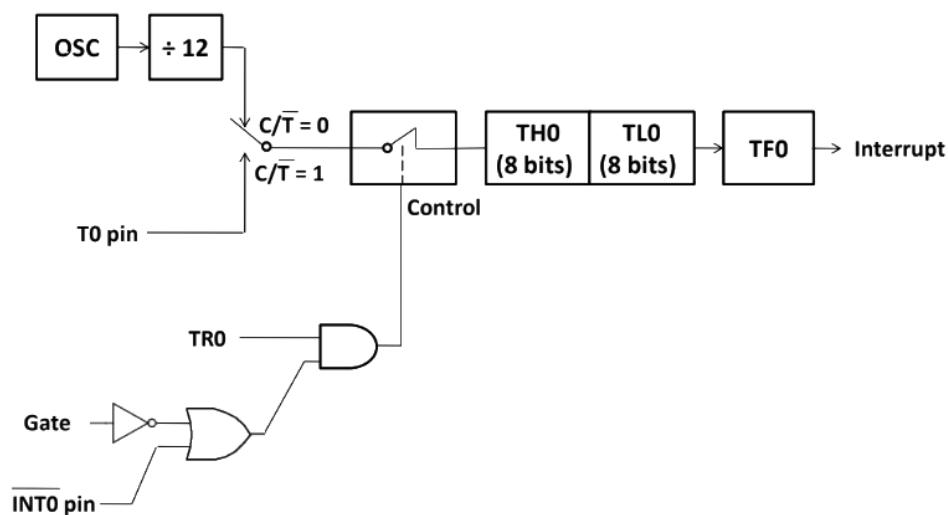


Figura 3.1.1.: Esquema del Timer 0 modo 1

Primero, se ha utilizado la frecuencia de oscilación (fosc=12MHz), la cual se proporciona en el enunciado, dividida entre 12:

$$f_{timer} = \frac{12MHz}{12} = 1MHz$$

Una vez se conoce este valor, como queremos conseguir que el timer cuente 50ms, se multiplica el valor obtenido por $5 \cdot 10^{-2}$ para conseguir el número de ciclos que van a ser utilizados por el timer:

$$n^o \text{ ciclos} = 1 \cdot 10^6 \cdot 5 \cdot 10^{-2} = 50.000 \text{ ciclos}$$

Como ya se conoce el n.^o ciclos, se procede a calcular el número de ciclos que hacen falta llenar para conseguir los 50 ms, esto se consigue restando n.^o ciclos al valor de desbordamiento:

$$65.536 - 50.000 = 15.536 \text{ ciclos hay que llenar para conseguir 50 ms}$$

Por último, para que se genere una interrupción cada 50 ms, se asigna 15.536 en hexadecimal al TL y TH del Timer0:

$$15.536d = 3CB0h \rightarrow TH = 3Ch \text{ y } TL = B0h$$

3.2 PWM

En la práctica el PWM es utilizado para controlar la potencia de giro de los rodillos horizontales y verticales, además de la potencia de los ventiladores.

Para configurar el PWM lo primero es fijar la frecuencia de conmutación. El enunciado de la práctica no nos indica que el f_{PWM} sea igual a $1 \cdot 10^3$, por lo que se calcula el PWMP para conseguir fijar la frecuencia de conmutación a $1 \cdot 10^3$:

$$f_{PWM} = \frac{f_{osc}}{2(1 + PWMP) \cdot 255}$$

Despejando:

$$PWMP = \frac{f_{osc}}{2 \cdot f_{PWM} \cdot 255} - 1$$

Sustituyendo:

$$PWMP = \frac{12 \cdot 10^6}{2 \cdot 1 \cdot 10^3 \cdot 255} - 1 \approx 22,53 \approx 23d = 17h$$

Una vez calculado esto, se define la formula del Duty-cycle para poder más adelante calcular los valores para activar el lavado al 50% o al 85%:

$$Duty - cycle \rightarrow D = 1 - \frac{PWMx}{255}$$

Al conocer esta fórmula, se aplica al 50%:

$$50\% \rightarrow 0,5 = 1 - \frac{PWM0}{255} = 127,5$$

Sustituyendo y despejando:

$$PWM0 = 127,5 \approx 128d = 80h$$

Ahora se repite el proceso con 85%:

$$85\% \rightarrow 0,85 = 1 - \frac{PWM1}{255} = 38,25$$

Sustituyendo y despejando:

$$PWM1 = 38,25 \approx 38d = 26h$$

Por último, se calcula el 0% y el 100% de velocidad:

$$0\% \rightarrow PWMx = FFh$$

$$100\% \rightarrow PWMx = 00h$$

3.3 ADC y Sensor de distancia

En el proyecto, el ADC ha sido utilizado para controlar que los rodillos verticales y horizontales estén a la distancia necesaria en todo momento. Para ello ha sido necesaria la realización de algunos cálculos.

Para empezar, gracias al enunciado sabemos que:

$$\text{Salida analógica} = 50 \frac{mV}{cm}$$

Sabiendo este dato se calculan en analógico todas las distancias requeridas:

$$37,5\text{cm} \rightarrow 50 \cdot 10^{-3} \cdot 37,5 = 1,875V$$

$$42,5\text{cm} \rightarrow 50 \cdot 10^{-3} \cdot 42,5 = 2,125V$$

$$40\text{cm} \rightarrow 50 \cdot 10^{-3} \cdot 40 = 2V$$

Una vez se conocen los valores de la distancia en analógico, se pasan estos valores a digital. Cabe destacar que sólo es necesaria la parte alta del ADC, por lo que solamente usamos el ADCH:

$$37,5\text{cm} \rightarrow 256 \cdot \frac{1,875 - 0}{5 - 0} = 96d = 60h$$

$$42,5\text{cm} \rightarrow 256 \cdot \frac{2,125 - 0}{5 - 0} = 108,8d \approx 108d = 6Ch$$

$$40\text{cm} \rightarrow 256 \cdot \frac{2 - 0}{5 - 0} = 102,4d \approx 102d = 66h$$

En el código, con el fin de saber si se ha superado o no la distancia, se restarán los valores que se han obtenido. Se obtendrá una conclusión según si el resultado es negativo, positivo o nulo.

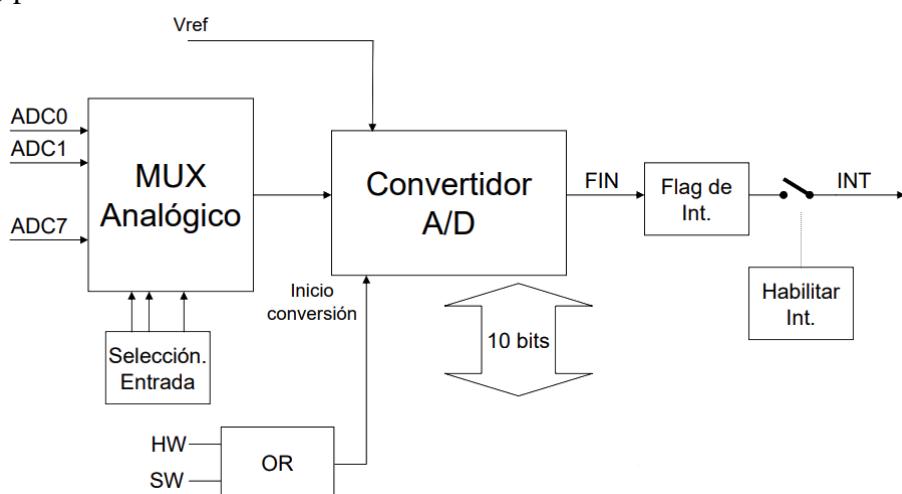


Figura 3.3.1.: Esquema del ADC

4 Código

```

ESTADO EQU R0 ; ESTADO
EVENTO EQU R1 ; EVENTO
;PUERTO 0
SEM_Verde EQU P0.0
SEM_Rojo EQU P0.1
S_PLAT EQU P0.2
FICHA_TIPO EQU P0.3
S_FICHA EQU P0.4
P_START EQU P0.5
AL_START EQU P0.6
BPOS_START EQU P0.7
;PUERTO 1
BMOV_FRONT EQU P1.0
BMOV_BACK EQU P1.1
FC_RH_TOP EQU P1.2
RH_UPTO EQU P1.3
RH_DOWNT0 EQU P1.4
FC_RV_BORDE EQU P1.5
RV_OUTTO EQU P1.6
RV_INTO EQU P1.7
;PUERTO 2
EV_AGUA EQU P2.0
EV_JABON EQU P2.1
S_CAR EQU P2.3
FC_RV_CENTRO EQU P2.4
LED_LNOR EQU P2.5
LED_LINT EQU P2.6
;CANALES PWM
PWM0 EQU 0x00FC
PWM1 EQU 0x00FD
PWMP EQU 0x00FE
;ADC
ADCON EQU 0x00C5
ADCH EQU 0x00C6
FLAG_ADC EQU 0x0020.6 ; Flag para fin de conversión
;TIMERS
TLO EQU 0x008A ; Parte baja del timer0
TH0 EQU 0x008C ; Parte alta del timer0
TICK_1 EQU 0x0020.0 ; Flag para 1 segundo
TICK_4 EQU 0x0020.1 ; Flag para 4 segundo
TICK_5 EQU 0x0020.2 ; Flag para 5 segundo
TICK_30 EQU 0x0020.3 ; Flag para 30 segundo
TICK_60 EQU 0x0020.4 ; Flag para 60 segundo
TICK_50ms EQU 0x0020.5 ; Flag para 50 milisegundos
;CONTADORES
COUNT_1 EQU 0x0021 ; CONTADOR PARA 1S
COUNT_4 EQU 0x0022 ; CONTADOR PARA 4S
COUNT_5 EQU 0x0023 ; CONTADOR PARA 5S
COUNT_30 EQU 0x0024 ; CONTADOR PARA 30S
COUNT_60 EQU 0x0025 ; CONTADOR PARA 60S
;RODILLOS
DIST_RH_40 EQU 0x0026.0 ; Flag cuando los rod. hor. estan a 40cm
DIST_RV_40 EQU 0x0026.1 ; Flag cuando los rod. ver. estan a 40cm
;VENTILADORES
VENTS_ON EQU 0x0026.2 ; Flag cuando los vent. estan activados
;-----
ORG 0x0000
AJMP INICIO
;-----
ORG 0x000B
TIMERO_ISR_SALTO:
LJMP TIMERO_ISR
;-----
ORG 0x0053
ADC_ISR:
PUSH ACC
PUSH PSW
SETB FLAG_ADC
ANL ADCON,#0xEF ; Bit ADCI a 0

