

# Arquitectura de Computadores (27698)

## Práctica Final de Laboratorio

Departamento:  
**Tecnología Electrónica**

Titulación:  
**Grado en Informática de Gestión y Sistemas de Información**

2º Curso (1º Cuatrimestre)

Fecha: **17 de diciembre de 2024**

Grandes Orons, Asier  
Requies Ruiz, Jon

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Diagramas de estados y de flujo</b>	<b>3</b>
2.1. Diagrama de estados . . . . .	3
2.2. Diagramas de flujo . . . . .	8
2.2.1. Diagrama de flujo del Programa Principal . . . . .	8
2.2.2. Diagrama de flujo de los Timers . . . . .	9
2.2.3. Diagrama de flujo del ADC . . . . .	10
2.2.4. Diagrama de flujo de la Máquina de Eventos del estado 'Frotado' . . . . .	10
<b>3. Cálculos y comentarios</b>	<b>11</b>
3.1. Cálculo del ADC . . . . .	11
3.2. Cálculo del PWM . . . . .	12
3.3. Cálculo del Timer . . . . .	13
<b>4. Código</b>	<b>14</b>

# 1. Introducción

En este proyecto final de la asignatura de Arquitectura de Computadores se pretende realizar el software de control de un lavadero de coches utilizando el microcontrolador 80C552 de Philips.

La tarea a realizar es la programación de las diversas funciones que nos proporciona este microcontrolador. Tales como Leds, el Convertidor Analógico-Digital –*Analog-to-Digital Converter (ADC)*–, la Modulación por Anchura de Pulso –*Pulse Width Modulation (PWM)*– y el Timer0 y el Timer1.

Cabe destacar que todo será realizado con el ya mencionado microcontrolador 80C552 de Philips, que es un derivado del microcontrolador 80C51. Tanto el programa como la simulación serán realizados en el entorno de desarrollo Keil uVision2.

Inicialmente el proyecto fue trabajado en las horas de laboratorio, sobre las 2-3 primeras clases prácticas dedicadas al proyecto. A partir de este momento el grupo decidió ir trabajando individualmente en casa, siempre actualizando cada cambio realizado para poder seguir de manera organizada y estructurada el trabajo del otro. Cabe destacar que el trabajo en las clases de laboratorio ha sido fundamental hasta el último día, debido a que servía para poner ideas en común de manera más clara y resolver las dudas que iban apareciendo por ambas partes.

Una vez discutida la manera de trabajar se va a presentar la cronología del proyecto:

## 1. Laboratorios 1-2

Los primeros laboratorios fueron fundamentales para aprender a usar el entorno de Keil uVision2 y familiarizarse con todas las funcionalidades que ofrece. No obstante, antes de comenzar con el código era imprescindible tener un primer diagrama de estados en el que se basaría el proyecto. Las primeras lecturas del enunciado fueron realmente importantes para tener la idea general de lo que este proyecto iba a ser.

En estas fechas aún no se había empezado a trabajar en casa.

## 2. Laboratorios 3-5

A partir del tercer laboratorio se empezó a trabajar en casa. Lo primero que fue realizado fueron los timers. Desde el cálculo de valores iniciales hasta su implementación en el código. Dieron algún problema debido a no estar tan acostumbrados al entorno ni al lenguaje pero fueron solucionados con cierta velocidad.

Una vez hechos los timers, el foco se centró en la implementación de los primeros estados del diagrama y el estado final. Esto se debió a la simplicidad de los estados comparados con el resto. Al usar únicamente algún timer los pudimos realizar sin ningún problema.

Después de finalizar lo ya mencionado anteriormente, el siguiente paso fue el cálculo y la implementación del PWM. El cálculo fue realizado sin problemas siguiendo las fórmulas de la teoría, en cambio, el código no fue tan sencillo. Con el PWM pudimos acabar algún estado más.

Para finalizar, el grupo se puso a trabajar en el ADC. Esta parte del proyecto fue la que más problemas dio, sobre todo la correcta implementación para los estados de los rodillos. El cálculo de los valores digitales no tomó mucho tiempo, pero el código fue completamente diferente. Realizar subrutinas de espera, hacer correctamente la lógica, los saltos entre subrutinas, las condiciones... Fue un gran reto para el grupo y se aprendió mucho.

El último laboratorio fue la demostración del proyecto al profesor.

## 2. Diagramas de estados y de flujo

En el desarrollo de este tipo de sistemas, la correcta estructuración y organización de los procesos es fundamental para garantizar un control eficiente. Los diagramas de estados, tablas de eventos y diagramas de flujo son herramientas clave que permiten visualizar y estructurar el comportamiento lógico de un sistema.

### 2.1. Diagrama de estados

Se presentarán los diagramas de estados, que describen los posibles estados en los que se puede encontrar el sistema, los eventos que provocan las transiciones entre ellos y las acciones asociadas. Para cada estado relevante, se desarrollarán sus tablas de estados, tablas de eventos y tablas de acciones, que permiten detallar de forma precisa los cambios de estado y los eventos que los provocan.

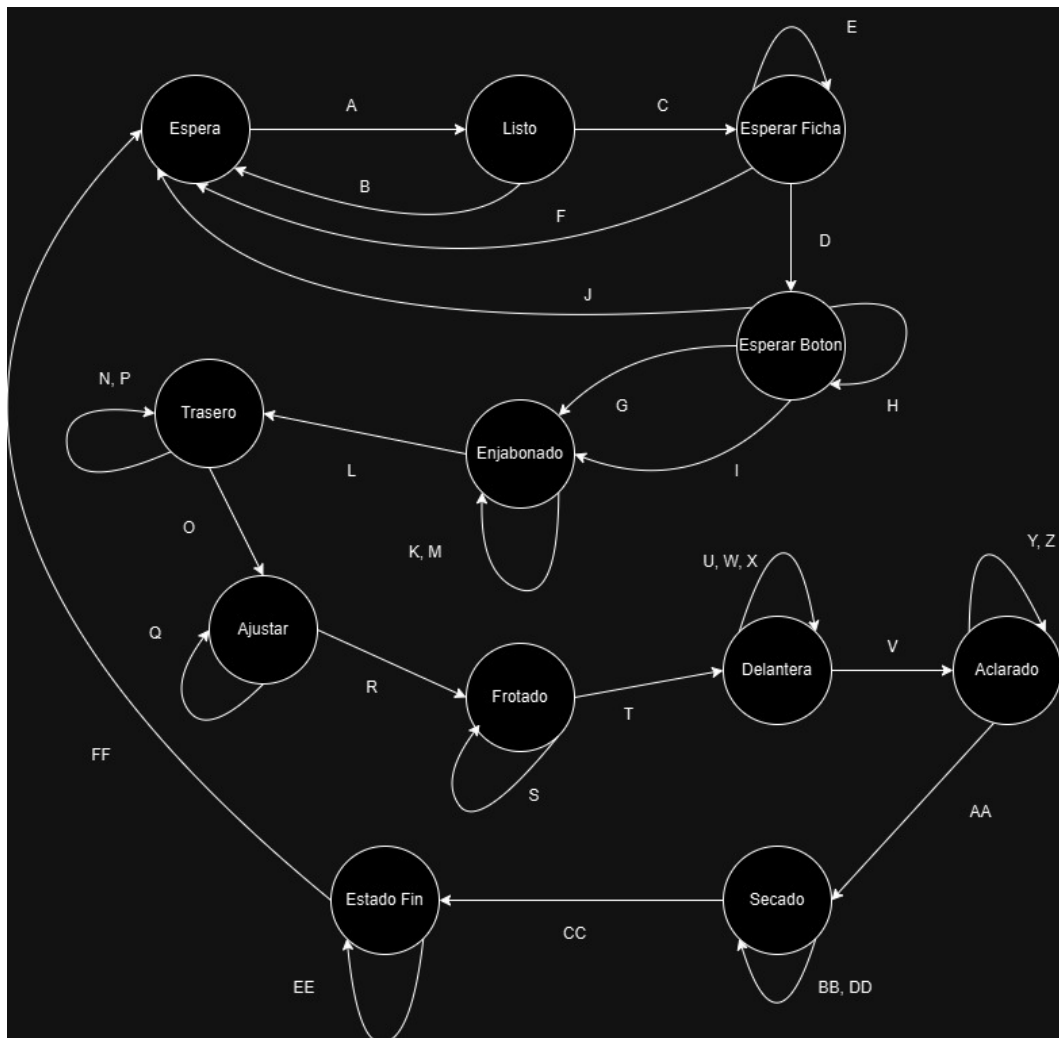


Figura 1: Diagrama de estados

NÚMERO	NOMBRE
0	ESPERA
1	LISTO
2	ESPERAR_FICHA
3	ESPERAR_BOTON
4	ENJABONADO
5	TRASERO
6	AJUSTAR
7	FROTADO
8	DELANTERO
9	ACLARADO
10	SECADO
11	ESTADO_FIN

Cuadro 1: Tabla de estados

<b>CÓDIGO</b>	<b>DESCRIPCIÓN</b>
A	HA PISADO LA PLACA (S_PLAT=1)
B	SE QUITA EL COCHE (S_PLAT=0)
C	ESTÁ 5 SEG EN LA PLACA (PASAN 5S)
D	HA ENTRADO FICHA (S_FICHA)
E	SE ACTIVA LA ALARMA (PASAN 30S)(HA SALTADO=0)
F	SE QUITA EL COCHE (S_PLAT=0)
G	LE DA AL BOTÓN (P_START)
H	SUENA LA ALARMA (PASAN 30S)
I	30 DESPUÉS DE ALARMA EMPIEZA (PASAN 30 OTRA VEZ)
J	SE QUITA EL COCHE (S_PLAT=0)
K	SENSOR DETECTA COCHE (S_CAR=1)
L	DESPLAZA 40CM (PASAN 4S)
M	CAMBIA EL ESTADO (FLAG_MOVER=0)
N	RODILLOS 80CM (FC_RV_CENTRO=1)
O	RODILLOS POS INICIAL (FC_RV_BORDE=1)
P	CAMBIA EL ESTADO (FLAG_RODILLOS=0)
Q	DETECTA EL COCHE (S_CAR=1)
R	CAMBIA EL ESTADO (FLAG_MOVER2=0)
S	NO DETECTA EL COCHE (S_CAR=0)
T	EL RODILLO HORIZONTAL VUELVE A SU POSICIÓN INICIAL (FC_RH_TOP=1)
U	RODILLOS 80CM (FC_RV_CENTRO=1)
V	RODILLOS POS INICIAL (FC_RV_BORDE=1)
W	CAMBIA EL ESTADO (FLAG_RODILLOS1=0)
X	DESPLAZA 40CM (PASAN 4S)
Y	DETECTA EL COCHE (S_CAR=1)
Z	DESPLAZA 40CM (PASAN 4S)
AA	CAMBIA EL ESTADO (FLAG_MOVER3=0)
BB	DETECTA EL COCHE (S_CAR=1)
CC	DETECTA FIN DE CARRERA (BPOS_START=1)
DD	CAMBIA EL ESTADO (FLAG_MOVER4=0)
EE	DETECTA FIN DE CARRERA (BPOS_START=1)
FF	SE QUITA EL COCHE (S_PLAT=0)

Cuadro 2: Tabla de eventos

NÚMERO ESTADO	CÓDIGO EVENTO EN ESTADO	CÓDIGO EVENTO
0	1	A
1	0	B
	2	C
2	0	F
	2	E
	3	D
3	0	J
	3	H
	4	I,G
4	4	K,M
	5	L
5	5	N,P
	6	O
6	6	Q
	7	R
7	7	S
	8	T
8	8	U,W,X
	9	V
9	9	Y,Z
	10	AA
10	10	BB,DD
	11	CC
11	11	EE
	0	FF