```

```

POP      PSW
POP      ACC
RETI

;-----
ORG 0x006B
LIMPIAR_TEMP:
    CLR      TCON.4          ; Para el Timer 0
    MOV      TL0, #0xB0        ; Parte baja para 50 ms
    MOV      TH0, #0x3C        ; Parte alta para 50 ms
    MOV      COUNT_1, #20
    MOV      COUNT_4, #4
    MOV      COUNT_5, #5
    MOV      COUNT_30, #6
    MOV      COUNT_60, #2
    CLR      TICK_1
    CLR      TICK_4
    CLR      TICK_5
    CLR      TICK_30
    CLR      TICK_60
    RET

;-----
ORG 0x008D
INICIO:
    ACALL   INICIALIZACIONES ; Inicializaciones
BUCLE:
    LCALL   MAQ_ESTADOS      ; Programa principal
    AJMP   BUCLE             ; Bucle

INICIALIZACIONES:
    MOV      COUNT_1, #20
    MOV      COUNT_4, #4
    MOV      COUNT_5, #5
    MOV      COUNT_30, #6
    MOV      COUNT_60, #2
    MOV      ESTADO, #00
    MOV      EVENTO, #00
    MOV      PWM0, #0xFF       ; PWM0 apagado
    MOV      PWM1, #0xFF       ; PWM1 apagado
    MOV      PWMP, #0x17        ; PWMP = 23
    MOV      TMOD, #01H         ; Configura el Timer 0 en modo 1 (16 bits)
    MOV      TL0, #0xB0        ; Parte baja para 50 ms
    MOV      TH0, #0x3C        ; Parte alta para 50 ms
    SETB   IE.1               ; Habilita interrupciones del Timer 0
    SETB   IE.6               ; Habilita interrupciones del ADC
    SETB   IE.7               ; Habilita interrupciones globales
    MOV      ADCON, #0x00
    CLR      VENTS_ON
;Configuramos los puertos
    CLR      SEM_VERDE
    CLR      SEM_ROJO
    CLR      AL_START
    CLR      BMOV_FRONT
    CLR      BMOV_BACK
    CLR      RH_UPTO
    CLR      RH_DWNTO
    CLR      RV_OUTTO
    CLR      RV_INTO
    CLR      EV_AQUA
    CLR      EV_JABON
    CLR      LED_LNOR
    CLR      LED_LINT
    CLR      P2.2
    CLR      P2.7
    SETB   SEM_VERDE
    CLR      DIST_RH_40
    CLR      DIST_RV_40
    RET

;-----
MAQ_ESTADOS:
    MOV      A, ESTADO
    RL      A
    MOV      DPTR, #LISTA_EST

```

```

JMP    @A+DPTR
LISTA_EST:
AJMP   ESPERA_COCHE           ; Estado 0
AJMP   NUEVO_COCHE            ; Estado 1
AJMP   ESPERA_FICHA            ; Estado 2
AJMP   INSERTADA_FICHA         ; Estado 3
AJMP   COLOCADO_COCHE1          ; Estado 4
AJMP   ENJABONADO               ; Estado 5
AJMP   COLOCADO_COCHE2          ; Estado 6
AJMP   ROD_HACIA_DENTRO_DETRES ; Estado 7
AJMP   ROD_HACIA FUERA_DETRES ; Estado 8
AJMP   COL_PUENTE_SOBRERO_COCHE ; Estado 9
AJMP   ROD_VDENTRO_HABAJO        ; Estado 10
AJMP   ROD_HV_DIST_SEGURIDAD      ; Estado 11
AJMP   ROD_HVPOSORIG             ; Estado 12
AJMP   ROD_HACIA_DENTRO_DELANTE ; Estado 13
AJMP   ROD_HACIA FUERA_DELANTE ; Estado 14
AJMP   COLOCADO_COCHE3            ; Estado 15
AJMP   ACLARADO                  ; Estado 16
AJMP   FIN_ACLARADO              ; Estado 17
AJMP   COLOCADO_COCHE4            ; Estado 18
AJMP   SECADO                     ; Estado 19
AJMP   FIN_SECADO                 ; Estado 20
AJMP   FINAL_LAVADO                ; Estado 21

ESPERA_COCHE:
ACALL  GN_EVEN_0
ACALL  MQ_EVEN_0 RET

NUEVO_COCHE:
ACALL  GN_EVEN_1
ACALL  MQ_EVEN_1
RET

ESPERA_FICHA:
ACALL  GN_EVEN_2
ACALL  MQ_EVEN_2
RET

INSERTADA_FICHA:
ACALL  GN_EVEN_3
ACALL  MQ_EVEN_3
RET

COLOCADO_COCHE1:
ACALL  GN_EVEN_4
ACALL  MQ_EVEN_4
RET

ENJABONADO:
ACALL  GN_EVEN_5
ACALL  MQ_EVEN_5
RET

COLOCADO_COCHE2:
ACALL  GN_EVEN_6
ACALL  MQ_EVEN_6
RET

ROD_HACIA_DENTRO_DETRES:
ACALL  GN_EVEN_7
ACALL  MQ_EVEN_7
RET

ROD_HACIA_FUERA_DETRES:
ACALL  GN_EVEN_8
ACALL  MQ_EVEN_8
RET

COL_PUENTE_SOBRERO_COCHE:
ACALL  GN_EVEN_9
ACALL  MQ_EVEN_9
RET

ROD_VDENTRO_HABAJO:
ACALL  GN_EVEN_10
ACALL  MQ_EVEN_10
RET

ROD_HV_DIST_SEGURIDAD:
ACALL  GN_EVEN_11
ACALL  MQ_EVEN_11

RET ROD_HVPOSORIG:

```

```

        ACALL  GN_EVEN_12
        ACALL  MQ_EVEN_12
        RET
ROD_HACIA_DENTRO_DELANTE:
        ACALL  GN_EVEN_13
        ACALL  MQ_EVEN_13
        RET
ROD_HACIA_FUERA_DELANTE:
        ACALL  GN_EVEN_14
        ACALL  MQ_EVEN_14
        RET
COLOCADO_COCH3:
        ACALL  GN_EVEN_15
        ACALL  MQ_EVEN_15
        RET
ACLARADO:
        ACALL  GN_EVEN_16
        ACALL  MQ_EVEN_16
        RET
FIN_ACLARADO:
        ACALL  GN_EVEN_17
        ACALL  MQ_EVEN_17
        RET
COLOCADO_COCH4:
        ACALL  GN_EVEN_18
        ACALL  MQ_EVEN_18
        RET
SECADO:
        ACALL  GN_EVEN_19
        LCALL  MQ_EVEN_19
        RET
FIN_SECADO:
        ACALL  GN_EVEN_20
        LCALL  MQ_EVEN_20
        RET
FINAL_LAVADO:
        ACALL  GN_EVEN_21
        LCALL  MQ_EVEN_21
        RET
;-----
GN_EVEN_0:
        MOV    EVENTO, #00
        JNB    S_PLAT, FIN_GN_EVEN_0
        MOV    EVENTO, #1
        AJMP   FIN_GN_EVEN_0
FIN_GN_EVEN_0:
        RET
;-----
GN_EVEN_1:
        MOV    EVENTO, #00
        JNB    S_PLAT, GN_EVEN_1_2
        JB     TICK_5, GN_EVEN_1_3
        AJMP   FIN_GN_EVEN_1
GN_EVEN_1_2:
        MOV    EVENTO, #2
        AJMP   FIN_GN_EVEN_1
GN_EVEN_1_3:
        ACALL  LIMPIAR_TEMP
        MOV    EVENTO, #3
        AJMP   FIN_GN_EVEN_1
FIN_GN_EVEN_1:
        RET
;-----
GN_EVEN_2:
        MOV    EVENTO, #00
        JNB    S_PLAT, GN_EVEN_2_2
        JB     TICK_30, GN_EVEN_2_4
        JB     S_FICHA, GN_EVEN_2_5
        AJMP   FIN_GN_EVEN_2
GN_EVEN_2_2:
        ACALL  LIMPIAR_TEMP