Cuadro 3: Tabla de eventos por estado

ESTADO	CÓDIGO	NOMBRE	NOMBRE EN EL CÓDIGO
0	0	CAMBIAR SEMÁFORO	ACCION1
1,2	1	QUITAR COCHE	QUITAR_COCHE
1	2	ESPERAR 5S ENCIMA DE LA PLACA	PLACA_ESPERAR
2	3	METER FICHA	METE_FICHA
2,3	4	PREPARAR ALARMA	PREPARAR_ALARMA
2,3	5	APAGAR ALARMA	APAGAR_ALARMA
4	6	ACTIVAR ELECTROVÁLVULAS	ACTIVAR_EV
4	7	DESPLAZAR EL PUENTE 40CM	MOVER_4S
4	8	MOVER EL PUENTE HACIA DELANTE	MOVER_COCHE
5,8	9	MOVER RODILLOS AFUERA	MOVER_RODILLOS_FUERA
5	10	RODILLOS VUELVEN A LA POSICIÓN INICIAL	RODILLOS_PRINCIPIO
5,8	11	MOVER RODILLOS ADENTRO	MOVER_RODILLOS_DENTRO
6	12	MOVER RODILLOS EN AJUSTADO	MOVER_RODILLOS_6
6	13	MOVER EL PUENTE HACIA ATRÁS	MOVER_COCHE2
7	14	MOVER RODILLOS EN FROTADO	EMPEZAR_RODILLOS
7	15	DETENER RODILLOS EN FROTADO	DETENER_RODILLOS
8	16	RODILLOS VUELVEN A LA POSICIÓN INICIAL	RODILLOS_PRINCIPIO2
8	17	DESPLAZAR EL PUENTE 40CM	MOVER_4S3
9	18	ACTIVAR ELECTROVÁLVULAS	ACTIVAR_EV2
9	19	DESPLAZAR EL PUENTE 40CM	MOVER_4S2
9	20	MOVER EL PUENTE HACIA DELANTE	MOVER_COCHE3
10	21	ACTIVAR SECADO	ACTIVAR_SECADO
10	22	FINALIZAR	FIN_DE_CARRERA
10	23	MOVER EL PUENTE HACIA ATRÁS	MOVER_COCHE4
11	24	SEMÁFORO INTERMITENTE	ACCION2

Cuadro 4: Tabla de acciones



## 2.2. Diagramas de flujo

En primer lugar, se describirá el flujo de control del programa principal, el cual establece la secuencia general de ejecución. Seguidamente, se elaborarán los diagramas de flujo de las subrutinas de interrupciones (como las generadas por el ADC y el Timer), que muestran el comportamiento del sistema ante eventos externos o internos que requieren atención inmediata. Finalmente, se presentará el flujo de control de la máquina de eventos del estado "Frotado", que es una de las partes críticas del sistema. Este flujo incluye el manejo de su generador de eventos, lo que permite gestionar de forma eficiente los cambios de estado.

En el caso de este proyecto, todos los estados presentados tienen su propio generador de eventos. Esta decisión de diseño permite que cada estado se encargue de controlar sus propios flujos, facilitando la depuración y mejorando la capacidad de respuesta del sistema.