```

```

        MOV     EVENTO, #2           ;Si S_PLAT es 0, se pasa al evento 2
        AJMP    FIN_GN_EVEN_2
GN_EVEN_2_4:
        ACALL   LIMPIAR_TEMP
        MOV     EVENTO, #4           ;Pasados 30 segs, se pasa al evento 4
        AJMP    FIN_GN_EVEN_2
GN_EVEN_2_5:
        ACALL   LIMPIAR_TEMP
        MOV     EVENTO, #5           ;Si S_FICHA se pone a 1, es decir, se
        AJMP    FIN_GN_EVEN_2       ;inserta una ficha, se pasa al evento 5
FIN_GN_EVEN_2:
        RET

;-----
GN_EVEN_3:
        MOV     EVENTO, #00          ;Evento nulo
        JB      TICK_60, GN_EVEN_3_6
        JB      TICK_30, GN_EVEN_3_4
        JB      P_START, GN_EVEN_3_7
        AJMP    FIN_GN_EVEN_3
GN_EVEN_3_4:
        MOV     EVENTO, #4           ;Pasados 30 segs, se pasa al evento 4
        CLR     TICK_30
        AJMP    FIN_GN_EVEN_3
GN_EVEN_3_6:
        ACALL   LIMPIAR_TEMP
        MOV     EVENTO, #6           ;Pasados 60 segs, se pasa al evento 6
        AJMP    FIN_GN_EVEN_3
GN_EVEN_3_7:
        ACALL   LIMPIAR_TEMP
        MOV     EVENTO, #7           ;Si P_START es 1, se pasa al evento 7
        AJMP    FIN_GN_EVEN_3
FIN_GN_EVEN_3:
        RET

;-----
GN_EVEN_4:
        MOV     EVENTO, #00          ;Evento nulo JB S_CAR, GN_EVEN_4_8
        AJMP    FIN_GN_EVEN_4
GN_EVEN_4_8:
        MOV     EVENTO, #8           ;Detectado el coche, se pasa al evento 8
        AJMP    FIN_GN_EVEN_4
FIN_GN_EVEN_4:
        RET

;-----
GN_EVEN_5:
        MOV     EVENTO, #00          ;Evento nulo
        JNB    S_CAR, GN_EVEN_5_9
        AJMP    FIN_GN_EVEN_5
GN_EVEN_5_9:
        MOV     EVENTO, #9           ;Si no se detecta el coche, se pasa al
        AJMP    FIN_GN_EVEN_5       ;evento 9
FIN_GN_EVEN_5:
        RET

;-----
GN_EVEN_6:
        MOV     EVENTO, #00          ;Evento nulo
        JB      TICK_4, GN_EVEN_6_10
        AJMP    FIN_GN_EVEN_6
GN_EVEN_6_10:
        ACALL   LIMPIAR_TEMP MOV EVENTO, #10 ;Si han pasado 4 segundos, se pasa al
        AJMP    FIN_GN_EVEN_6       ;evento 10
FIN_GN_EVEN_6:
        RET

;-----
GN_EVEN_7:
        MOV     EVENTO, #00          ;Evento nulo
        JNB    FC_RV_CENTRO, FIN_GN_EVEN_7
        MOV     EVENTO, #11          ;Si los rod. ver. están en el centro, se
                                ;pasa al evento 11

```

```

        AJMP    FIN_GN_EVEN_7
FIN_GN_EVEN_7:
        RET
;-----
GN_EVEN_8:
        MOV     EVENTO, #00
        JNB     FC_RV_BORDE, FIN_GN_EVEN_8
        MOV     EVENTO, #12
;Evento nulo
;Si los rod. Ver. están en el borde, se pasa al evento 12

        AJMP    FIN_GN_EVEN_8
FIN_GN_EVEN_8:
        RET
;-----
GN_EVEN_9:
        MOV     EVENTO, #00
        JNB     S_CAR, FIN_GN_EVEN_9
        MOV     EVENTO, #8
        AJMP    FIN_GN_EVEN_9
;Evento nulo
;Detectado el coche, se pasa al evento 8

FIN_GN_EVEN_9:
        RET
;-----
GN_EVEN_10:
        SETB    TCON.4
        MOV     EVENTO, #00
        ACALL   DIST_RH
        ACALL   DIST_RV
        JNB     DIST_RH_40, FIN_GN_EVEN_10
        JNB     DIST_RV_40, FIN_GN_EVEN_10
        MOV     EVENTO, #15
;Evento nulo
;Si los rod. ver. y hor. están a 40cm o menos, se pasa al evento 13

        AJMP    FIN_GN_EVEN_10
DIST_RH:
        JB      DIST_RH_40, VOLVER
;Si los rod. hor. están a 40cm sale de la rutina

        ANL    ADCON, #0x00
        ORL    ADCON, #0x08
        JNB    FLAG_ADC, $
        CLR    FLAG_ADC
        JNB    TICK_50ms, $
        CLR    TICK_50ms
        CLR    C
        MOV    A, ADCH
        SUBB   A, #0x66
;Se calcula que la dist. del rod. Hor. sea menor o igual de 40cm

        JZ     SALTAR1
        JNC     VOLVER
;Si no está a 40cm o menos, sale de la rutina

SALTAR1:
        SETB    DIST_RH_40
        MOV     EVENTO, #13
        RET
;Si los rod. Ver. ya están a 40cm sale de la rutina

DIST_RV:
        JB      DIST_RV_40, VOLVER
;Esperar a que FLAG_ADC sea 1

        ANL    ADCON, #0x00
        ORL    ADCON, #0x09
        JNB    FLAG_ADC, $
        CLR    FLAG_ADC
        JNB    TICK_50ms, $
        CLR    TICK_50ms
        CLR    C
        MOV    A, ADCH
        SUBB   A, #0x66
;Espera 50ms
;Se calcula que la distancia del rod. Ver. menor o igual de 40cm

        JZ     SALTAR2
        JNC     VOLVER
;Si no está a 40cm o menos, sale de la rutina

SALTAR2:
        SETB    DIST_RV_40
        MOV     EVENTO, #14
        RET

```

```

VOLVER:
    RET

FIN_GN_EVEN_10:
    ACALL LIMPIAR_TEMP
    RET
;-----
GN_EVEN_11:
    MOV EVENTO, #00 ;Evento nulo
    SETB TCON.4
    ACALL GN_EVEN_11_16_17
    ACALL GN_EVEN_11_18_19
    JB S_CAR, FIN_GN_EVEN_11
    MOV EVENTO, #9 ;Si no se detecta el coche, se pasa al evento 9
    AJMP FIN_GN_EVEN_11

GN_EVEN_11_16_17:
    ANL ADCON, #0x00 ;Inicio de una nueva conversión en el puerto P5.1
    ORL ADCON, #0x09 ;Esperar a que FLAG_ADC sea 1

    JNB FLAG_ADC,$ ;Espera 50ms
    CLR FLAG_ADC
    JNB TICK_50ms,$ ;Valor para 37,5cm
    CLR TICK_50ms ;Si es negativo es que está a una dist. menor de 37,5cm
    CLR C
    MOV A, ADCH
    SUBB A, #0x60
    JC GN_EVEN_11_14 ;Valor para 42,5cm
    ;Si es positivo es que está a una dist. mayor de 42,5cm

    CLR C
    MOV A, ADCH
    SUBB A, #0x6C
    JNC GN_EVEN_11_15 ;Valor para 42,5cm
    ;Si es positivo es que está a una dist. mayor de 42,5cm

    RET

GN_EVEN_11_14:
    MOV EVENTO, #16
    RET

GN_EVEN_11_15:
    MOV EVENTO, #17
    RET

GN_EVEN_11_18_19:
    ANL ADCON, #0x00 ;Inicio de una nueva conversión en el puerto P5.0
    ORL ADCON, #0x08 ;Esperar a que FLAG_ADC sea 1

    JNB FLAG_ADC,$ ;Espera 50ms
    CLR FLAG_ADC
    JNB TICK_50ms,$ ;Valor para 37,5cm
    CLR TICK_50ms ;Si es negativo es que está a una dist. menor de 37,5cm
    CLR C
    MOV A, ADCH
    SUBB A, #0x60
    JC GN_EVEN_11_16 ;Valor para 42,5cm
    ;Si es positivo es que está a una dist. mayor de 42,5cm

    CLR C
    MOV A, ADCH
    SUBB A, #0x6C
    JNC GN_EVEN_11_17 ;Valor para 42,5cm
    ;Si es positivo es que está a una dist. mayor de 42,5cm

    RET

GN_EVEN_11_16:
    MOV EVENTO, #18
    RET

GN_EVEN_11_17:
    MOV EVENTO, #19
    RET

FIN_GN_EVEN_11:
    RET
;-----
GN_EVEN_12:
    MOV EVENTO, #00 ;Evento nulo
    ACALL GN_EVEN_12_18

```

```

        ACALL  GN_EVEN_12_12
        JB     TICK_4, GN_EVEN_12_10
                                ;Los rod. ver. y hor. tienen que estar en
                                ;sus posiciones originales

        AJMP   FIN_GN_EVEN_12
GN_EVEN_12_10:
        MOV    EVENTO, #10
        AJMP   FIN_GN_EVEN_12
GN_EVEN_12_18:
        JNB    RH_UPTO, FIN_GN_EVEN_12
        JNB    FC_RH_TOP, FIN_GN_EVEN_12
        MOV    EVENTO, #20
                                ;El rod. hor. está en el TOP, pasa al
                                ;evento 18

        RET
GN_EVEN_12_12:
        JNB    RV_OUTTO, FIN_GN_EVEN_12
        JNB    FC_RV_BORDE, FIN_GN_EVEN_12
        MOV    EVENTO, #12
                                ;Si el rod. ver. está en el BORDE, se pasa
                                ;al evento 12

        RET
FIN_GN_EVEN_12:
        RET
;-----
GN_EVEN_13:
        MOV    EVENTO, #00
        JNB    FC_RV_CENTRO, FIN_GN_EVEN_13
        MOV    EVENTO, #11
                                ;Evento nulo
                                ;Si los rod. ver. están en el centro, se
                                ;pasa al evento 11

        AJMP   FIN_GN_EVEN_13
FIN_GN_EVEN_13:
        RET
;-----
GN_EVEN_14:
        MOV    EVENTO, #00
        JNB    FC_RV_BORDE, FIN_GN_EVEN_14
        MOV    EVENTO, #12
                                ;Evento nulo
                                ;Si los rod. ver. están en el borde, se
                                ;pasa al evento 12

        AJMP   FIN_GN_EVEN_14
FIN_GN_EVEN_14:
        RET
;-----
GN_EVEN_15:
        MOV    EVENTO, #00
        JNB    S_CAR, FIN_GN_EVEN_15
        MOV    EVENTO, #8
        AJMP   FIN_GN_EVEN_15
FIN_GN_EVEN_15:
        RET
                                ;Evento nulo
                                ;Detectado el coche, se pasa al evento 8

;-----
GN_EVEN_16:
        MOV    EVENTO, #00
        JB     S_CAR, FIN_GN_EVEN_16
        MOV    EVENTO, #9
        AJMP   FIN_GN_EVEN_16
FIN_GN_EVEN_16:
        RET
                                ;Evento nulo
                                ;No detectado el coche, se pasa al evento 9

;-----
GN_EVEN_17:
        MOV    EVENTO, #00
        JB     TICK_4, GN_EVEN_17_10
        AJMP   FIN_GN_EVEN_17
GN_EVEN_17_10:
        ACALL  LIMPIAR_TEMP
        MOV    EVENTO, #10
        AJMP   FIN_GN_EVEN_17
FIN_GN_EVEN_17:
        RET
                                ;Pasados 4 segs, se pasa al evento 10

;-----
GN_EVEN_18:
        MOV    EVENTO, #00
        JB     S_CAR, GN_EVEN_18_8
        AJMP   FIN_GN_EVEN_18
                                ;Evento nulo