### 2.2.1. Diagrama de flujo del Programa Principal

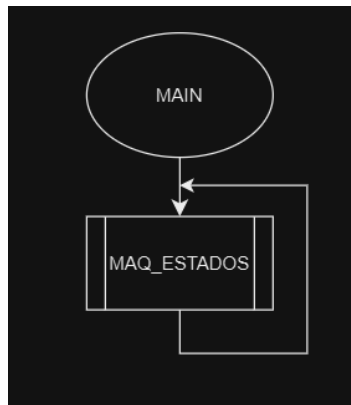


Figura 2: Diagrama de flujo del programa principal

### 2.2.2. Diagrama de flujo de los Timers

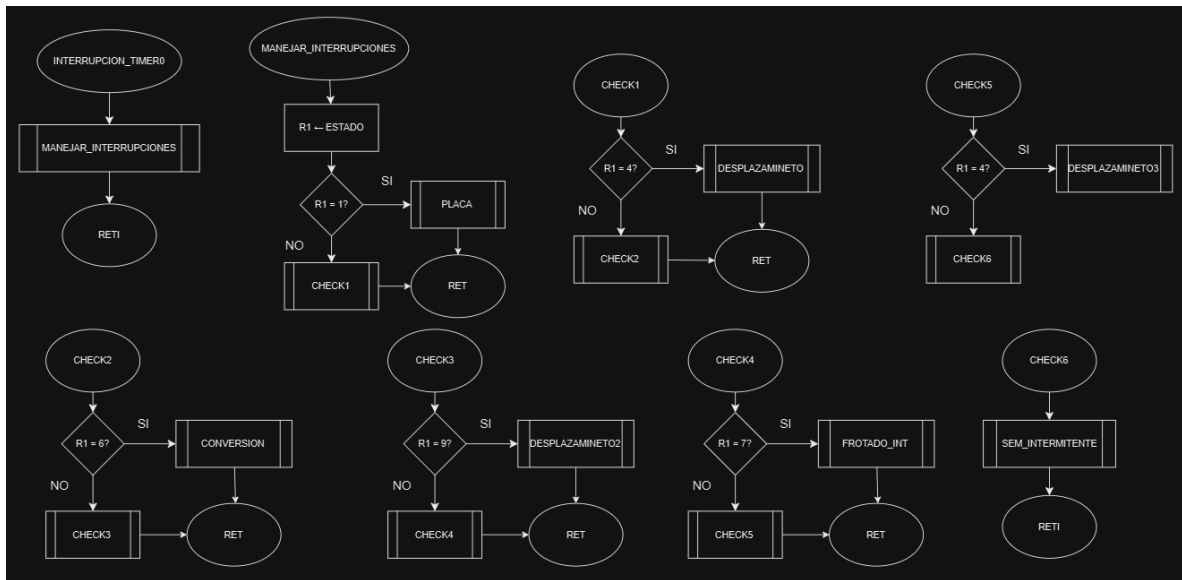


Figura 3: Diagrama de flujo del Timer0

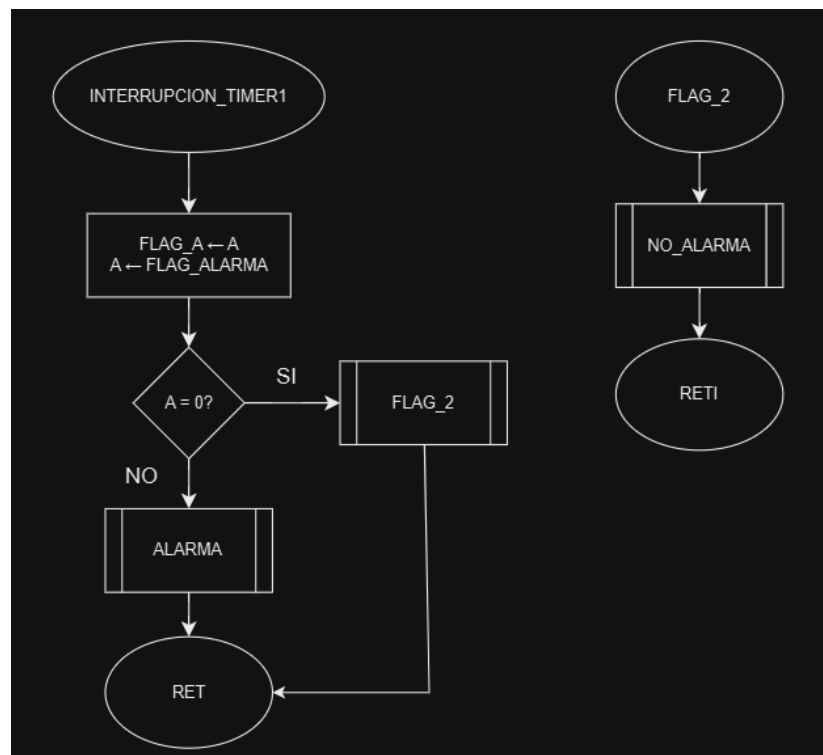


Figura 4: Diagrama de flujo del Timer1

### 2.2.3. Diagrama de flujo del ADC

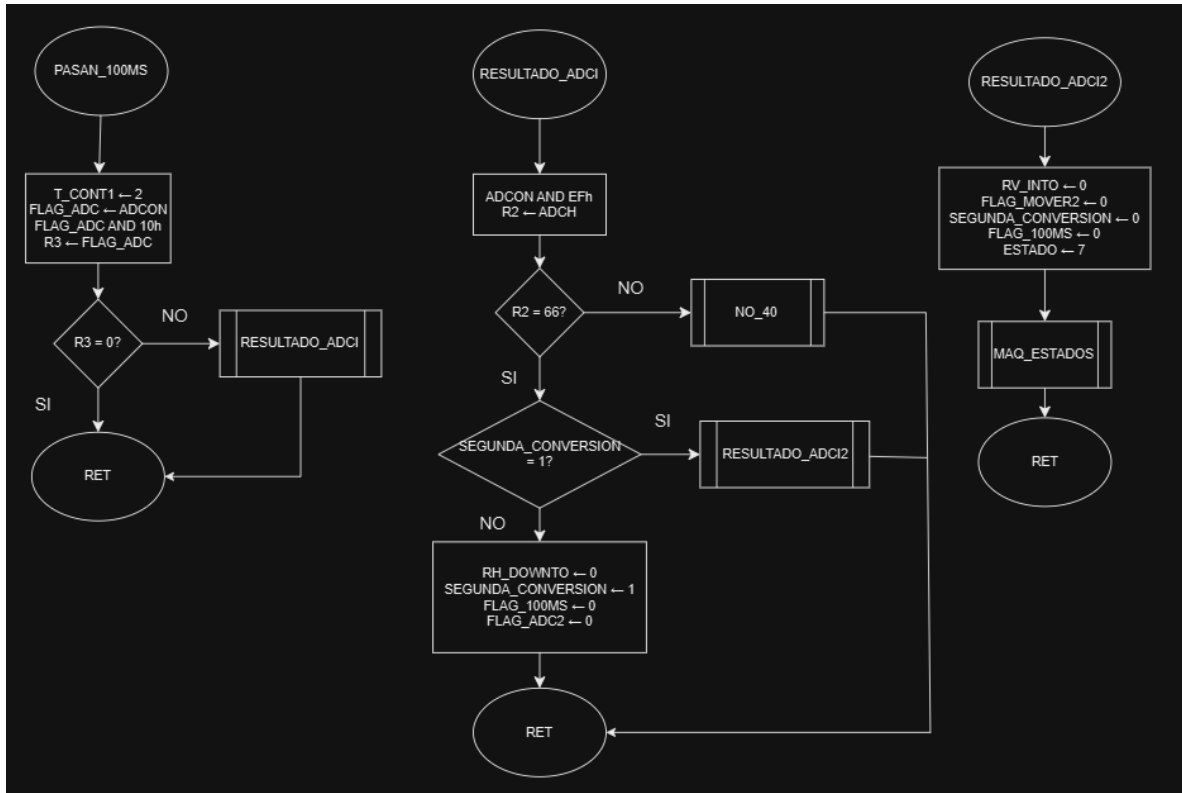


Figura 5: Diagrama de flujo del ADC

### 2.2.4. Diagrama de flujo de la Máquina de Eventos del estado 'Frotado'

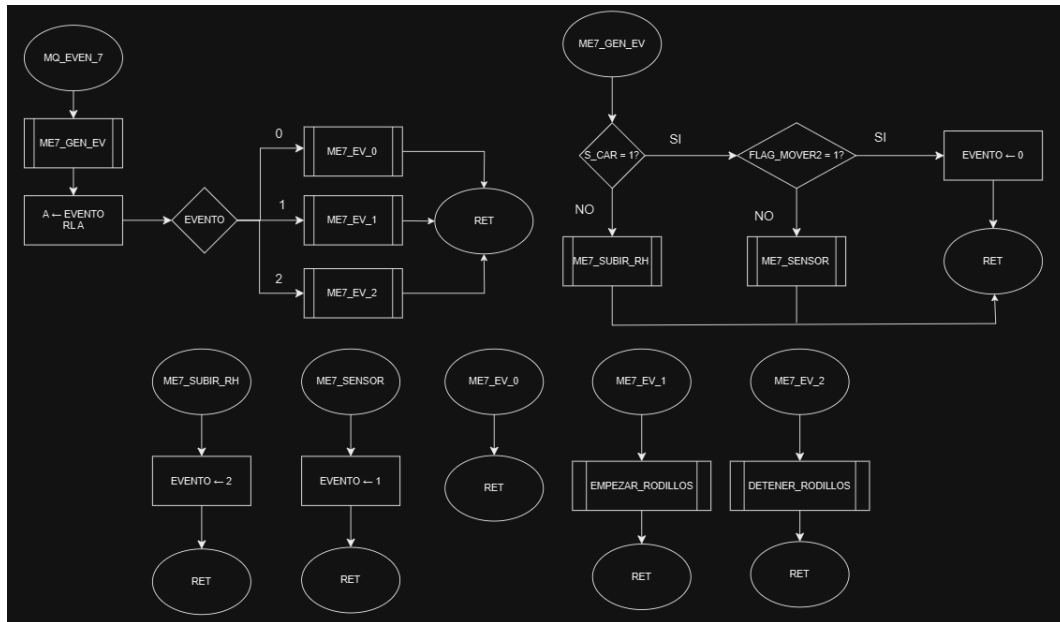


Figura 6: Diagrama de flujo de la máquina de eventos del estado 'Frotado'

### 3. Cálculos y comentarios

En la programación de microcontroladores, la implementación eficiente y precisa del ADC, PWM y los Timers es fundamental. Estos periféricos son herramientas esenciales, permitiendo convertir señales analógicas en digitales, generar señales de control precisas y medir o regular intervalos de tiempo.

#### 3.1. Cálculo del ADC

Los cálculos relacionados con la resolución, voltaje de referencia y escalado de las señales son cruciales para interpretar datos analógicos provenientes de sensores. Una mala configuración puede resultar en lecturas inexactas que comprometan el rendimiento del sistema.

Teniendo en cuenta la fórmula de obtención de valor digital:

$$D = \frac{V_{in} - V_{ref-}}{V_{ref+} - V_{ref-}} \cdot 2^n \quad (1)$$

Y los datos proporcionados por el enunciado del proyecto:

1. *Los sensores de distancia del sistema proporcionan una salida analógica de 50 mV/cm, que nunca superará los 5 V. Cuando la medida es nula, la tensión de salida es 0 V.*
2. *Las señales RST, STADC, VSS, AVSS y AV REF-, están conectadas a tierra.*
3. *Las señales /EA, VDD, AVDD y AVREF+, están conectadas a 5 V.*

Es posible calcular el voltaje recibido del sensor dependiendo de la distancia a la que se encuentre el coche. El enunciado dice que los rodillos deben estar entre 37.5 y 42.5 centímetros. Además, es necesario saber el voltaje cuando el sensor detecte 40 centímetros, debido a que es la distancia de la posición inicial de los rodillos.

$$37.5: 50 \text{ mV/cm} \times 37.5\text{cm} = 1.875\text{V}$$

$$40: 50 \text{ mV/cm} \times 40\text{cm} = 2\text{V}$$

$$42.5: 50 \text{ mV/cm} \times 42.5\text{cm} = 2.125\text{V}$$

Una vez calculado los voltajes es posible calcular los valores digitales para cada distancia detectada por el sensor.

$$D_{37,5cm} = \frac{1,875 - 0}{5 - 0} \cdot 2^8 = 96 \quad (2)$$

$$D_{40cm} = \frac{2 - 0}{5 - 0} \cdot 2^8 = 102 \quad (3)$$

$$D_{42,5cm} = \frac{2,125 - 0}{5 - 0} \cdot 2^8 = 108 \quad (4)$$

Una vez obtenidos los valores, ya es posible utilizarlos en la implementación de la estructura del ADC. Los voltajes se usan en la simulación, cuando sea necesario comprobar que los rodillos están a menos de 37.5 centímetros del coche usaremos un voltaje menor a 1.875. Así mismo, es posible probarlo con cualquiera de los tres resultados.

Los valores digitales en cambio, son usados en el código del programa. Insertándolos en registros para poder comprobar con el ADCH si es necesario mover los rodillos o no.

### 3.2. Cálculo del PWM

El diseño de ciclos de trabajo y frecuencias adecuados es clave para aplicaciones como control de motores, que es exactamente lo que pide el enunciado del proyecto. Controlar los motores para la velocidad de giro de los rodillos del frotado y la velocidad de giro de los ventiladores del secado.

El proyecto menciona la ficha introducida al sistema, dependiendo de ella realizaremos un lavado u otro, por lo tanto es esta moneda la que dice cómo se debe configurar el PWM:

1. *P0.3 -Entrada-. Tipo de lavado. '1' – intensivo, '0' – Normal / no hay ficha (FICHA\_TIPO).*
2. *PWM0 -Salida-. Canal control de la velocidad de giro de ambos juegos de rodillos, horizontal y vertical. En el lavado de normal, la señal PWM se establecerá al 50% En el lavado intenso, la señal PWM se establecerá al 100 %.*
3. *PWM1 -Salida-. Canal control de la velocidad de giro de los ventiladores de secado. En el lavado de normal, la señal PWM se establecerá al 50 %. En el lavado intenso, la señal PWM se establecerá al 85 %.*

Para realizar estos cálculos es necesario saber la frecuencia del microcontrolador y la de conmutación. Estos datos son mencionados en el enunciado:

1. *Los drivers de motores necesitan que la orden enviada por medio de los controladores PWM tenga una frecuencia de conmutación de 1 kHz.*
2. *El microcontrolador funciona a una frecuencia de 12 MHz.*

Usando la fórmula del cálculo del valor del PWMP (valor que se debe cargar en el registro para generar la señal PWM):

$$PWMP = \frac{f_{osc}}{2 \cdot f_{PWM} \cdot 255} - 1 \quad (5)$$

$f_{osc}$ : Frecuencia del microcontrolador  
 $f_{PWM}$ : Frecuencia de conmutación

Entonces el PWMP usado en el proyecto sería:

$$PWMP = \frac{12MHz}{2 \cdot 1kHz \cdot 255} - 1 = 22,52 \approx 23 \quad (6)$$

Además se debe calcular el ciclo de trabajo (C) de los PWM siguiendo esta fórmula:

$$C = (1 - Signal) \cdot 255 \quad (7)$$

En el caso del proyecto, se usan señales de 50 %, 85 % y 100 % para los motores de los rodillos y los secadores.

$$C_{50\%} = (1 - 0,5) \cdot 255 = 127,5 \approx 128 \quad (8)$$

$$C_{85\%} = (1 - 0,85) \cdot 255 = 38,25 \approx 38 \quad (9)$$

$$C_{100\%} = (1 - 1) \cdot 255 = 0 \quad (10)$$

Estos son los valores que se insertan en los registros PWM0 y PWM1, dependiendo del tipo de ficha introducida anteriormente.

### 3.3. Cálculo del Timer

Los temporizadores son indispensables para medir eventos, controlar interrupciones periódicas y sincronizar procesos. Su configuración depende de factores como el reloj del sistema y los prescaladores, que requieren cálculos precisos para garantizar un funcionamiento acorde a las especificaciones.

Para conseguir una correcta configuración de los timers es necesario realizar ciertos cálculos, para después introducir esos datos en la implementación del programa.

Cálculo del número de desbordes para un timer de 1 segundo, usando los datos mencionados en el enunciado del proyecto:

$$T_{interrupcion} = T_{osc} \cdot contador \quad (11)$$

$$1s = \frac{1}{12MHz} \cdot contador \quad (12)$$

$$contador = 12 \cdot 10^6 ciclos \quad (13)$$

Teniendo en cuenta el tamaño de los registros de un timer (16 bits) logramos obtener el número de desbordamientos necesarios para llegar al segundo:

$$n^\circ \text{ desbordamientos} = \frac{12 \cdot 10^6 ciclos}{2^{16}} = 183,11 \approx 183 \quad (14)$$

Este dato es el introducido en los contadores usados en el proyecto, de esta manera es posible saber cuando llega a 1 segundo. Para el timer de 100 milisegundos usado en el proyecto simplemente se divide entre 10 este resultado. Pero la configuración de los timers no son posibles únicamente con el número de desbordamientos, es necesario darle un valor inicial al timer.

$$T_{desbordamiento} = \frac{1}{183} = 5,46ms \quad (15)$$

$$T_{desbordamiento} = contador \cdot T_{osc} \quad (16)$$

$$5,46 = contador \cdot \frac{1}{12 \cdot 10^6} \quad (17)$$

$$contador = 65220 \quad (18)$$

$$Valorinicial = 65535 - 65220 = 316 = 013Ch \quad (19)$$

El dato calculado sería el dato introducido en el timer antes de iniciarlo. Con estos pasos realizados, el timer ya estaría listo para empezar.

Esto es el cálculo teórico de la configuración del timer, pero al hacer las pruebas en el entorno Keil uVision2 ha sido usado otro dato porque no era viable con 183 ni 18. El valor final usado ha sido 15 para el timer de 1 segundo y 2 para el timer de 100 milisegundos.