```

```

GN_EVEN_18_8:
    ACALL LIMPIAR_TEMP
    MOV EVENTO, #8
    AJMP FIN_GN EVEN_18
FIN_GN EVEN_18:
    RET
;-----
GN_EVEN_19:
    MOV EVENTO, #00
    JB S_CAR, FIN_GN EVEN_19
    MOV EVENTO, #9
    AJMP FIN_GN EVEN_19
FIN_GN EVEN_19:
    RET
;-----
GN_EVEN_20:
    MOV EVENTO, #00
    JNB BPOS_START, FIN_GN EVEN_20
    MOV EVENTO, #21
    AJMP FIN_GN EVEN_20
FIN_GN EVEN_20:
    RET
;-----
GN_EVEN_21:
    MOV EVENTO, #00
    JNB S_PLAT, GN EVEN_21_2
    JB TICK_1, GN EVEN_21_22
    AJMP FIN_GN EVEN_21
GN EVEN_21_2:
    MOV EVENTO, #2
    AJMP FIN_GN EVEN_21
GN EVEN_21_22:
    ACALL LIMPIAR_TEMP
    MOV EVENTO, #22
    AJMP FIN_GN EVEN_21
FIN_GN EVEN_21:
    RET
;-----
MQ_EVEN_0:
    MOV A, EVENTO
    RL A
    MOV DPTR, #LIST_EVEN_MQEV_0
    JMP @A+DPTR
LIST_EVEN_MQEV_0:
    RET ; Evento 0
    NOP ; Descripción Estado 0. Evento 1
    NOP ; Descripción Estado 0. Evento 2
    RET ; Descripción Estado 0. Evento 3
    NOP ; Descripción Estado 0. Evento 4
    RET ; Descripción Estado 0. Evento 5
    NOP ; Descripción Estado 0. Evento 6
    RET ; Descripción Estado 0. Evento 7
    NOP ; Descripción Estado 0. Evento 8
    RET ; Descripción Estado 0. Evento 9
    NOP ; Descripción Estado 0. Evento 10
    RET ; Descripción Estado 0. Evento 11
    NOP ; Descripción Estado 0. Evento 12
    RET ; Descripción Estado 0. Evento 13
    NOP ; Descripción Estado 0. Evento 14

```

```

NOP ; Descripción Estado 0. Evento 14
RET
NOP ; Descripción Estado 0. Evento 15
RET
NOP ; Descripción Estado 0. Evento 16
RET
NOP ; Descripción Estado 0. Evento 17
RET
NOP ; Descripción Estado 0. Evento 18
RET
NOP ; Descripción Estado 0. Evento 19
RET
NOP ; Descripción Estado 0. Evento 20
RET
NOP ; Descripción Estado 0. Evento 21
RET
NOP ; Descripción Estado 0. Evento 22
MEO_EV_1:
    MOV ESTADO, #01
    RET
;-----
MQ_EVEN_1:
    SETB TCON.4
    MOV A, EVENTO
    RL A
    MOV DPTR, #LIST_EVEN_MQEV_1
    JMP @A+DPTR
LIST_EVEN_MQEV_1:
    RET ; Evento 0
    NOP
    RET ; Descripción Estado 1. Evento 1
    NOP
    AJMP ME1_EV_2 ; Descripción Estado 1. Evento 2
    AJMP ME1_EV_3 ; Descripción Estado 1. Evento 3
    RET ; Descripción Estado 1. Evento 4
    NOP
    RET ; Descripción Estado 1. Evento 5
    NOP
    RET ; Descripción Estado 1. Evento 6
    NOP
    RET ; Descripción Estado 1. Evento 7
    NOP
    RET ; Descripción Estado 1. Evento 8
    NOP
    RET ; Descripción Estado 1. Evento 9
    NOP
    RET ; Descripción Estado 1. Evento 10
    NOP
    RET ; Descripción Estado 1. Evento 11
    NOP
    RET ; Descripción Estado 1. Evento 12
    NOP
    RET ; Descripción Estado 1. Evento 13
    NOP
    RET ; Descripción Estado 1. Evento 14
    NOP
    RET ; Descripción Estado 1. Evento 15
    NOP
    RET ; Descripción Estado 1. Evento 16
    NOP
    RET ; Descripción Estado 1. Evento 17
    NOP
    RET ; Descripción Estado 1. Evento 18
    NOP
    RET ; Descripción Estado 1. Evento 19
    NOP
    RET ; Descripción Estado 1. Evento 20
    NOP
    RET ; Descripción Estado 1. Evento 21
    NOP

```

```

        RET ; Descripción Estado 1. Evento 22
ME1_EV_2:
        MOV ESTADO, #00
        RET
ME1_EV_3:
        CLR SEM_VERDE ; Acción 2
        SETB SEM_ROJO ; Acción 3
        MOV ESTADO, #02
        RET
;-----
MQ_EVEN_2:
        SETB TCON.4
        MOV A, EVENTO
        RL A
        MOV DPTR, #LIST EVEN_MQEV_2
        JMP @A+DPTR
LIST_EVEN_MQEV_2:
        RET ; Evento 0
        NOP
        RET ; Descripción Estado 2. Evento 1
        NOP
        AJMP ME2_EV_2 ; Descripción Estado 2. Evento 2
        RET ; Descripción Estado 2. Evento 3
        NOP
        AJMP ME2_EV_4 ; Descripción Estado 2. Evento 4
        AJMP ME2_EV_5 ; Descripción Estado 2. Evento 5
        RET ; Descripción Estado 2. Evento 6
        NOP
        RET ; Descripción Estado 2. Evento 7
        NOP
        RET ; Descripción Estado 2. Evento 8
        NOP
        RET ; Descripción Estado 2. Evento 9
        NOP
        RET ; Descripción Estado 2. Evento 10
        NOP
        RET ; Descripción Estado 2. Evento 11
        NOP
        RET ; Descripción Estado 2. Evento 12
        NOP
        RET ; Descripción Estado 2. Evento 13
        NOP
        RET ; Descripción Estado 2. Evento 14
        NOP
        RET ; Descripción Estado 2. Evento 15
        NOP
        RET ; Descripción Estado 2. Evento 16
        NOP
        RET ; Descripción Estado 2. Evento 17
        NOP
        RET ; Descripción Estado 2. Evento 18
        NOP
        RET ; Descripción Estado 2. Evento 19
        NOP
        RET ; Descripción Estado 2. Evento 20
        NOP
        RET ; Descripción Estado 2. Evento 21
        NOP
        RET ; Descripción Estado 2. Evento 2
ME2_EV_2:
        SETB SEM_VERDE ; Acción 1
        CLR SEM_ROJO ; Acción 4
        CLR AL_START ; Acción 6
        MOV ESTADO, #00
        RET
ME2_EV_4:
        SETB AL_START ; Acción 5
        RET
ME2_EV_5:
        CLR AL_START ; Acción 6
        LCALL LED_MODO_ONOFF ; Acción 7

```

```

        MOV      ESTADO, #03
        RET

;-----
MQ_EVEN_3:
        SETB    TCON.4
        MOV     A, EVENTO
        RL     A
        MOV     DPTR, #LIST_EVEN_MQEV_3
        JMP     @A+DPTR
LIST_EVEN_MQEV_3:
        RET          ; Evento 0
        NOP
        RET          ; Descripción Estado 3. Evento 1
        NOP
        RET          ; Descripción Estado 3. Evento 2
        NOP
        RET          ; Descripción Estado 3. Evento 3
        NOP
        AJMP    ME3_EV_4
        RET          ; Descripción Estado 3. Evento 4
        NOP          ; Descripción Estado 3. Evento 5
        AJMP    ME3_EV_7
        RET          ; Descripción Estado 3. Evento 6
        NOP          ; Descripción Estado 3. Evento 7
        RET          ; Descripción Estado 3. Evento 8
        NOP
        RET          ; Descripción Estado 3. Evento 9
        NOP
        RET          ; Descripción Estado 3. Evento 10
        NOP
        RET          ; Descripción Estado 3. Evento 11
        NOP
        RET          ; Descripción Estado 3. Evento 12
        NOP
        RET          ; Descripción Estado 3. Evento 13
        NOP
        RET          ; Descripción Estado 3. Evento 14
        NOP
        RET          ; Descripción Estado 3. Evento 15
        NOP
        RET          ; Descripción Estado 3. Evento 16
        NOP
        RET          ; Descripción Estado 3. Evento 17
        NOP
        RET          ; Descripción Estado 3. Evento 18
        NOP
        RET          ; Descripción Estado 3. Evento 19
        NOP
        RET          ; Descripción Estado 3. Evento 20
        NOP
        RET          ; Descripción Estado 3. Evento 21
        NOP
        RET          ; Descripción Estado 3. Evento 22
ME3_EV_4:
        SETB    AL_START
        RET          ; Acción 5
ME3_EV_6:
        CLR     AL_START
        CLR     BMOV_BACK
        SETB    BMOV_FRONT
        MOV     ESTADO, #04
        RET          ; Acción 6
        ; Acción 10
ME3_EV_7:
        CLR     AL_START
        CLR     BMOV_BACK
        SETB    BMOV_FRONT
        MOV     ESTADO, #04
        RET          ; Acción 6
        ; Acción 10
;-----
MQ_EVEN_4:
        MOV     A, EVENTO
        RL     A