## 4. Código

```
1
2
3 ;=====;
4 ;VARIABLES GLOBALES
5 ESTADO EQU 0X20
6 EVENTO EQU 0X21
7 T_CONT1 EQU 0x22 ;BYTE DONDE SE GUARDA EL CONTADOR[1] USADO EN EL TIMER
8 T_CONT2 EQU 0x23 ;BYTE DONDE SE GUARDA EL CONTADOR[2] USADO EN EL TIMER
9
10 FLAG_PLACA EQU 0X26 ;BYTE PARA ESPERAR 5 SEG PLACA
11 FLAG_ALARMA EQU 0X27 ;FLAG PARA ESTADO 2 Y 3
12 FLAG_30 EQU 0X28 ;BYTE PARA ESPERAR 30 SEG ALARMA
13 FLAG_MOVER EQU 0X29 ;BYTE PARA MOVER EL COCHE2
14 FLAG_A EQU 0X30 ;BYTE PARA GUARDAR A
15 FLAG_AGUA EQU 0X31 ;BYTE PARA SABER SI AGUA HA SIDO ACTIVADO
16 FLAG_4S EQU 0X32 ;BYTE PARA SABER CUANDO EMPEZAR EL DESPLAZAMIENTO DE 4S
17 FLAG_SECADO EQU 0X33 ;BYTE PARA SABER SI SECADO HA SIDO ACTIVADO
18 FLAG_ADC EQU 0X34 ;BYTE PARA SABER SI ADC
19 FLAG_100MS EQU 0X35 ;BYTE PARA SABER SI LO HA ACTIVADO
20 FLAG_ADC2 EQU 0X36 ;BYTE PARA SABER SI LO HEMOS ACTICVADO
21 SEGUNDA_CONVERSION EQU 0X37 ;BYTE PARA PONER LOS RODILLOS VERTICALES EN SU SITIO
22 FLAG_RODILLOS EQU 0X38 ;BYTE PARA LOS RODILLOS DEL PRINCIPIO
23 FLAG_RODILLOS1 EQU 0X39 ;BYTE PARA LOS RODILLOS DEL PRINCIPIO
24 FLAG_MOVER2 EQU 0X40 ;BYTE PARA MOVER EL COCHE2
25 FLAG_MOVER3 EQU 0X41 ;BYTE PARA MOVER EL COCHE3
26 FLAG_MOVER4 EQU 0X42 ;BYTE PARA MOVER EL COCHE4
27 FLAG_4S3 EQU 0X43 ;BYTE PARA MOVER EL PUENTE 4S
28 HA_SALTADO EQU 0X44
29
30 ADCON EQU 0xC5 ;BYTE DE CONFIGURACION DE ADC
31 ADCH EQU 0xC6 ;BYTE DONDE SE GUARDA EL VALOR DE LECTURA DEL ADC
32 PWMP EQU 0xFE ;BYTE DEL PRESCALER DEL PWM
33 PWM0 EQU 0xFC ;BYTE DONDE SE GUARDA EL DUTY CICLE
34 PWM1 EQU 0xFD ;BYTE DONDE SE GUARDA EL DUTY CICLE
35 IEN0 EQU 0xA8 ;BYTE DE CONFIGURACION DE LAS INTERRUPCIONES
36
37 ;PUERTOS
38 SEM_VERDE EQU P0.0
39 SEM_ROJO EQU P0.1
40 S_PLAT EQU P0.2 ;SENSOR DE PLACA
41 FICHA_TIPO EQU P0.3 ;TIPO DE LAVADO
42 S_FICHA EQU P0.4 ;SENSOR DE FICHA
43 P_START EQU P0.5 ;BOTON DE EMPEZAR
44 AL_START EQU P0.6 ;ALARMA POR FALTA DE FICHA O COMIENZO MANUAL
45 BPOS_START EQU P0.7 ;POSICION INICIAL DEL PUENTE DE LAVADO
46
47 BMOV_FRONT EQU P1.0 ;MUEVE EL PUENTE DE LAVADO HACIA ADELANTE
48 BMOV_BACK EQU P1.1 ;MUEVE EL PUENTE DE LAVADO HACIA ATRAS
49 FC_RH_TOP EQU P1.2 ;SENSOR DE FIN DE CARRERA DEL RODILLO HORIZONTAL EN PARTE MAS
    ALTA
50 RH_UPTO EQU P1.3 ;ACTUADOR PARA MOVER HACIA ARRIBA EL RODILLO HORIZONTAL
51 RH_DOWNTO EQU P1.4 ;ACTUADOR PARA MOVER HACIA ABAJO EL RODILLO HORIZONTAL
52 FC_RV BORDE EQU P1.5 ;SENSOR DE FIN DE CARRERA DE LOS RODILLOS VERTICALES EN
    POSICION MAS EXTERIOR
53 RV_OUTTO EQU P1.6 ;ACTUADOR PARA MOVER HACIA AFUERA LOS RODILLOS VERTICALES
54 RV_INTTO EQU P1.7 ;ACTUADOR PARA MOVER HACIA ADENTRO LOS RODILLOS VERTICALES
55
56 EV_AGUA EQU P2.0 ;ACTIVA LA BOMBA DE AGUA DE LA MAQUINA DE LAVADO
57 EV_JABON EQU P2.1 ;ACTIVA LA BOMBA DE AGUA JABONADA
58 S_CAR EQU P2.3 ;SENSOR DE PRESENCIA
59 FC_RV_CENTRO EQU P2.4 ;SENSOR DE FIN DE CARRERA DE RODILLOS VERTICALES EN POSICION
    INTERNA
60 LED_LNOR EQU P2.5 ;LED ENCENDIDO PARA INDICAR 'LAVADO NORMAL'
61 LED_LINT EQU P2.6 ;LED ENCENDIDO PARA INDICAR 'LAVADO INTENSO'
62
```

```

63
64 ;=====INICIO=====;
65 ORG 0X0000
66     LJMP LAVADERO
67 ORG 0X007B
68 LAVADERO:
69     ACALL INICIALIZAR
70 MAIN:
71     ACALL MAQ_ESTADOS
72     AJMP     MAIN
73
74 ;=====INIT=====;
75 INICIALIZAR:
76     ;VALORES GLOBALES
77     MOV ESTADO, #0
78     MOV EVENTO, #0
79     MOV T_CONT1, #0
80     MOV T_CONT2, #0
81     MOV FLAG_PLACA, #0
82     MOV FLAG_ALARMA, #0
83     MOV FLAG_30, #0
84     MOV FLAG_AGUA, #0
85     MOV FLAG_4S, #0
86     MOV FLAG_SECADO, #0
87     MOV FLAG_ADC, #0
88     MOV FLAG_100MS, #0
89     MOV FLAG_ADC2, #0
90     MOV SEGUNDA_CONVERSION, #0
91     MOV FLAG_MOVER, #0
92     MOV FLAG_MOVER2, #0
93     MOV FLAG_MOVER3, #0
94     MOV FLAG_MOVER4, #0
95     MOV FLAG_4S3, #0
96     MOV HA_SALTADO, #0
97
98
99     MOV PWMP, #23 ;ACTIVA EL PWMP CON EL DUTY-CYCLE CORRESPONDIENTE
100
101     MOV P0, #0xBC
102     MOV P1, #0x24 ;CONFIGURAR PUERTOS DE ENTRADA Y PUERTOS DE SALIDA
103     MOV P2, #0x18
104
105     SETB SEM_VERDE ;ENCENDER EL SEMAFORO VERDE
106
107     RET
108 ;-----
109
110 MAQ_ESTADOS:
111     MOV A, ESTADO
112     RL A
113     MOV DPTR, #LISTA_EST
114     JMP @A+DPTR
115
116 LISTA_EST:
117     AJMP ESPERA ; Nombre Estado 0
118     AJMP LISTO ; Nombre Estado 1
119     AJMP ESPERAR_FICHA ; Nombre Estado 2
120     AJMP ESPERAR_BOTON ; Nombre Estado 3
121     AJMP ENJABONADO ; Nombre Estado 4
122     AJMP TRASERA ; Nombre Estado 5
123     AJMP AJUSTAR ; Nombre Estado 6
124     AJMP FROTADO ; Nombre Estado 7
125     AJMP DELANTERA; Nombre Estado 8
126     AJMP ACLARADO ; Nombre Estado 9
127     AJMP SECADO ; Nombre Estado 10
128     AJMP ESTADO_FIN ; Nombre Estado 11
129
130 ESPERA :

```



```

131     ACALL MQ_EVEN_0
132     RET
133 LISTO :
134     ACALL MQ_EVEN_1
135     RET
136 ESPERAR_FICHA :
137     ACALL MQ_EVEN_2
138     RET
139 ESPERAR_BOTON:
140     ACALL MQ_EVEN_3
141     RET
142 ENJABONADO:
143     ACALL MQ_EVEN_4
144     RET
145 TRASERA:
146     ACALL MQ_EVEN_5
147     RET
148 AJUSTAR:
149     ACALL MQ_EVEN_6
150     RET
151 FROTADO:
152     ACALL MQ_EVEN_7
153     RET
154 DELANTERA:
155     ACALL MQ_EVEN_8
156     RET
157 ACLARADO:
158     ACALL MQ_EVEN_9
159     RET
160 SECADO:
161     ACALL MQ_EVEN_10
162     RET
163 ESTADO_FIN :
164     ACALL MQ_EVEN_11
165     RET
166 ;-----
167 MQ_EVEN_0:
168     ACALL MEO_GEN_EV
169     MOV A, EVENTO
170     RL A
171     MOV DPTR, #LIST_EVEN_MQEV_0
172     JMP @A+DPTR
173 LIST_EVEN_MQEV_0:
174     AJMP MEO_EV_0 ; Evento 0
175     AJMP MEO_EV_1 ; SI DETECTA EL COCHE
176
177 MEO_EV_0:
178     RET
179 MEO_EV_1:
180     MOV ESTADO, #1
181     ACALL ACCION1 ;EL COCHE ESTA PUESTO
182     RET
183 MEO_GEN_EV:
184     JB S_PLAT, MEO_PLACA ;COMPROBAR SI SE HA PULSADO LA PLACA
185     MOV EVENTO, #0 ;SIEMPRE SE DA EL EVENTO 0 SI NO SE HA PULSADO LA PLACA
186     RET
187 MEO_PLACA:
188     MOV EVENTO, #1
189     RET
190 ;-----
191 MQ_EVEN_1:
192     ACALL ME1_GEN_EV
193     MOV A, EVENTO
194     RL A
195     MOV DPTR, #LIST_EVEN_MQEV_1
196     JMP @A+DPTR
197 LIST_EVEN_MQEV_1:
198     AJMP ME1_EV_0 ;Evento 0

```

```

199     AJMP ME1_EV_1           ;QUITAR EL COCHE
200     AJMP ME1_EV_2           ;PASAN 5 SEG Y SE AVANZA
201
202 ME1_EV_0:
203     RET
204 ME1_EV_1:
205     MOV ESTADO, #0
206     ACALL QUITAR_COCHE       ;HE QUITADO EL COCHE
207     RET
208 ME1_EV_2:
209     ACALL PLACA_ESPERAR      ;ESPERAR 5S
210     RET
211
212 ME1_GEN_EV:
213
214     JB S_PLAT, ME1_PLACAIN   ;SALTA SI ESTA EN PLACA
215     JNB S_PLAT, ME1_PLACAOOUT ;COMPROBAR SI SE HA DEJADO DE PULSAR LA PLACA
216
217     MOV EVENTO, #0           ;SIEMPRE SE DA EL EVENTO 0 SI NO SE HA PULSADO LA PLACA
218     RET
219 ME1_PLACAOOUT:
220     MOV EVENTO, #01
221     RET
222 ME1_PLACAIN:
223     MOV EVENTO, #02
224     RET
225
226 ;-----
227
228 MQ_EVEN_2:
229     ACALL ME2_GEN_EV
230     MOV A, EVENTO
231     RL A
232     MOV DPTR, #LIST_EVEN_MQEV_2
233     JMP @A+DPTR
234 LIST_EVEN_MQEV_2:
235     AJMP ME2_EV_0           ;Evento 0
236     AJMP ME2_EV_1           ;HA ENTRADO FICHA
237     AJMP ME2_EV_2           ;TIMER = 30S ALARMA
238     AJMP ME2_EV_3           ;SE QUITA EL COCHE
239 ME2_EV_0:
240     RET
241 ME2_EV_1:
242     ACALL METE_FICHA         ;APAGA LA ALARMA SI HA SONADO Y METE FICHA
243     MOV ESTADO, #3
244     RET
245 ME2_EV_2:
246     MOV FLAG_ALARMA, #1      ;FLAG PARA EL TIMER DE 30S, EN EL ESTADO2 SALTA ALARMA A LOS
                                30S ENTONCES LO PONGO A UNO,
247                                ;SI NO HICIESE FALTA SALTAR LA ALARMA PERO SI EL TIMER DE 30S LO PONGO
                                A 0
248     ACALL PREPARAR_ALARMA
249     RET
250 ME2_EV_3:
251     ACALL APAGAR_ALARMA      ;APAGA LA ALARMA SI HA SONADO SI QUITA EL COCHE
252     MOV ESTADO, #0
253     ACALL QUITAR_COCHE       ;HE QUITADO EL COCHE
254     RET
255
256 ME2_GEN_EV:
257     JB S_FICHA, ME2_FICHA
258     JNB S_PLAT, ME2_PLACAOOUT ;COMPROBAR SI SE HA DEJADO DE PULSAR LA PLACA
259     JNB HA_SALTADO, ME2_ALARMA ;PASAN 30 SEGS ME2_ALARMA
260     MOV EVENTO, #0           ;SIEMPRE SE DA EL EVENTO 0
261     RET
262 ME2_FICHA:
263     MOV EVENTO, #1
264     RET

```

```

265 ME2_ALARMA:
266     MOV EVENTO, #2
267     RET
268 ME2_PLACAOUT:
269     MOV EVENTO, #3
270     RET
271 ;-----
272
273 MQ_EVEN_3:
274     ACALL ME3_GEN_EV
275     MOV A, EVENTO
276     RL A
277     MOV DPTR, #LIST_EVEN_MQEV_3
278     JMP @A+DPTR
279 LIST_EVEN_MQEV_3:
280     AJMP ME3_EV_0           ;Evento 0
281     AJMP ME3_EV_1           ;LE DA AL BOTON
282     AJMP ME3_EV_2           ;PASAN 30S
283     AJMP ME2_EV_3           ;SE QUITA EL COCHE
284
285 ME3_EV_0:
286     RET
287 ME3_EV_1:
288     ACALL APAGAR_ALARMA     ;APAGA LA ALARMA SI HA SONADO SI QUITA EL COCHE
289     MOV ESTADO, #4
290     RET
291 ME3_EV_2:
292     ACALL PREPARAR_ALARMA
293     RET
294 ME3_EV_3:
295     ACALL APAGAR_ALARMA     ;APAGA LA ALARMA SI HA SONADO SI QUITA EL COCHE
296     MOV ESTADO, #0
297     ACALL QUITAR_COCHE       ;HE QUITADO EL COCHE
298     RET
299 ME3_GEN_EV:
300     JB P_START, ME3_BOTON    ;LE DA AL BOTON
301     JNB S_PLAT, ME3_PLACAOUT ;COMPROBAR SI SE HA DEJADO DE PULSAR LA PLACA
302     AJMP ME3_ALARMA          ;PASAN 30 SEGS ME3_ALARMA
303     MOV EVENTO, #0           ;SIEMPRE SE DA EL EVENTO 0
304     RET
305 ME3_BOTON:
306     MOV EVENTO, #1
307     RET
308 ME3_ALARMA:
309     MOV EVENTO, #2
310     RET
311 ME3_PLACAOUT:
312     MOV EVENTO, #3
313     RET
314
315 ;-----
316
317 MQ_EVEN_4:
318     ACALL ME4_GEN_EV
319     MOV A, EVENTO
320     RL A
321     MOV DPTR, #LIST_EVEN_MQEV_4
322     JMP @A+DPTR
323 LIST_EVEN_MQEV_4:
324     AJMP ME4_EV_0           ; Evento 0
325     AJMP ME4_EV_1           ; SENSOR QUE DETECTA COCHE
326     AJMP ME4_EV_2           ; DESPLAZA EL PUENTE 40 CM
327     AJMP ME4_EV_3           ; MUEVE EL COCHE Y APAGA EV
328 ME4_EV_0:
329     RET
330 ME4_EV_1:
331     AJMP ACTIVAR_EV
332     RET

```

```

333 ME4_EV_2:
334     AJMP MOVER_4S
335     RET
336 ME4_EV_3:
337     AJMP MOVER_COCHE
338     RET
339 ME4_GEN_EV:
340     JB S_CAR, ME4_SENSOR      ;EL SENSOR DETECTA EL COCHE
341     JB FLAG_4S, ME4_MOVER_4S  ;DESPLAZAR PUENTE 4S
342     JNB FLAG_MOVER, ME4_MOVER_COCHE ;MOVER EL COCHE
343     MOV EVENTO, #0           ;SIEMPRE SE DA EL EVENTO 0
344     RET
345 ME4_SENSOR:
346     MOV EVENTO, #1
347     RET
348 ME4_MOVER_4S:
349     MOV EVENTO, #2
350     RET
351 ME4_MOVER_COCHE:
352     MOV EVENTO, #3
353     RET
354     ;-----
355
356 MQ_EVEN_5:
357     ACALL ME5_GEN_EV
358     MOV A, EVENTO
359     RL A
360     MOV DPTR, #LIST_EVEN_MQEV_5
361     JMP @A+DPTR
362 LIST_EVEN_MQEV_5:
363     AJMP ME5_EV_0             ;NO HACE NADA
364     AJMP ME5_EV_1             ;CUANDO ESTAN A 80 CM LOS RODILLOS VAN HACIA FUERA
365     AJMP ME5_EV_2             ;DETECTA CUANDO ESTAN EN SU POSICION INICIAL
366     AJMP ME5_EV_3             ;MUEVE LOS RODILLOS HACIA AL CENTRO
367
368 ME5_EV_0:
369     RET
370 ME5_EV_1:
371     AJMP MOVER_RODILLOS_FUERA
372     RET
373 ME5_EV_2:
374     AJMP RODILLOS_PRINCIPIO
375     RET
376 ME5_EV_3:
377     AJMP MOVER_RODILLOS_DENTRO
378     RET
379 ME5_GEN_EV:
380     JB FC_RV_BORDE, ME5_RODILLOS_PRINCIPIO ;LOS RODILLOS VUELVEN A SU POSICION INICAL
381     JB FC_RV_CENTRO, ME5_RODILLOS_FUERA ;LOS RODILLOS ESTAN A 80 CM EN EL CENTRO
382     JNB FLAG_RODILLOS, ME5_RODILLOS_NADA ;MUEVE LOS RODILLOS HACIA EL CENTRO
383     MOV EVENTO, #0
384     RET
385 ME5_RODILLOS_FUERA:
386     MOV EVENTO, #1
387     RET
388 ME5_RODILLOS_PRINCIPIO:
389     MOV EVENTO, #2
390     RET
391 ME5_RODILLOS_NADA:
392     MOV EVENTO, #3
393     RET
394     ;-----
395 MQ_EVEN_6:
396     ACALL ME6_GEN_EV
397     MOV A, EVENTO
398     RL A
399     MOV DPTR, #LIST_EVEN_MQEV_6
400     JMP @A+DPTR

```

```

401 LIST_EVEN_MQEV_6:
402     AJMP ME6_EV_0           ;Evento 0
403     AJMP ME6_EV_1           ;COLOCA LOS RODILLOS A 40 CM
404     AJMP ME6_EV_2           ;MUEVE EL COCHE
405
406 ME6_EV_0:
407     RET
408 ME6_EV_1:
409     AJMP MOVER_RODILLOS_6
410     RET
411 ME6_EV_2:
412     AJMP MOVER_COCHE2
413     RET
414 ME6_GEN_EV:
415     JNB FLAG_MOVER2, ME6_MOVER ;MUEVE EL COCHE
416     JB S_CAR, ME6_SENSOR      ;EL SENSOR DETECTA EL COCHE
417     MOV EVENTO, #0           ;SIEMPRE SE DA EL EVENTO 0
418     RET
419 ME6_SENSOR:
420     MOV EVENTO, #1
421     RET
422 ME6_MOVER:
423     MOV EVENTO, #2
424     RET
425 ;-----
426 MQ_EVEN_7:
427     ACALL ME7_GEN_EV
428     MOV A, EVENTO
429     RL A
430     MOV DPTR, #LIST_EVEN_MQEV_7
431     JMP @A+DPTR
432 LIST_EVEN_MQEV_7:
433     AJMP ME7_EV_0           ; Evento 0
434     AJMP ME7_EV_1           ; MUEVE LOS RODILLOS
435     AJMP ME7_EV_2           ; PONE EL RODILLO HORIZONTAL EN SU POSICION
436
437 ME7_EV_0:
438     RET
439 ME7_EV_1:
440     AJMP EMPEZAR_RODILLOS
441     RET
442 ME7_EV_2:
443     AJMP DETENER_RODILLOS
444     RET
445 ME7_GEN_EV:
446     JNB S_CAR, ME7_SUBIR_RH ;DEJA DE DETECTAR EL COCHE
447     JNB FLAG_MOVER2, ME7_SENSOR ;MUEVE LOS RODILLOS Y LOS AJUSTA
448     MOV EVENTO, #0           ;SIEMPRE SE DA EL EVENTO 0
449     RET
450 ME7_SENSOR:
451     MOV EVENTO, #1
452     RET
453 ME7_SUBIR_RH:
454     MOV EVENTO, #2
455     RET
456 ;-----
457 MQ_EVEN_8:
458     ACALL ME8_GEN_EV
459     MOV A, EVENTO
460     RL A
461     MOV DPTR, #LIST_EVEN_MQEV_8
462     JMP @A+DPTR
463 LIST_EVEN_MQEV_8:
464     AJMP ME8_EV_0           ; NO HACE NADA
465     AJMP ME8_EV_1           ; CUANDO ESTAN A 80 CM LOS RODILLOS VAN HACIA FUERA
466     AJMP ME8_EV_2           ; DETECTA CUANDO ESTAN EN SU POSICION INICIAL
467     AJMP ME8_EV_3           ; MUEVE LOS RODILLOS HACIA AL CENTRO
468     AJMP ME8_EV_4           ; MUEVE EL PUENTE 4S

```

```

469
470 ME8_EV_0:
471     RET
472 ME8_EV_1:
473     AJMP MOVER_RODILLOS_FUERA
474     RET
475 ME8_EV_2:
476     AJMP RODILLOS_PRINCIPIO2
477     RET
478 ME8_EV_3:
479     AJMP MOVER_RODILLOS_DENTRO
480     RET
481 ME8_EV_4:
482     AJMP MOVER_4S3
483     RET
484 ME8_GEN_EV:
485     JNB FLAG_4S3, ME8_MOVER_COCHE3      ;MUEVE EL PUENTE 4S
486     JB FC_RV_BORDE, ME8_RODILLOS_PRINCIPIO ;DETECTA CUANDO ESTAN EN SU POS INICIAL
487     JB FC_RV_CENTRO, ME8_RODILLOS_FUERA ;DETECTA CUANDO ESTAN A 80 CM
488     JNB FLAG_RODILLOS1, ME8_RODILLOS_NADA ;MUEVE LOS DRODILLOS AL CENRTO
489     MOV EVENTO, #0
490     RET
491 ME8_RODILLOS_FUERA:
492     MOV EVENTO, #1
493     RET
494 ME8_RODILLOS_PRINCIPIO:
495     MOV EVENTO, #2
496     RET
497 ME8_RODILLOS_NADA:
498     MOV EVENTO, #3
499     RET
500 ME8_MOVER_COCHE3:
501     MOV EVENTO, #4
502     RET
503
504 ;-----
505 MQ_EVEN_9:
506     ACALL ME9_GEN_EV
507     MOV A, EVENTO
508     RL A
509     MOV DPTR, #LIST_EVEN_MQEV_9
510     JMP @A+DPTR
511 LIST_EVEN_MQEV_9:
512     AJMP ME9_EV_0      ; Evento 0
513     AJMP ME9_EV_1      ; SENSOR QUE DETECTA COCHE
514     AJMP ME9_EV_2      ; DESPLAZA EL PUENTE 40 CM
515     AJMP ME9_EV_3      ; MOVER COCHE2
516
517 ME9_EV_0:
518     RET
519 ME9_EV_1:
520     AJMP ACTIVAR_EV2
521     RET
522 ME9_EV_2:
523     AJMP MOVER_4S2
524     RET
525 ME9_EV_3:
526     AJMP MOVER_COCHE3
527     RET
528 ME9_GEN_EV:
529     JB S_CAR, ME9_SENSOR      ;EL SENSOR DETECTA EL COCHE
530     JB FLAG_4S, ME9_MOVER_4S  ;DETECTA CUANDO TIENE QUE MOVER EL PUENTE 40 CM
531     JNB FLAG_MOVER3, ME9_MOVER ;MUEVE EL COCHE
532     MOV EVENTO, #0           ;SIEMPRE SE DA EL EVENTO 0 Y MUEVE EL COCHE
533     RET
534 ME9_SENSOR:
535     MOV EVENTO, #1
536     RET