```

```

        MOV      DPTR, #LIST_EVEN_MQEV_4
        JMP      @A+DPTR
LIST_EVEN_MQEV_4:
        RET          ; Evento 0
        NOP
        RET          ; Descripción Estado 4. Evento 1
        NOP
        RET          ; Descripción Estado 4. Evento 2
        NOP
        RET          ; Descripción Estado 4. Evento 3
        NOP
        RET          ; Descripción Estado 4. Evento 4
        NOP
        RET          ; Descripción Estado 4. Evento 5
        NOP
        RET          ; Descripción Estado 4. Evento 6
        NOP
        RET          ; Descripción Estado 4. Evento 7
        NOP
        AJMP     ME4_EV_8
        RET          ; Descripción Estado 4. Evento 8
        NOP          ; Descripción Estado 4. Evento 9
        RET          ; Descripción Estado 4. Evento 10
        NOP
        RET          ; Descripción Estado 4. Evento 11
        NOP
        RET          ; Descripción Estado 4. Evento 12
        NOP
        RET          ; Descripción Estado 4. Evento 13
        NOP
        RET          ; Descripción Estado 4. Evento 14
        NOP
        RET          ; Descripción Estado 4. Evento 15
        NOP
        RET          ; Descripción Estado 4. Evento 16
        NOP
        RET          ; Descripción Estado 4. Evento 17
        NOP
        RET          ; Descripción Estado 4. Evento 18
        NOP
        RET          ; Descripción Estado 4. Evento 19
        NOP
        RET          ; Descripción Estado 4. Evento 20
        NOP
        RET          ; Descripción Estado 4. Evento 21
        NOP
        RET          ; Descripción Estado 4. Evento 22
ME4_EV_8:
        SETB     EV_AGUA       ; Acción 8
        SETB     EV_JABON       ; Acción 9
        MOV      ESTADO,#05
        RET
;-----
MQ_EVEN_5:
        MOV      A, EVENTO
        RL      A
        MOV      DPTR, #LIST_EVEN_MQEV_5
        JMP      @A+DPTR
LIST_EVEN_MQEV_5:
        RET          ; Evento 0
        NOP
        RET          ; Descripción Estado 5. Evento 1
        NOP
        RET          ; Descripción Estado 5. Evento 2
        NOP
        RET          ; Descripción Estado 5. Evento 3
        NOP
        RET          ; Descripción Estado 5. Evento 4
        NOP
        RET          ; Descripción Estado 5. Evento 5
        NOP

```

```

RET ; Descripción Estado 5. Evento 6
NOP
RET ; Descripción Estado 5. Evento 7
NOP
RET ; Descripción Estado 5. Evento 8
NOP
AJMP ME5_EV_9 ; Descripción Estado 5. Evento 9
RET ; Descripción Estado 5. Evento 10
NOP
RET ; Descripción Estado 5. Evento 11
NOP
RET ; Descripción Estado 5. Evento 12
NOP
RET ; Descripción Estado 5. Evento 13
NOP
RET ; Descripción Estado 5. Evento 14
NOP
RET ; Descripción Estado 5. Evento 15
NOP
RET ; Descripción Estado 5. Evento 16
NOP
RET ; Descripción Estado 5. Evento 17
NOP
RET ; Descripción Estado 5. Evento 18
NOP
RET ; Descripción Estado 5. Evento 19
NOP
RET ; Descripción Estado 5. Evento 20
NOP
RET ; Descripción Estado 5. Evento 21
NOP
RET ; Descripción Estado 5. Evento 22
ME5_EV_9:
CLR EV_AGUA ; Acción 8
CLR EV_JABON ; Acción 9
MOV ESTADO,#06
RET
-----
MQ_EVEN_6:
SETB TCON.4
MOV A, EVENTO
RL A
MOV DPTR, #LIST_EVEN_MQEV_6
JMP @A+DPTR
LIST_EVEN_MQEV_6:
RET ; Evento 0
NOP
RET ; Descripción Estado 6. Evento 1
NOP
RET ; Descripción Estado 6. Evento 2
NOP
RET ; Descripción Estado 6. Evento 3
NOP
RET ; Descripción Estado 6. Evento 4
NOP
RET ; Descripción Estado 6. Evento 5
NOP
RET ; Descripción Estado 6. Evento 6
NOP
RET ; Descripción Estado 6. Evento 7
NOP
RET ; Descripción Estado 6. Evento 8
NOP
RET ; Descripción Estado 6. Evento 9
NOP
AJMP ME6_EV_10 ; Descripción Estado 6. Evento 10
RET ; Descripción Estado 6. Evento 11
NOP
RET ; Descripción Estado 6. Evento 12
NOP
RET ; Descripción Estado 6. Evento 13

```

```

NOP ; Descripción Estado 6. Evento 14
RET
NOP ; Descripción Estado 6. Evento 15
RET
NOP ; Descripción Estado 6. Evento 16
RET
NOP ; Descripción Estado 6. Evento 17
RET
NOP ; Descripción Estado 6. Evento 18
RET
NOP ; Descripción Estado 6. Evento 19
RET
NOP ; Descripción Estado 6. Evento 20
RET
NOP ; Descripción Estado 6. Evento 21
RET
NOP ; Descripción Estado 6. Evento 22
RET

ME6_EV_10:
CLR    BMOV_FRONT ; Acción 10
LCALL  ROD_VER_DENTRO_ONOFF ; Acción 12
MOV    ESTADO,#07
RET

;-----
MQ_EVEN_7:
MOV    A, EVENTO
RL     A
MOV    DPTR, #LIST_EVEN_MQEV_7
JMP    @A+DPTR

LIST_EVEN_MQEV_7:
RET    ; Evento 0
NOP
RET    ; Descripción Estado 7. Evento 1
NOP
RET    ; Descripción Estado 7. Evento 2
NOP
RET    ; Descripción Estado 7. Evento 3
NOP
RET    ; Descripción Estado 7. Evento 4
NOP
RET    ; Descripción Estado 7. Evento 5
NOP
RET    ; Descripción Estado 7. Evento 6
NOP
RET    ; Descripción Estado 7. Evento 7
NOP
RET    ; Descripción Estado 7. Evento 8
NOP
RET    ; Descripción Estado 7. Evento 9
NOP
RET    ; Descripción Estado 7. Evento 10
NOP
AJMP  ME7_EV_11 ; Descripción Estado 7. Evento 11
RET    ; Descripción Estado 7. Evento 12
NOP
RET    ; Descripción Estado 7. Evento 13
NOP
RET    ; Descripción Estado 7. Evento 14
NOP
RET    ; Descripción Estado 7. Evento 15
NOP
RET    ; Descripción Estado 7. Evento 16
NOP
RET    ; Descripción Estado 7. Evento 17
NOP
RET    ; Descripción Estado 7. Evento 18
NOP
RET    ; Descripción Estado 7. Evento 19
NOP
RET    ; Descripción Estado 7. Evento 20

```

```

        RET                                ; Descripción Estado 7. Evento 21
        NOP
        RET                                ; Descripción Estado 7. Evento 22
ME7_EV_11:
        LCALL    ROD_VER_FUERA_ONOFF      ; Acción 13
        MOV     ESTADO, #08
        RET

;-----
MQ_EVEN_8:
        MOV     A, EVENTO
        RL      A
        MOV     DPTR, #LIST EVEN_MQEV_8
        JMP     @A+DPTR

LIST_EVEN_MQEV_8:
        RET                                ; Evento 0
        NOP
        RET                                ; Descripción Estado 8. Evento 1
        NOP
        RET                                ; Descripción Estado 8. Evento 2
        NOP
        RET                                ; Descripción Estado 8. Evento 3
        NOP
        RET                                ; Descripción Estado 8. Evento 4
        NOP
        RET                                ; Descripción Estado 8. Evento 5
        NOP
        RET                                ; Descripción Estado 8. Evento 6
        NOP
        RET                                ; Descripción Estado 8. Evento 7
        NOP
        RET                                ; Descripción Estado 8. Evento 8
        NOP
        RET                                ; Descripción Estado 8. Evento 9
        NOP
        RET                                ; Descripción Estado 8. Evento 10
        NOP
        RET                                ; Descripción Estado 8. Evento 11
        NOP
        AJMP    ME8_EV_12                ; Descripción Estado 8. Evento 12
        RET                                ; Descripción Estado 8. Evento 13
        NOP
        RET                                ; Descripción Estado 8. Evento 14
        NOP
        RET                                ; Descripción Estado 8. Evento 15
        NOP
        RET                                ; Descripción Estado 8. Evento 16
        NOP
        RET                                ; Descripción Estado 8. Evento 17
        NOP
        RET                                ; Descripción Estado 8. Evento 18
        NOP
        RET                                ; Descripción Estado 8. Evento 19
        NOP
        RET                                ; Descripción Estado 8. Evento 20
        NOP
        RET                                ; Descripción Estado 8. Evento 21
        NOP
        RET                                ; Descripción Estado 8. Evento 22
ME8_EV_12:
        CLR     BMOV_FRONT               ; Acción 11
        SETB    BMOV_BACK
        LCALL    ROD_VER_FUERA_ONOFF      ; Acción 13
        MOV     ESTADO, #09
        RET

;-----
MQ_EVEN_9:
        MOV     A, EVENTO
        RL      A
        MOV     DPTR, #LIST EVEN_MQEV_9
        JMP     @A+DPTR

LIST_EVEN_MQEV_9:

```

```

RET ; Evento 0
NOP
RET ; Descripción Estado 9. Evento 1
NOP
RET ; Descripción Estado 9. Evento 2
NOP
RET ; Descripción Estado 9. Evento 3
NOP
RET ; Descripción Estado 9. Evento 4
NOP
RET ; Descripción Estado 9. Evento 5
NOP
RET ; Descripción Estado 9. Evento 6
NOP
RET ; Descripción Estado 9. Evento 7
NOP
AJMP ME9_EV_8 ; Descripción Estado 9. Evento 8
RET ; Descripción Estado 9. Evento 9
NOP
RET ; Descripción Estado 9. Evento 10
NOP
RET ; Descripción Estado 9. Evento 11
NOP
RET ; Descripción Estado 9. Evento 12
NOP
RET ; Descripción Estado 9. Evento 13
NOP
RET ; Descripción Estado 9. Evento 14
NOP
RET ; Descripción Estado 9. Evento 15
NOP
RET ; Descripción Estado 9. Evento 16
NOP
RET ; Descripción Estado 9. Evento 17
NOP
RET ; Descripción Estado 9. Evento 18
NOP
RET ; Descripción Estado 9. Evento 19
NOP
RET ; Descripción Estado 9. Evento 20
NOP
RET ; Descripción Estado 9. Evento 21
NOP
RET ; Descripción Estado 9. Evento 22
ME9_EV_8:
CLR BMOV_BACK ; Acción 11
LCALL ROD_VER_DENTRO_ONOFF ; Acción 12
LCALL ROD_HOR_ABAJO_ONOFF ; Acción 14
MOV ESTADO,#10
RET
-----
MQ_EVEN_10:
MOV A, EVENTO
RL A
MOV DPTR, #LIST_EVEN_MQEV_10
JMP @A+DPTR
LIST_EVEN_MQEV_10:
RET ; Evento 0
NOP
RET ; Descripción Estado 10. Evento 1
NOP
RET ; Descripción Estado 10. Evento 2
NOP
RET ; Descripción Estado 10. Evento 3
NOP
RET ; Descripción Estado 10. Evento 4
NOP
RET ; Descripción Estado 10. Evento 5
NOP
RET ; Descripción Estado 10. Evento 6

```

```

RET ; Descripción Estado 10. Evento 7
NOP
RET ; Descripción Estado 10. Evento 8
NOP
RET ; Descripción Estado 10. Evento 9
NOP
RET ; Descripción Estado 10. Evento 10
NOP
RET ; Descripción Estado 10. Evento 11
NOP
RET ; Descripción Estado 10. Evento 12
NOP
AJMP ME10_EV_13 ; Descripción Estado 10. Evento 13
AJMP ME10_EV_14 ; Descripción Estado 10. Evento 14
AJMP ME10_EV_15 ; Descripción Estado 10. Evento 15
RET ; Descripción Estado 10. Evento 16
NOP
RET ; Descripción Estado 10. Evento 17
NOP
RET ; Descripción Estado 10. Evento 18
NOP
RET ; Descripción Estado 10. Evento 19
NOP
RET ; Descripción Estado 10. Evento 20
NOP
RET ; Descripción Estado 10. Evento 21
NOP
RET ; Descripción Estado 10. Evento 22
ME10_EV_13:
    CLR RH_DOWNT0 ; Acción 14
    RET
ME10_EV_14:
    CLR RV_INTO ; Acción 12
    RET
ME10_EV_15:
    CLR BMOV_FRONT ; Acción 11
    SETB BMOV_BACK
    LCALL OFF_TODOS_MOV_ROD_HV ; Acción 16
    MOV ESTADO, #11
    RET
;-----
MQ_EVEN_11:
    MOV A, EVENTO
    RL A
    MOV DPTR, #LIST_EVEN_MQEV_11
    JMP @A+DPTR
LIST_EVEN_MQEV_11:
    RET ; Evento 0
    NOP
    RET ; Descripción Estado 11. Evento 1
    NOP
    RET ; Descripción Estado 11. Evento 2
    NOP
    RET ; Descripción Estado 11. Evento 3
    NOP
    RET ; Descripción Estado 11. Evento 4
    NOP
    RET ; Descripción Estado 11. Evento 5
    NOP
    RET ; Descripción Estado 11. Evento 6
    NOP
    RET ; Descripción Estado 11. Evento 7
    NOP
    RET ; Descripción Estado 11. Evento 8
    NOP
    AJMP ME11_EV_9 ; Descripción Estado 11. Evento 9
    RET ; Descripción Estado 11. Evento 10
    NOP
    RET ; Descripción Estado 11. Evento 11
    NOP
    RET ; Descripción Estado 11. Evento 12

```

```

NOP
RET ; Descripción Estado 11. Evento 13
NOP
RET ; Descripción Estado 11. Evento 14
NOP
RET ; Descripción Estado 11. Evento 15
NOP
AJMP ME11_EV_16 ; Descripción Estado 11. Evento 16
AJMP ME11_EV_17 ; Descripción Estado 11. Evento 17
AJMP ME11_EV_18 ; Descripción Estado 11. Evento 18
AJMP ME11_EV_19 ; Descripción Estado 11. Evento 19
RET ; Descripción Estado 11. Evento 20
NOP
RET ; Descripción Estado 11. Evento 21
NOP
RET ; Descripción Estado 11. Evento 22
ME11_EV_9:
LCALL OFF_TODOS_MOV_ROD_HV ; Acción 16
LCALL ROD_HOR_ARRIBA_ONOFF ; Acción 15
LCALL ROD_VER_FUERA_ONOFF ; Acción 13
MOV ESTADO, #12
RET
ME11_EV_16:
CLR RV_INTO ; Acción 13
SETB RV_OUTTO
RET
ME11_EV_17:
CLR RV_OUTTO ; Acción 12
SETB RV_INTO
RET
ME11_EV_18:
CLR RH_DWNTO ; Acción 15
SETB RH_UPTO
RET
ME11_EV_19:
CLR RH_UPTO ; Acción 14
SETB RH_DWNTO
RET
;-----
MQ_EVEN_12:
SETB TCON.4
MOV A, EVENTO
RL A
MOV DPTR, #LIST_EVEN_MQEV_12
JMP @A+DPTR
LIST_EVEN_MQEV_12:
RET ; Evento 0
NOP
RET ; Descripción Estado 12. Evento 1
NOP
RET ; Descripción Estado 12. Evento 2
NOP
RET ; Descripción Estado 12. Evento 3
NOP
RET ; Descripción Estado 12. Evento 4
NOP
RET ; Descripción Estado 12. Evento 5
NOP
RET ; Descripción Estado 12. Evento 6
NOP
RET ; Descripción Estado 12. Evento 7
NOP
RET ; Descripción Estado 12. Evento 8
NOP
RET ; Descripción Estado 12. Evento 9
NOP AJMP ME12_EV_10 ; Descripción Estado 12. Evento 10
RET ; Descripción Estado 12. Evento 11
NOP
AJMP ME12_EV_12 ; Descripción Estado 12. Evento 12
RET ; Descripción Estado 12. Evento 13
NOP

```

```

RET ; Descripción Estado 12. Evento 14
NOP
RET ; Descripción Estado 12. Evento 15
NOP
RET ; Descripción Estado 12. Evento 16
NOP
RET ; Descripción Estado 12. Evento 17
NOP
RET ; Descripción Estado 12. Evento 18
NOP
RET ; Descripción Estado 12. Evento 19
NOP
AJMP ME12_EV_20 ; Descripción Estado 12. Evento 20
RET ; Descripción Estado 12. Evento 21
NOP
RET ; Descripción Estado 12. Evento 22
ME12_EV_10:
CLR BMOV_BACK ; Acción 11
JB RH_UPTO,ME12_EV_10_AUX
JB RV_OUTTO,ME12_EV_10_AUX
LCALL ROD_VER_DENTRO_ONOFF ; Acción 12
ACALL LIMPIAR_TEMP
MOV ESTADO,#13

ME12_EV_10_AUX:
RET

ME12_EV_12:
CLR RV_OUTTO ; Acción 13
RET

ME12_EV_20:
CLR RH_UPTO ; Acción 15
RET
-----
MQ_EVEN_13:
MOV A, EVENTO
RL A
MOV DPTR, #LIST_EVEN_MQEV_13
JMP @A+DPTR
LIST_EVEN_MQEV_13:
RET ; Evento 0
NOP
RET ; Descripción Estado 13. Evento 1
NOP
RET ; Descripción Estado 13. Evento 2
NOP
RET ; Descripción Estado 13. Evento 3
NOP
RET ; Descripción Estado 13. Evento 4
NOP
RET ; Descripción Estado 13. Evento 5
NOP
RET ; Descripción Estado 13. Evento 6
NOP
RET ; Descripción Estado 13. Evento 7
NOP
RET ; Descripción Estado 13. Evento 8
NOP
RET ; Descripción Estado 13. Evento 9
NOP
RET ; Descripción Estado 13. Evento 10
NOP
AJMP ME13_EV_11 ; Descripción Estado 13. Evento 11
RET ; Descripción Estado 13. Evento 12
NOP
RET ; Descripción Estado 13. Evento 13
NOP
RET ; Descripción Estado 13. Evento 14
NOP
RET ; Descripción Estado 13. Evento 15
NOP
RET ; Descripción Estado 13. Evento 16