```

```

537 ME9_MOVER_4S:
538     MOV EVENTO, #2
539     RET
540 ME9_MOVER:
541     MOV EVENTO, #3
542     RET
543 ;-----
544 MQ_EVEN_10:
545     ACALL ME10_GEN_EV
546     MOV A, EVENTO
547     RL A
548     MOV DPTR, #LIST_EVEN_MQEV_10
549     JMP @A+DPTR
550
551 LIST_EVEN_MQEV_10:
552     AJMP ME10_EV_0           ;NO HACE NADA
553     AJMP ME10_EV_1           ;ACTIVAR LOS VENTILADORES PARA SECADO
554     AJMP ME10_EV_2           ;HA LLEGADO AL FIN DE CARRERA
555     AJMP ME10_EV_3           ;MOVER_COCHE3
556
557 ME10_EV_0:
558     RET
559
560 ME10_EV_1:
561     AJMP ACTIVAR_SECADO
562     RET
563
564 ME10_EV_2:
565     AJMP FIN_DE_CARRERA
566     RET
567 ME10_EV_3:
568     AJMP MOVER_COCHE4
569     RET
570
571 ME10_GEN_EV:
572     JB S_CAR, ME10_SENSOR    ;EL SENSOR DETECTA EL COCHE
573     JB BPOS_START, ME10_START ;DETECTA EL FIN DE CARRERA
574     JNB FLAG_MOVER4, ME10_MOVER ;DETECTA CUANDO HAY QUE MOVER EL COCHE
575     MOV EVENTO, #0           ;SIEMPRE SE DA EL EVENTO 0
576     RET
577 ME10_SENSOR:
578     MOV EVENTO, #1
579     RET
580 ME10_START:
581     MOV EVENTO, #2
582     RET
583 ME10_MOVER:
584     MOV EVENTO, #3
585     RET
586 ;-----
587 MQ_EVEN_11:
588     ACALL ME11_GEN_EV
589     MOV A, EVENTO
590     RL A
591     MOV DPTR, #LIST_EVEN_MQEV_11
592     JMP @A+DPTR
593 LIST_EVEN_MQEV_11:
594     AJMP ME11_EV_0           ;Evento 0
595     AJMP ME11_EV_1           ;PONE EL SEMAFORO INTERMITENTE
596     AJMP ME11_EV_2           ;SE QUITA EL COHE
597 ME11_EV_0:
598     RET
599 ME11_EV_1:
600     ACALL ACCION2
601     RET
602 ME11_EV_2:
603     SETB SEM_VERDE
604     MOV ESTADO, #0

```

```

605     RET
606 ME11_GEN_EV:
607     JB BPOS_START, ME11_SEM_INTERMITENTE    ;DETECTA CUANDO PONER EL SEMAFORO VERDE
        INTERMITENTE
608     JNB S_PLAT, ME11_PLACAOUT                ;DETCETA CUANDO SE VA EL COCHE
609     MOV EVENTO, #0                          ;SIEMPRE SE DA EL EVENTO 0
610     RET
611 ME11_PLACAOUT:
612     MOV EVENTO, #2
613     RET
614 ME11_SEM_INTERMITENTE:
615     MOV EVENTO, #1
616     RET
617
618 ;-----
619
620
621 ACCION1:
622     CLR SEM_VERDE
623     SETB SEM_ROJO
624     RET
625
626 ;=====ESTADO_1=====
627 QUITAR_COCHE:
628     CLR SEM_ROJO
629     SETB SEM_VERDE
630     RET
631
632 PLACA_ESPERAR:
633     JNB FLAG_PLACA, PRIMERA
634     JNB S_PLAT, PLACA_OUT
635     MOV A, T_CONT2
636     JZ PASAN5S
637     RET
638 PRIMERA:
639     ACALL ENCENDER_TIMER0
640     MOV TH0, #01H    ;HEMOS HECHO EL CALCULO DE CICLOS DE RELOJ PARA 1S USANDO LA
        FRECUENCIA DEL ENUNCIADO Y NOS DA QUE TENEMOS QUE INICIALIZAR EL TIMER0 EN 316
641     MOV TL0, #3CH
642     MOV T_CONT1, #15    ;CADA 15 OVERFLOWS PASA 1S
643     MOV T_CONT2, #5
644     SETB FLAG_PLACA
645     JNB S_PLAT, PLACA_OUT
646     RET
647 PLACA_OUT:
648     LJMP ME1_PLACAOUT
649     RET
650 PASAN5S:
651     ACALL APAGAR_TIMER0
652     MOV ESTADO, #2
653     RET
654 ;=====ESTADO_2 Y ESTADO_3=====
655
656 PREPARAR_ALARMA:
657     MOV A, ESTADO
658     CJNE A, #4, ESTADO_ESTADO
659     RET
660
661 SALTADO:
662     RET
663
664 ESTADO_ESTADO:
665     ACALL ESTADO_ALARMA
666     MOV C, FLAG_30
667     JNC PRIMERA1
668     RET
669
670 ESTADO_ALARMA:

```



```

671     JNB AL_START, NO_SONADO
672     JB AL_START, SONADO
673
674 NO_SONADO:
675     MOV FLAG_ALARMA, #1 ;FLAG PARA EL TIMER DE 30S, EN EL ESTADO2 SALTA ALARMA A LOS 30S
        ENTONCES LO PONGO A UNO,
676         ;SI NO HICIESE FALTA SALTAR LA ALARMA PERO SI EL TIMER DE 30S LO PONGO A 0
677     RET
678 SONADO:
679     MOV FLAG_ALARMA, #0 ;FLAG PARA EL TIMER DE 30S, EN EL ESTADO2 SALTA ALARMA A LOS 30S
        ENTONCES LO PONGO A UNO,
680         ;SI NO HICIESE FALTA SALTAR LA ALARMA PERO SI EL TIMER DE 30S LO PONGO A 0
681     RET
682
683 PRIMERA1:
684     ACALL ENCENDER_TIMER1
685     MOV TH1, #01H ;HEMOS HECHO EL CALCULO DE CICLOS DE RELOJ PARA 1S USANDO LA
        FRECUENCIA DEL ENUNCIADO Y NOS DA QUE TENEMOS QUE INICIALIZAR EL TIMER0 EN 316
686     MOV TL1, #3CH
687     MOV T_CONT1, #15 ;CADA 15 OVERFLOWS PASA 1S
688     MOV T_CONT2, #30
689     SETB FLAG_30
690     SETB HA_SALTADO
691     RET
692
693 PASAN30S:
694     ACALL APAGAR_ALARMA ;APAGA LA ALARMA SI HA SONADO SI QUITA EL COCHE
695     MOV ESTADO, #4
696     RET
697
698 METE_FICHA:
699     ACALL APAGAR_ALARMA
700     MOV C, FICHA_TIPO ;MIRA TIPO_FICHA
701     JNC LAVADO_NORMAL ;SI 0 LAVADO NORMAL
702     AJMP LAVADO_INTENSO ;SI 1 LAVADO INTENSO
703     RET
704 APAGAR_ALARMA:
705     CLR AL_START
706     RET
707
708 LAVADO_NORMAL:
709     SETB LED_LNOR
710     RET
711 LAVADO_INTENSO:
712     SETB LED_LINT
713     RET
714
715 ;=====ESTADO_4=====
716
717 MOVER_COCHE:
718     SETB FLAG_MOVER
719     JNB FLAG_AGUA, PRIMERA_ACTIVACION
720     CLR BMOV_FRONT
721     CLR EV_AGUA
722     CLR EV_JABON
723     SETB FLAG_4S
724     RET
725
726 PRIMERA_ACTIVACION:
727     SETB BMOV_FRONT
728     RET
729
730 ACTIVAR_EV:
731     CLR FLAG_MOVER
732     SETB EV_AGUA
733     SETB EV_JABON
734     SETB FLAG_AGUA
735     RET

```

```

736
737 MOVER_4S:
738     JNB BMOV_FRONT, PRIMERA_4S
739     RET
740
741
742
743 PRIMERA_4S:
744     ACALL ENCENDER_TIMER0
745     MOV TH0, #01H           ;HEMOS HECHO EL CALCULO DE CICLOS DE RELOJ PARA 1S USANDO LA
746                             FRECUENCIA DEL ENUNCIADO Y NOS DA QUE TENEMOS QUE INICIALIZAR EL TIMER0 EN 316
747     MOV TL0, #3CH
748     MOV T_CONT1, #15        ;CADA 15 OVERFLOWS PASA 1S
749     MOV T_CONT2, #4
750     SETB BMOV_FRONT
751     RET
752
753 PASAN_4S:
754     ACALL APAGAR_TIMER0
755     CLR BMOV_FRONT
756     CLR FLAG_4S
757     CLR FLAG_AGUA
758     MOV ESTADO, #5
759     RET
760     ;=====ESTADO_5=====
761
762
763 MOVER_RODILLOS_DENTRO:
764     MOV R1, ESTADO           ; Carga el valor de ESTADO en R1
765     CJNE R1, #5, C1          ; Compara con el valor 1, si no es igual salta a C1
766     SETB FLAG_RODILLOS
767     JNB FICHA_TIPO, PWM0_NORMAL_FLAG
768     JB FICHA_TIPO, PWM0_INTENSO_FLAG
769     RET
770
771 C1:
772     SETB FLAG_RODILLOS1
773     JNB FICHA_TIPO, PWM0_NORMAL_FLAG
774     JB FICHA_TIPO, PWM0_INTENSO_FLAG
775     RET
776
777 PWM0_NORMAL_FLAG:
778     AJMP PWM0_NORMAL
779     RET
780
781 PWM0_INTENSO_FLAG:
782     AJMP PWM0_INTENSO
783     RET
784
785 MOVIMIENTO_DENTRO:
786     ACALL ENCENDER_PWMP
787     SETB RV_INT0
788     RET
789
790
791
792 MOVER_RODILLOS_FUERA:
793     CLR RV_INT0
794     SETB RV_OUTTO
795     RET
796
797
798 RODILLOS_PRINCIPIO:
799     CLR FC_RV_CENTRO
800     MOV ESTADO, #6
801     RET
802     ;=====ESTADO_6=====

```

```

803 MOVER_COCHE2:
804     SETB FLAG_MOVER2
805     SETB BMOV_BACK
806     RET
807
808
809 MOVER_RODILLOS_6:
810     JB SEGUNDA_CONVERSION, PONER_RODILLOS_VE
811     CLR BMOV_BACK
812     ACALL TIMER_100MS
813     ACALL ENCENDER_ADC
814     SETB RH_DOWNTO
815     RET
816
817 PONER_RODILLOS_VE:
818     ACALL TIMER_100MS
819     ACALL ENCENDER_ADC
820     SETB RV_INT0
821     RET
822
823 TIMER_100MS:
824     JNB FLAG_100MS, PRIMERA_100MS
825     RET
826
827 PRIMERA_100MS:
828     ACALL ENCENDER_TIMER0
829     MOV TH0, #01H ;HEMOS HECHO EL CALCULO DE CICLOS DE RELOJ PARA 1S USANDO LA
830                     FRECUENCIA DEL ENUNCIADO Y NOS DA QUE TENEMOS QUE INICIALIZAR EL TIMER0 EN 316
831     MOV TL0, #3CH
832     MOV T_CONTR1, #2 ;CADA 2 OVERFLOWS PASA 100MS
833     SETB FLAG_100MS
834     RET
835
836 PASAN_100MS:
837     MOV T_CONTR1, #2 ;CADA 2 OVERFLOWS PASA 100MS
838     MOV FLAG_ADC, ADCON
839     ANL FLAG_ADC, #0X10
840     MOV R3, FLAG_ADC
841     CJNE R3, #0, RESULTADO_ADCI
842     RET
843
844 RESULTADO_ADCI:
845     ANL ADCON, #0XEF ;CLR ADCI PARA Q PUEDA VOLVER A CONVERTIR
846     MOV R2, ADCH
847     CJNE R2, #0x66, NO_40
848     JB SEGUNDA_CONVERSION, RESULTADO_ADCI2
849     CLR RH_DOWNTO
850     SETB SEGUNDA_CONVERSION
851     CLR FLAG_100MS
852     CLR FLAG_ADC2
853     RET
854
855 RESULTADO_ADCI2:
856     CLR RV_INT0
857     CLR FLAG_MOVER2
858     CLR SEGUNDA_CONVERSION
859     CLR FLAG_100MS
860     MOV ESTADO, #07
861     LJMP MAQ_ESTADOS
862     RET
863
864 NO_40:
865     ORL ADCON, #0x08 ;VUELVE A INICIAR LA CONVERSION
866     SETB TR0
867     RET
868 ;=====ESTADO_7=====
869 EMPEZAR_RODILLOS:
870     SETB BMOV_BACK