```

```

RET ; Descripción Estado 13. Evento 17
NOP
RET ; Descripción Estado 13. Evento 18
NOP
RET ; Descripción Estado 13. Evento 19
NOP
RET ; Descripción Estado 13. Evento 20
NOP
RET ; Descripción Estado 13. Evento 21
NOP
RET ; Descripción Estado 13. Evento 22
ME13_EV_11:
    LCALL ROD_VER_FUERA_ONOFF ; Acción 13
    MOV ESTADO, #14
    RET
;-----
MQ_EVEN_14:
    MOV A, EVENTO
    RL A
    MOV DPTR, #LIST EVEN_MQEV_14
    JMP @A+DPTR
LIST EVEN_MQEV_14:
    RET ; Evento 0
    NOP
    RET ; Descripción Estado 14. Evento 1
    NOP
    RET ; Descripción Estado 14. Evento 2
    NOP
    RET ; Descripción Estado 14. Evento 3
    NOP
    RET ; Descripción Estado 14. Evento 4
    NOP
    RET ; Descripción Estado 14. Evento 5
    NOP
    RET ; Descripción Estado 14. Evento 6
    NOP
    RET ; Descripción Estado 14. Evento 7
    NOP
    RET ; Descripción Estado 14. Evento 8
    NOP
    RET ; Descripción Estado 14. Evento 9
    NOP
    RET ; Descripción Estado 14. Evento 10
    NOP
    RET ; Descripción Estado 14. Evento 11
    NOP
    AJMP ME14_EV_12 ; Descripción Estado 14. Evento 12
    RET ; Descripción Estado 14. Evento 13
    NOP
    RET ; Descripción Estado 14. Evento 14
    NOP
    RET ; Descripción Estado 14. Evento 15
    NOP
    RET ; Descripción Estado 14. Evento 16
    NOP
    RET ; Descripción Estado 14. Evento 17
    NOP
    RET ; Descripción Estado 14. Evento 18
    NOP
    RET ; Descripción Estado 14. Evento 19
    NOP
    RET ; Descripción Estado 14. Evento 20
    NOP
    RET ; Descripción Estado 14. Evento 21
    NOP
    RET ; Descripción Estado 14. Evento 22
ME14_EV_12:
    CLR BMOV_BACK ; Acción 10
    SETB BMOV_FRONT
    LCALL ROD_VER_FUERA_ONOFF ; Acción 13
    MOV PWM0, #0xFF ; Acción 19

```

```

        MOV      ESTADO, #15
        RET
;-----
MQ_EVEN_15:
        MOV      A, EVENTO
        RL      A
        MOV      DPTR, #LIST_EVEN_MQEV_15
        JMP      @A+DPTR
LIST_EVEN_MQEV_15:
        RET          ; Evento 0
        NOP
        RET          ; Descripción Estado 15. Evento 1
        NOP
        RET          ; Descripción Estado 15. Evento 2
        NOP
        RET          ; Descripción Estado 15. Evento 3
        NOP
        RET          ; Descripción Estado 15. Evento 4
        NOP
        RET          ; Descripción Estado 15. Evento 5
        NOP
        RET          ; Descripción Estado 15. Evento 6
        NOP
        RET          ; Descripción Estado 15. Evento 7
        NOP
        AJMP     ME15_EV_8   ; Descripción Estado 15. Evento 8
        RET          ; Descripción Estado 15. Evento 9
        NOP
        RET          ; Descripción Estado 15. Evento 10
        NOP
        RET          ; Descripción Estado 15. Evento 11
        NOP
        RET          ; Descripción Estado 15. Evento 12
        NOP
        RET          ; Descripción Estado 15. Evento 13
        NOP
        RET          ; Descripción Estado 15. Evento 14
        NOP
        RET          ; Descripción Estado 15. Evento 15
        NOP
        RET          ; Descripción Estado 15. Evento 16
        NOP
        RET          ; Descripción Estado 15. Evento 17
        NOP
        RET          ; Descripción Estado 15. Evento 18
        NOP
        RET          ; Descripción Estado 15. Evento 19
        NOP
        RET          ; Descripción Estado 15. Evento 20
        NOP
        RET          ; Descripción Estado 15. Evento 21
        NOP
        RET          ; Descripción Estado 15. Evento 22
ME15_EV_8:
        SETB     EV_AGUA    ; Acción 8
        MOV      ESTADO, #16
        RET
;-----
MQ_EVEN_16:
        MOV      A, EVENTO
        RL      A
        MOV      DPTR, #LIST_EVEN_MQEV_16
        JMP      @A+DPTR
LIST_EVEN_MQEV_16:
        RET          ; Evento 0
        NOP
        RET          ; Descripción Estado 16. Evento 1
        NOP
        RET          ; Descripción Estado 16. Evento 2
        NOP
        RET          ; Descripción Estado 16. Evento 3

```

```

NOP
RET ; Descripción Estado 16. Evento 4
NOP
RET ; Descripción Estado 16. Evento 5
NOP
RET ; Descripción Estado 16. Evento 6
NOP
RET ; Descripción Estado 16. Evento 7
NOP
RET ; Descripción Estado 16. Evento 8
NOP
AJMP ME16_EV_9 ; Descripción Estado 16. Evento 9
RET ; Descripción Estado 16. Evento 10
NOP
RET ; Descripción Estado 16. Evento 11
NOP
RET ; Descripción Estado 16. Evento 12
NOP
RET ; Descripción Estado 16. Evento 13
NOP
RET ; Descripción Estado 16. Evento 14
NOP
RET ; Descripción Estado 16. Evento 15
NOP
RET ; Descripción Estado 16. Evento 16
NOP
RET ; Descripción Estado 16. Evento 17
NOP
RET ; Descripción Estado 16. Evento 18
NOP
RET ; Descripción Estado 16. Evento 19
NOP
RET ; Descripción Estado 16. Evento 20
NOP
RET ; Descripción Estado 16. Evento 21
NOP
RET ; Descripción Estado 16. Evento 22
ME16_EV_9:
CLR EV_AGUA ; Acción 8
MOV ESTADO,#17
RET
;-----
MQ_EVEN_17:
SETB TCON.4
MOV A, EVENTO
RL A
MOV DPTR, #LIST_EVEN_MQEV_17
JMP @A+DPTR
LIST_EVEN_MQEV_17:
RET ; Evento 0
NOP
RET ; Descripción Estado 17. Evento 1
NOP
RET ; Descripción Estado 17. Evento 2
NOP
RET ; Descripción Estado 17. Evento 3
NOP
RET ; Descripción Estado 17. Evento 4
NOP
RET ; Descripción Estado 17. Evento 5
NOP
RET ; Descripción Estado 17. Evento 6
NOP
RET ; Descripción Estado 17. Evento 7
NOP
RET ; Descripción Estado 17. Evento 8
NOP
RET ; Descripción Estado 17. Evento 9
AJMP ME17_EV_10 ; Descripción Estado 17. Evento 10
RET ; Descripción Estado 17. Evento 11

```

```

NOP ; Descripción Estado 17. Evento 12
RET
NOP ; Descripción Estado 17. Evento 13
RET
NOP ; Descripción Estado 17. Evento 14
RET
NOP ; Descripción Estado 17. Evento 15
RET
NOP ; Descripción Estado 17. Evento 16
RET
NOP ; Descripción Estado 17. Evento 17
RET
NOP ; Descripción Estado 17. Evento 18
RET
NOP ; Descripción Estado 17. Evento 19
RET
NOP ; Descripción Estado 17. Evento 20
RET
NOP ; Descripción Estado 17. Evento 21
RET
NOP ; Descripción Estado 17. Evento 22
RET

ME17_EV_10:
    CLR    BMOV_FRONT
    SETB   BMOV_BACK
    MOV    ESTADO,#18
    RET

;-----
MQ_EVEN_18:
    SETB   TCON.4
    MOV    A, EVENTO
    RL     A
    MOV    DPTR, #LIST_EVEN_MQEV_18
    JMP    @A+DPTR

LIST_EVEN_MQEV_18:
    RET    ; Evento 0
    NOP
    RET    ; Descripción Estado 18. Evento 1
    NOP
    RET    ; Descripción Estado 18. Evento 2
    NOP
    RET    ; Descripción Estado 18. Evento 3
    NOP
    RET    ; Descripción Estado 18. Evento 4
    NOP
    RET    ; Descripción Estado 18. Evento 5
    NOP
    RET    ; Descripción Estado 18. Evento 6
    NOP
    RET    ; Descripción Estado 18. Evento 7
    NOP
    LJMP   ME18_EV_8 ; Descripción Estado 18. Evento 8
    RET    ; Descripción Estado 18. Evento 9
    NOP
    RET    ; Descripción Estado 18. Evento 10
    NOP
    RET    ; Descripción Estado 18. Evento 11
    NOP
    RET    ; Descripción Estado 18. Evento 13
    NOP
    RET    ; Descripción Estado 18. Evento 14
    NOP
    RET    ; Descripción Estado 18. Evento 15
    NOP
    RET    ; Descripción Estado 18. Evento 16
    NOP
    RET    ; Descripción Estado 18. Evento 17
    NOP
    RET    ; Descripción Estado 18. Evento 18
    NOP
    RET    ; Descripción Estado 18. Evento 19

```

```

NOP
RET ; Descripción Estado 18. Evento 20
NOP
RET ; Descripción Estado 18. Evento 21
NOP
RET ; Descripción Estado 18. Evento 22
ME18_EV_8:
LCALL VENT_ONOFF
MOV ESTADO, #19
RET ; Acción 17
;-----
MQ_EVEN_19:
MOV A, EVENTO
RL A
MOV DPTR, #LIST EVEN_MQEV_19
JMP @A+DPTR
LIST_EVEN_MQEV_19:
RET ; Evento 0
NOP
RET ; Descripción Estado 19. Evento 1
NOP
RET ; Descripción Estado 19. Evento 2
NOP
RET ; Descripción Estado 19. Evento 3
NOP
RET ; Descripción Estado 19. Evento 4
NOP
RET ; Descripción Estado 19. Evento 5
NOP
RET ; Descripción Estado 19. Evento 6
NOP
RET ; Descripción Estado 19. Evento 7
NOP
RET ; Descripción Estado 19. Evento 8
NOP
AJMP ME19_EV_9 ; Descripción Estado 19. Evento 9
RET ; Descripción Estado 19. Evento 10
NOP
RET ; Descripción Estado 19. Evento 11
NOP
RET ; Descripción Estado 19. Evento 12
NOP
RET ; Descripción Estado 19. Evento 13
NOP
RET ; Descripción Estado 19. Evento 14
NOP
RET ; Descripción Estado 19. Evento 15
NOP
RET ; Descripción Estado 19. Evento 16
NOP
RET ; Descripción Estado 19. Evento 17
NOP
RET ; Descripción Estado 19. Evento 18
NOP
RET ; Descripción Estado 19. Evento 19
NOP
RET ; Descripción Estado 19. Evento 20
NOP
RET ; Descripción Estado 19. Evento 21
NOP
RET ; Descripción Estado 19. Evento 22
ME19_EV_9:
MOV PWM1, #0xFF ; Acción 17
MOV ESTADO, #20
RET
;-----
MQ_EVEN_20:
MOV A, EVENTO
RL A
MOV DPTR, #LIST EVEN_MQEV_20
JMP @A+DPTR