```

```

870     MOV FLAG_ADC, ADCON
871     ANL FLAG_ADC, #0X10
872     MOV R3, FLAG_ADC
873     CJNE R3, #0, PRIMERA_CONVERSION
874     ACALL ENCENDER_ADC ;ENCENDER EL ADC UNA SOLA VEZ
875     ACALL TIMER_100MS ;ESPERAR 100ms
876     RET
877
878 PRIMERA_CONVERSION:
879     JB SEGUNDA_CONVERSION, SEGUNDA_CONVERSION2
880     MOV R5, ADCH ;LEER VALOR DEL ADC
881     SETB SEGUNDA_CONVERSION
882     CLR FLAG_ADC2
883     ANL ADCON, #0XEF ;CLR ADCI PARA Q PUEDA VOLVER A CONVERTIR
884     ACALL ENCENDER_ADC ;ENCENDER EL ADC UNA SOLA VEZ
885     RET
886
887 SEGUNDA_CONVERSION2:
888     MOV R6, ADCH ;LEER VALOR DEL ADC
889     ANL ADCON, #0XEF ;CLR ADCI PARA Q PUEDA VOLVER A CONVERTIR
890     RET
891
892 MOVIMIENTO_RH:
893     MOV A, #108 ;CARGAR LIMITE SUPERIOR
894     SUBB A, R5
895     JNC MIRAR_ABAJO ;SI ES POSITIVO, ESTA DENTRO DEL LIMITE SUPERIOR
896     ACALL MOVER_ABAJO ;MOVER HACIA ABAJO SI NO CUMPLE CONDICION
897     RET ;REGRESAR DESPUES DE MOVER
898
899 MIRAR_ABAJO:
900     MOV A, R5 ;CARGAR LIMITE INFERIOR
901     SUBB A, #96
902     JC MOVER_ARRIBA ;SI ES NEGATIVO, ESTA FUERA DEL LIMITE INFERIOR
903     RET ;REGRESAR SI NO HAY MOVIMIENTO NECESARIO
904
905 MOVER_ARRIBA:
906     SETB RH_UPTO ;MOVER HACIA ARRIBA
907     CLR RH_DWNT0
908     RET ;REGRESAR DESPUES DE MOVER
909
910 MOVER_ABAJO:
911     SETB RH_DWNT0 ;MOVER HACIA ABAJO
912     CLR RH_UPTO
913     RET ;REGRESAR DESPUES DE MOVER
914
915 MOVIMIENTO_RV:
916     MOV A, #108 ;CARGAR LIMITE SUPERIOR
917     SUBB A, R6
918     JNC MIRAR_DENTRO ;SI ES POSITIVO, ESTA DENTRO DEL LIMITE SUPERIOR
919     ACALL MOVER_DENTRO ;MOVER HACIA ADENTRO SI NO CUMPLE CONDICION
920     RET ;REGRESAR DESPUES DE MOVER
921
922 MIRAR_DENTRO:
923     MOV A, ADCH ;CARGAR VALOR
924     SUBB A, #96 ;RESTARLE AL ADCH EL LIMITE INFERIOR
925     JC MOVER_FUERA ;SI ES NEGATIVO, ESTA FUERA DEL LIMITE INFERIOR
926     RET ;REGRESAR SI NO HAY MOVIMIENTO NECESARIO
927
928 MOVER_FUERA:
929     SETB RV_OUTTO ;MOVER HACIA AFUERA
930     CLR RV_INT0
931     RET ;REGRESAR DESPUES DE MOVER
932
933 MOVER_DENTRO:
934     SETB RV_INT0 ;MOVER HACIA ADENTRO
935     CLR RV_OUTTO
936     RET ;REGRESAR DESPUES DE MOVER
937

```

```

938 PASAN_100MS_F:
939     CLR SEGUNDA_CONVERSION
940     MOV T_CONT1, #2      ;CADA 2 OVERFLOWS PASA 100MS
941     SETB TR0
942     ACALL MOVIMIENTO_RH   ;REALIZAR EL CONTROL HORIZONTAL
943     ACALL MOVIMIENTO_RV   ;REALIZAR EL CONTROL VERTICAL
944     RET
945
946 FIN_FROTADO:
947     ANL ADCON, #0XEF
948     ACALL APAGAR_ADC
949     CLR FLAG_100MS
950     CLR FLAG_ADC2
951     SETB FLAG_MOVER2
952     ACALL APAGAR_TIMER0
953     RET
954
955 DETENER_RODILLOS:
956     CLR BMOV_BACK
957     CLR RV_OUTTO
958     CLR RV_INTTO
959     CLR RH_UPTO
960     CLR RH_DOWNTO
961     ACALL FIN_FROTADO
962     ACALL SUBIR_RH_FIN
963     RET
964
965 SUBIR_RH_FIN:
966     SETB RH_UPTO
967     JNB FC_RH_TOP, SUBIR_RH_FIN ;SUBIMOS EL RODILLO HASTA SU POS INICIAL MARCADA POR
968     CLR RH_UPTO
969     MOV ESTADO, #8
970     RET
971
972
973 ;=====ESTADO_8=====
974 RODILLOS_PRINCIPIO2:
975     CLR FC_RV_CENTRO
976     MOV ESTADO, #9
977     RET
978
979 MOVER_4S3:
980     JNB BMOV_BACK, PRIMERA_4S3
981     RET
982
983
984
985 PRIMERA_4S3:
986     ACALL ENCENDER_TIMER0
987     MOV TH0, #01H      ;HEMOS HECHO EL CALCULO DE CICLOS DE RELOJ PARA 1S USANDO LA
988     FRECUENCIA DEL ENUNCIADO Y NOS DA QUE TENEMOS QUE INICIALIZAR EL TIMER0 EN 316
989     MOV TL0, #3CH
990     MOV T_CONT1, #15    ;CADA 15 OVERFLOWS PASA 1S
991     MOV T_CONT2, #4
992     SETB BMOV_BACK
993     RET
994
995 PASAN_4S3:
996     ACALL APAGAR_TIMER0
997     CLR BMOV_BACK
998     SETB FLAG_4S3
999     RET
1000
1001 ;=====ESTADO_9=====
1002
1003 MOVER_COCHE3:

```

```

1004     SETB FLAG_MOVER3
1005     JNB FLAG_AGUA, PRIMERA_ACTIVACION2
1006     CLR BMOV_FRONT
1007     CLR EV_AGUA
1008     SETB FLAG_4S
1009     RET
1010
1011 PRIMERA_ACTIVACION2:
1012     SETB BMOV_FRONT
1013     RET
1014
1015 ACTIVAR_EV2:
1016     CLR FLAG_MOVER3
1017     SETB EV_AGUA
1018     SETB FLAG_AGUA
1019     RET
1020
1021 MOVER_4S2:
1022     JNB BMOV_FRONT, PRIMERA_4S2
1023     RET
1024
1025
1026
1027 PRIMERA_4S2:
1028     ACALL ENCENDER_TIMER0
1029     MOV TH0, #01H           ;HEMOS HECHO EL CALCULO DE CICLOS DE RELOJ PARA 1S USANDO LA
1030                             FRECUENCIA DEL ENUNCIADO Y NOS DA QUE TENEMOS QUE INICIALIZAR EL TIMER0 EN 316
1031     MOV TL0, #3CH
1032     MOV T_CONT1, #15        ;CADA 15 OVERFLOWS PASA 1S
1033     MOV T_CONT2, #4
1034     SETB BMOV_FRONT
1035     RET
1036
1037 PASAN_4S2:
1038     ACALL APAGAR_TIMER0
1039     CLR BMOV_FRONT
1040     CLR FLAG_4S
1041     MOV ESTADO, #10
1042     RET
1043
1044 ;=====ESTADO_10=====
1045 MOVER_COCHE4:
1046     SETB FLAG_MOVER4
1047     JNB FLAG_SECAO, PRIMERA_ACTIVACION3
1048     SETB BMOV_BACK
1049     ACALL APAGAR_PWM
1050     RET
1051
1052 PRIMERA_ACTIVACION3:
1053     SETB BMOV_BACK
1054     RET
1055
1056 ACTIVAR_SECAO:
1057     CLR FLAG_MOVER4
1058     JNB FICHA_TIPO, PWM1_NORMAL
1059     JB FICHA_TIPO, PWM1_INTENSO
1060     RET
1061
1062 FIN_DE_CARRERA:
1063     CLR BMOV_BACK
1064     MOV ESTADO, #11
1065     RET
1066 ;=====ESTADO_11=====
1067 ACCION2:
1068     ACALL ENCENDER_TIMER0
1069     CLR BPOS_START
1070     CLR SEM_ROJO

```

```

1071     SETB SEM_VERDE
1072     MOV TH0, #01H           ;HEMOS HECHO EL CALCULO DE CICLOS DE RELOJ PARA 1S USANDO LA
                                FRECUENCIA DEL ENUNCIADO Y NOS DA QUE TENEMOS QUE INICIALIZAR EL TIMER0 EN 316
1073     MOV TL0, #3CH
1074     MOV T_CONTR1, #15      ;CADA 15 OVERFLOWS PASA 1S
1075     RET
1076 ;=====PWM=====;
1077
1078 ENCENDER_PWMP:              ;ENCIENDE EL PWM
1079     MOV PWMP, #4d
1080     RET
1081 APAGAR_PWM:                 ;APAGA EL PWM
1082     MOV PWMP, #0
1083     MOV PWM0, #255
1084     MOV PWM1, #255
1085     RET
1086 PWM0_NORMAL:                ;VALOR DE POTENCIA RODILLOS NORMAL 50%
1087     MOV PWM0, #128
1088     ACALL MOVIMIENTO_DENTRO
1089     RET
1090 PWM0_INTENSO:               ;VALOR DE POTENCIA RODILLOS INTENSO 100%
1091     MOV PWM0, #255
1092     ACALL MOVIMIENTO_DENTRO
1093     RET
1094 PWM1_NORMAL:                ;VALOR DE POTENCIA DE LOS VENTILADORES NORMAL 50%
1095     MOV PWM1, #128
1096     ACALL ENCENDER_PWMP
1097     SETB FLAG_SECADO
1098     RET
1099 PWM1_INTENSO:               ;VALOR DE POTENCIA DE LOS VENTILADORES INTENSO 85%
1100     MOV PWM1, #38
1101     ACALL ENCENDER_PWMP
1102     SETB FLAG_SECADO
1103     RET
1104
1105 ;=====ADC=====;
1106 ENCENDER_ADC:
1107     JNB SEGUNDA_CONVERSION, ADC_H
1108     JB SEGUNDA_CONVERSION, ADC_V
1109     RET
1110
1111 ADC_H:
1112     JNB FLAG_ADC2, PRIMERA_ADC1
1113     ANL ADCON, #0xD8        ;SELECCION PUERTO 5.0 RODILLO HORIZONTAL
1114     ORL ADCON, #0x08        ;INICIAR CONVERSION
1115     RET
1116
1117 PRIMERA_ADC1:
1118     ANL ADCON, #0xD8        ;SELECCION PUERTO 5.0 RODILLO HORIZONTAL
1119     ORL ADCON, #0x08        ;INICIAR CONVERSION
1120     SETB FLAG_ADC2
1121     RET
1122 ADC_V:
1123     JNB FLAG_ADC2, PRIMERA_ADC2
1124     ANL ADCON, #0xD9        ;SELECCION PUERTO 5.1 RODILLO HORIZONTAL
1125     ORL ADCON, #0x08        ;INICIAR CONVERSION
1126     RET
1127
1128 PRIMERA_ADC2:
1129     ORL ADCON, #0x11        ;SELECCION PUERTO 5.1 RODILLO HORIZONTAL
1130     ORL ADCON, #0x08        ;INICIAR CONVERSION
1131     SETB FLAG_ADC2
1132     RET
1133
1134 APAGAR_ADC:                 ;APAGA EL ADC Y DESCATIVA INTERRUPCIONES
1135     ANL IEN0, #0000000b
1136     RET
1137

```

```

1138
1139 ;=====TIMER=====;
1140 ENCENDER_TIMER0:          ; Rutina para encender el temporizador
1141     MOV TMOD, #00000001B
1142     ORL IEN0, #10000010b   ; Habilita interrupciones globales y del Timer 0
1143     CLR T_CONT1            ; Reinicia contadores si los estas usando
1144     CLR T_CONT2
1145     SETB TR0               ; Inicia el Timer 0
1146     RET
1147
1148 APAGAR_TIMER0:            ;APAGA EL TIMER
1149     ANL IEN0, #00000000b
1150     CLR TR0
1151     RET
1152
1153 ENCENDER_TIMER1:          ; Rutina para encender el temporizador
1154     MOV TMOD, #00010000B   ;PONER TIMER 1 EN MODO 1
1155     ORL IEN0, #10001000b   ; Habilita interrupciones globales y del Timer 1
1156     CLR T_CONT1            ; Reinicia contadores si los estas usando
1157     CLR T_CONT2
1158     SETB TR1               ; Inicia el Timer 1
1159     RET
1160
1161 APAGAR_TIMER1:            ;APAGA EL TIMER
1162     ANL IEN0, #00000000b
1163     CLR TR1
1164     CLR FLAG_30
1165     RET
1166
1167 MANEJAR_INTERRUPCIONES:
1168     MOV R1, ESTADO          ; Carga el valor de ESTADO en R1
1169     CJNE R1, #1, CHECK1     ; Compara con el valor 1, si no es igual salta a CHECK1
1170     AJMP PLACA              ; Si es igual a 1, salta a la rutina PLACA
1171
1172 CHECK1:
1173     CJNE R1, #4, CHECK2     ; Compara con el valor 4, si no es igual salta a CHECK2
1174     AJMP DESPLAZAMIENTO     ; Si es igual a 4, salta a la rutina DESPLAZAMIENTO
1175
1176 CHECK2:
1177     CJNE R1, #6, CHECK3     ; Compara con el valor 4, si no es igual salta a CHECK2
1178     AJMP CONVERSION         ; Si es igual a 4, salta a la rutina DESPLAZAMIENTO
1179
1180 CHECK3:
1181     CJNE R1, #9, CHECK4     ; Compara con el valor 9, si no es igual salta a CHECK3
1182     AJMP DESPLAZAMIENTO2    ; Si es igual a 9, salta a la rutina DESPLAZAMIENTO2
1183
1184 CHECK4:
1185     CJNE R1, #7, CHECK5
1186     AJMP FROTADO_INT
1187     RETI
1188
1189 CHECK5:
1190     CJNE R1, #8, CHECK6
1191     AJMP DESPLAZAMIENTO3
1192     RETI
1193
1194 CHECK6:
1195     AJMP SEM_INTERMITENTE   ; Si no coincide con los valores anteriores, salta a
1196     SEM_INTERMITENTE
1197     RETI
1198
1199 SEM_INTERMITENTE:         ;FUNCION DEL TIMER SEMAFORO INTERMITENTE
1200     PUSH PSW
1201     PUSH ACC
1202
1203     CLR TR0
1204     MOV TH0, #01H
1205     MOV TL0, #3CH
1206     DJNZ T_CONT1, SIGUE0
1207     CPL SEM_VERDE          ;HA PASADO 1000ms!
1208     MOV T_CONT1, #15

```



```

1205     JNB S_PLAT, FIN_TIMER0
1206     SETB TR0           ;VOLVEMOS A PONER EN MARCHA EL TIMER0
1207     POP ACC            ;Restaura el acumulador (A)
1208     POP PSW
1209     RETI
1210
1211 FIN_TIMER0:
1212     ACALL APAGAR_TIMER0
1213     POP ACC            ;Restaura el acumulador (A)
1214     POP PSW
1215     RETI
1216
1217 PLACA:
1218     PUSH PSW
1219     PUSH ACC
1220
1221     CLR TR0
1222     MOV TH0, #01H
1223     MOV TL0, #3CH
1224     DJNZ T_CONT1, SIGUE0 ;HA PASADO 1s! SI T_CONT1 ES 0
1225     MOV T_CONT1, #15     ;CARGAMOS OTRA VEZ 15 OVERFLOWS
1226     JNB S_PLAT, FIN_PLACA
1227     DJNZ T_CONT2, SIGUE0 ;HAN PASADO 5 SEGS
1228     JNB S_PLAT, FIN_PLACA
1229
1230     POP ACC
1231     POP PSW
1232     RETI
1233
1234 FIN_PLACA:
1235     ACALL APAGAR_TIMER0
1236     POP ACC
1237     POP PSW
1238     LJMP ME1_PLACAOOUT
1239     RETI
1240
1241 SIGUE0:
1242     SETB TR0
1243     POP ACC
1244     POP PSW
1245     RETI
1246 DESPLAZAMIENTO:           ;SUBROUTINA PARA LA ATENCION A LA INTERRUPCION DE DESPLAZAR EL COCHE
1247     40 CM
1248     PUSH PSW
1249     PUSH ACC
1250
1251     CLR TR0
1252     MOV TH0, #01H
1253     MOV TL0, #3CH
1254     DJNZ T_CONT1, SIGUE0 ;HA PASADO 1s! SI T_CONT1 ES 0
1255     MOV T_CONT1, #15     ;CARGAMOS OTRA VEZ 15 OVERFLOWS
1256     DJNZ T_CONT2, SIGUE0 ;HAN PASADO 4 SEGS
1257     ACALL PASAN_4S
1258
1259     POP ACC
1260     POP PSW
1261     RETI
1262 DESPLAZAMIENTO2:         ;SUBROUTINA PARA LA ATENCION A LA INTERRUPCION DE DESPLAZAR EL COCHE
1263     40 CM
1264     PUSH PSW
1265     PUSH ACC
1266
1267     CLR TR0
1268     MOV TH0, #01H
1269     MOV TL0, #3CH
1270     DJNZ T_CONT1, SIGUE0 ;HA PASADO 1s! SI T_CONT1 ES 0
1271     MOV T_CONT1, #15     ;CARGAMOS OTRA VEZ 15 OVERFLOWS
1272     DJNZ T_CONT2, SIGUE0 ;HAN PASADO 4 SEGS

```

```

1271     ACALL PASAN_4S2
1272
1273     POP ACC
1274     POP PSW
1275     RETI
1276 DESPLAZAMIENTO3:      ;SUBROUTINA PARA LA ATENCION A LA INTERRUPCION DE DESPLAZAR EL COCHE
1277     40 CM
1278     PUSH PSW
1279     PUSH ACC
1280
1281     CLR TR0
1282     MOV TH0, #01H
1283     MOV TL0, #3CH
1284     DJNZ T_CONT1, SIGUE0      ;HA PASADO 1s! SI T_CONT1 ES 0
1285     MOV T_CONT1, #15      ;CARGAMOS OTRA VEZ 15 OVERFLOWS
1286     DJNZ T_CONT2, SIGUE0      ;HAN PASADO 4 SEGS
1287     ACALL PASAN_4S3
1288
1289     POP ACC
1290     POP PSW
1291     RETI
1292
1293 CONVERSION:
1294     PUSH PSW
1295     PUSH ACC
1296
1297     CLR TR0
1298     MOV TH0, #01H
1299     MOV TL0, #3CH
1300     DJNZ T_CONT1, SIGUE0      ;HA PASADO 100MS! SI T_CONT1 ES 0
1301     ACALL PASAN_100MS
1302
1303     POP ACC
1304     POP PSW
1305     RETI
1306
1307 FROTADO_INT:
1308     PUSH PSW
1309     PUSH ACC
1310
1311     CLR TR0
1312     MOV TH0, #01H
1313     MOV TL0, #3CH
1314     DJNZ T_CONT1, SIGUE0      ;HA PASADO 100MS! SI T_CONT1 ES 0
1315     ACALL PASAN_100MS_F
1316
1317     POP ACC
1318     POP PSW
1319     RETI
1320
1321
1322 ALARMA:
1323     PUSH PSW
1324     PUSH ACC      ;Guarda el acumulador (A)
1325
1326     CLR TR1
1327     MOV TH1, #01H
1328     MOV TL1, #3CH
1329     DJNZ T_CONT1, SIGUE1      ;HA PASADO 1s! SI T_CONT1 ES 0
1330     MOV T_CONT1, #15      ;CARGAMOS OTRA VEZ 15 OVERFLOWS
1331     DJNZ T_CONT2, SIGUE1      ;HAN PASADO 30 SEGS
1332     AJMP ACTIVAR_ALARMA
1333
1334 SIGUE1:
1335     SETB TR1
1336     POP ACC      ;Restaura el acumulador (A)
1337     POP PSW

```

```

1338     MOV A, FLAG_A
1339     RETI
1340
1341 FIN_ALARMA:
1342     ACALL APAGAR_TIMER1
1343     POP ACC
1344     POP PSW
1345     MOV A, FLAG_A
1346     RETI
1347
1348 ACTIVAR_ALARMA:
1349     SETB AL_START
1350     ACALL APAGAR_TIMER1
1351     POP ACC
1352     POP PSW
1353     MOV A, FLAG_A
1354     RETI
1355
1356 NO_ALARMA:
1357     PUSH PSW
1358     PUSH ACC
1359
1360     CLR TR1
1361     MOV TH1, #01H
1362     MOV TL1, #3CH
1363     DJNZ T_CONT1, SIGUE1      ;HA PASADO 1s! SI T_CONT1 ES 0
1364     MOV T_CONT1, #15         ;CARGAMOS OTRA VEZ 15 OVERFLOWS
1365     DJNZ T_CONT2, SIGUE1     ;HAN PASADO 30 SEGS
1366     ACALL PASAN30S
1367
1368     ACALL APAGAR_TIMER1
1369     POP ACC
1370     POP PSW
1371     MOV A, FLAG_A
1372     RETI
1373
1374 ;=====INTERRUPTIONES=====;
1375 ORG 0x0B
1376 INTERRUPCION_TIMER0:
1377     AJMP MANEJAR_INTERRUPTIONES
1378     RETI
1379
1380
1381 ORG 0x1B
1382 INTERRUPCION_TIMER1:
1383     MOV FLAG_A, A
1384     MOV A, FLAG_ALARMA
1385     JZ FLAG2
1386     AJMP ALARMA
1387     RETI
1388
1389 FLAG2:
1390     AJMP NO_ALARMA
1391     RETI
1392 ;=====END=====;
1393 END

```