```

```

LIST_EVEN_MQEV_20:
    RET          ; Evento 0
    NOP
    RET          ; Descripción Estado 20. Evento 1
    NOP
    RET          ; Descripción Estado 20. Evento 2
    NOP
    RET          ; Descripción Estado 20. Evento 3
    NOP
    RET          ; Descripción Estado 20. Evento 4
    NOP
    RET          ; Descripción Estado 20. Evento 5
    NOP
    RET          ; Descripción Estado 20. Evento 6
    NOP
    RET          ; Descripción Estado 20. Evento 7
    NOP
    RET          ; Descripción Estado 20. Evento 8
    NOP
    RET          ; Descripción Estado 20. Evento 9
    NOP
    RET          ; Descripción Estado 20. Evento 10
    NOP
    RET          ; Descripción Estado 20. Evento 11
    NOP
    RET          ; Descripción Estado 20. Evento 12
    NOP
    RET          ; Descripción Estado 20. Evento 13
    NOP
    RET          ; Descripción Estado 20. Evento 14
    NOP
    RET          ; Descripción Estado 20. Evento 15
    NOP
    RET          ; Descripción Estado 20. Evento 16
    NOP
    RET          ; Descripción Estado 20. Evento 17
    NOP
    RET          ; Descripción Estado 20. Evento 18
    NOP
    RET          ; Descripción Estado 20. Evento 19
    NOP
    RET          ; Descripción Estado 20. Evento 20
    NOP
    AJMP     ME20_EV_21 ; Descripción Estado 20. Evento 21
    RET          ; Descripción Estado 20. Evento 22
ME20_EV_21:
    CLR          BMOV_BACK ; Acción 11
    CLR          SEM_ROJO   ; Acción 4
    MOV          ESTADO,#21
    RET

;-----
MQ_EVEN_21:
    SETB     TCON.4
    MOV      A, EVENTO
    RL       A
    MOV      DPTR, #LIST_EVEN_MQEV_21
    JMP      @A+DPTR
LIST_EVEN_MQEV_21:
    RET          ; Evento 0
    NOP
    RET          ; Descripción Estado 21. Evento 1
    NOP
    LJMP     ME21_EV_2 ; Descripción Estado 21. Evento 2
    RET          ; Descripción Estado 21. Evento 3
    NOP
    RET          ; Descripción Estado 21. Evento 4
    NOP
    RET          ; Descripción Estado 21. Evento 5
    NOP
    RET          ; Descripción Estado 21. Evento 6
    NOP

```

```

RET ; Descripción Estado 21. Evento 7
NOP
RET ; Descripción Estado 21. Evento 8
NOP
RET ; Descripción Estado 21. Evento 9
NOP
RET ; Descripción Estado 21. Evento 10
NOP
RET ; Descripción Estado 21. Evento 11
NOP
RET ; Descripción Estado 21. Evento 12
NOP
RET ; Descripción Estado 21. Evento 13
NOP
RET ; Descripción Estado 21. Evento 14
NOP
RET ; Descripción Estado 21. Evento 15
NOP
RET ; Descripción Estado 21. Evento 16
NOP
RET ; Descripción Estado 21. Evento 17
NOP
RET ; Descripción Estado 21. Evento 18
NOP
RET ; Descripción Estado 21. Evento 19
NOP
RET ; Descripción Estado 21. Evento 20
NOP
RET ; Descripción Estado 21. Evento 21
NOP
AJMP ME21_EV_22 ; Descripción Estado 21. Evento 22

ME21_EV_2:
SETB SEM_VERDE ; Acción 1
ACALL LED_MODO_ONOFF ; Acción 7
MOV ESTADO, #00
RET

ME21_EV_22:
SETB TCON.4
LCALL SEM_VERDE_ONOFF ; Acción 18
RET
;-----

LED_MODO_ONOFF: ;Acción 7
JB LED_LNOR, LED_MODONOR_OFF
JB LED_LINT, LED_MODOINT_OFF
JB FICHA_TIPO, ON_LINT
SETB LED_LNOR
RET

ON_LINT:
SETB LED_LINT
RET

LED_MODONOR_OFF:
CLR LED_LNOR
RET

LED_MODOINT_OFF:
CLR LED_LINT
RET
;-----

ROD_VER_DENTRO_ONOFF: ;Acción 12
JB RV_INTO, OFF_VDENTRO
CLR RV_OUTTO
SETB RV_INTO
JB FICHA_TIPO, LAVINT_PWM0
MOV PWM0, #0x80
RET

OFF_VDENTRO:
CLR RV_INTO
RET
;-----

ROD_VER_FUERA_ONOFF: ;Acción 13
JB RV_OUTTO, OFF_VFUERA

```

```

CLR      RV_INTO
SETB    RV_OUTTO
JB      FICHA_TIPO,LAVINT_PWM0
MOV     PWM0,#0x80
RET

OFF_VFUERA:
CLR      RV_OUTTO
RET

;-----
ROD_HOR_ABAJO_ONOFF:                      ;Acción 14
JB      RH_DOWNTO, OFF_HABAJO
CLR      RH_UPTO
SETB    RH_DOWNTO
JB      FICHA_TIPO,LAVINT_PWM0
MOV     PWM0,#0x80
RET

OFF_HABAJO:
CLR      RH_DOWNTO
RET

;-----
ROD_HOR_ARRIBA_ONOFF:                      ;Acción 15
JB      RH_UPTO, OFF_HARRIBA
CLR      RH_DOWNTO
SETB    RH_UPTO
JB      FICHA_TIPO,LAVINT_PWM0
MOV     PWM0,#0x80
RET

OFF_HARRIBA:
CLR      RH_UPTO
RET

;-----
LAVINT_PWM0:
MOV     PWM0,#0x00
RET

;-----
OFF_TODOS_MOV_ROD_HV:                      ;Acción 16
CLR      RV_INTO
CLR      RV_OUTTO
CLR      RH_DOWNTO
CLR      RH_UPTO
RET

;-----
VENT_ONOFF:                                ;Acción 17
JB      VENTS_ON,OFF_VENT
JB      FICHA_TIPO,VENTINT_PWM1
MOV     PWM1,#0x80
RET

VENTINT_PWM1:
MOV     PWM1,#0x26
RET

OFF_VENT:
MOV     PWM1,#0xFF
CLR      VENTS_ON
RET

;-----
SEM_VERDE_ONOFF:                           ;Acción 18
JB      SEM_VERDE, SEM_VERDE_OFF2
SETB    SEM_VERDE
RET

SEM_VERDE_OFF2:
CLR      SEM_VERDE
RET

;-----
TIMER0_ISR:
PUSH    ACC
PUSH    PSW
MOV     TL0, #0xB0          ; Carga parte baja (TL0) para 50 ms
MOV     TH0, #0x3C          ; Carga parte alta (TH0) para 50 ms
SETB    TICK_50ms          ; Activa la bandera de 50ms
DEC     COUNT_1             ; Decrementa el contador de 1s
MOV     ACC, COUNT_1        ; Mete en el ACC el valor del contador

```

```

JNZ    TERMINAR_ISR          ; 20 x 50 ms = 1 segundo
MOV    COUNT_1, #20           ; Reinicia el contador de 50 ms
SETB   TICK_1                ; Activa la bandera de 1s

TIME_4S:
DEC    COUNT_4                ; Decrementa el contador de 4s
MOV    ACC, COUNT_4           ; Mete en el acumulador el valor del cont.
JNZ    TIME_5S                ; 4 x 1 s = 4 segundos
MOV    COUNT_4, #4             ; Reinicia el contador de 1 segundo
SETB   TICK_4                ; Activa la bandera de 4s

TIME_5S:
DEC    COUNT_5                ; Decrementa el contador de 5s
MOV    ACC, COUNT_5           ; Mete en el acumulador el valor del cont.
JNZ    TERMINAR_ISR          ; 4s + 1s = 5 segundos
MOV    COUNT_5, #5             ; Reinicia el contador de 1 segundo
SETB   TICK_5                ; Activa la bandera de 5s

TIME_30S:
DEC    COUNT_30               ; Decrementa el contador de 30s
MOV    ACC, COUNT_30           ; Mete en el acumulador el valor del cont.
JNZ    TERMINAR_ISR          ; 6 x 5 s = 30 segundos
MOV    COUNT_30, #6             ; Reinicia el contador de 5 segundos
SETB   TICK_30                ; Activa la bandera de 30s

TIME_60S:
DEC    COUNT_60               ; Decrementa el contador de 60s
MOV    ACC, COUNT_60           ; Mete en el acc. el valor del contador
JNZ    TERMINAR_ISR          ; 2 x 30 s = 60 segundos
MOV    COUNT_60, #2             ; Reinicia el contador de 30 segundos
SETB   TICK_60                ; Activa la bandera de 60s

TERMINAR_ISR:
POP    PSW
POP    ACC
RETI                         ; Retorno de la interrupción
-----
END

```