

**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)

…

18 de diciembre de 2024

Bidaguren Ibáñez Gorka

Rocha Astutiani Eduardo

**Índice**

[**1.** **Introducción** 4](#_Toc185893963)

[**1.2.** **Bucle Principal** 5](#_Toc185893964)

[**1.2.1.** **Generador de Eventos** 6](#_Toc185893965)

[**1.2.2.** **Máquina de Estados** 6](#_Toc185893966)

[**2.** **Diagramas de estado y de flujo** 7](#_Toc185893967)

[**3.** **Cálculos y comentarios** 7](#_Toc185893968)

[**4.** **Código** 7](#_Toc185893969)

**Índice de ilustraciones**

[Ilustración 1: Modelo de Lavadero de Coches 4](#_Toc185893880)

[Ilustración 2: Bucle Principal 5](#_Toc185893881)

[Ilustración 3: Generador de Eventos 6](#_Toc185893882)

[Ilustración 4: Maquina de Estados 7](#_Toc185893883)

# **Introducción**

En este informe se recoge el trabajo desarrollado relativo a la Práctica Final de la asignatura de Arquitectura de Computadores.

El objetivo principal ha sido el desarrollo de software para microcontroladores. Por ello, se ha utilizado el microcontrolador 80C552 del fabricante Philips Semiconductors. Se ha llevado a cabo en el entorno de simulación Keil µVision2.

Se han trabajo los conceptos de Interrupciones, ADC (Analogic-Digital Converter) para los sensores, PWM (Pulse Width Modulation) para la intensidad de los rodillos y Timers para controlar los tiempos.

La temática de la práctica ha sido la realización de un lavadero de coches. Su descripción se encuentra en el enunciado facilitado por el profesorado.

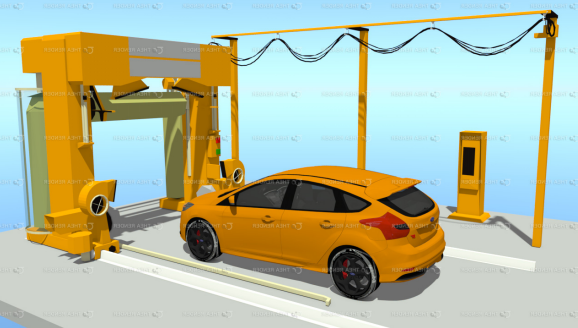


Ilustración : Modelo de Lavadero de Coches

Con el fin de lograr el completo desarrollo del programa, en primer lugar, se han diseñado el diagrama de estados, eventos y acciones y los diagramas de flujo correspondientes. Además, se implementaron soluciones específicas como el uso de interrupciones para gestionar eventos en tiempo real, el ADC para leer datos de sensores y el PWM para controlar la intensidad de los actuadores. Asimismo, se establecieron estrategias de manejo de errores y de transición entre estados para asegurar un flujo continuo y eficiente del sistema.

## **Bucle Principal**

El bucle principal es el núcleo del sistema desarrollado para el lavadero de coches, se realiza un bucle infinito mientras el sistema esté en funcionamiento realizando las dos tareas principales del software:

1. **Llamar al generador de eventos**: Detecta y registra las transiciones de los sensores y entradas del sistema.
2. **Ejecutar la máquina de estados**: Dependiendo del evento detectado y el estado actual, se toman decisiones para controlar el sistema y avanzar en el flujo del programa.

Texto

Descripción generada automáticamente con confianza baja

Ilustración : Bucle Principal

Esta implementación garantiza que el sistema reaccione de manera continua y rápida procesando los eventos y actualizando el estado rápidamente cuando sea necesario.

### **Generador de Eventos**

En el sistema, se diseñó un **generador de eventos único** para gestionar las interacciones entre hardware y software. El generador de eventos único ocurre en el bucle principal y detecta cambios en los bits de los puertos de entrada/salida del microcontrolador (P0, P1, P2) y activa los eventos correspondientes en función de las transiciones detectadas para luego aplicarlos en el estado en el que se encuentra. De esta manera, todos los estados son sensibles a todos los eventos, sin embargo, en los estados en los cuales el evento no tiene ninguna acción asociada, se considera como si se tratara de un evento 0.

El generador de eventos monitorea los puertos de entrada del microcontrolador (P0, P1 y P2) y utiliza una operación XOR para detectar flancos ascendentes y descendentes en las señales. Esto permite identificar transiciones específicas, como la activación de un sensor.

Ejemplo de código para el puerto P0:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración : Generador de Eventos

### **Máquina de Estados**

La máquina de estados es responsable de gestionar el flujo del sistema según el estado actual y los eventos de cada estado. Define, en cada bucle, el comportamiento actual del sistema y maneja las diferentes fases del lavado de los coches.

La máquina de estados evalúa el estado actual y utiliza una tabla de saltos para ejecutar las acciones correspondientes:

Texto

Descripción generada automáticamente con confianza baja

Ilustración : Maquina de Estados

# **Diagramas de estado y de flujo**

# **Cálculos y comentarios**

# **Código**

;Variables

ESTADO EQU R0

EVENTO EQU R1

ANT0 EQU 0x30

ACT0 EQU 0x31

ANT1 EQU 0x32

ACT1 EQU 0x33

ANT2 EQU 0x34

ACT2 EQU 0x35

EV1 EQU 0x20.0

EV2 EQU 0x20.1

EV3 EQU 0x20.2

EV4 EQU 0x20.3

EV5 EQU 0x20.4

EV6 EQU 0x20.5

EV7 EQU 0x20.6

EV8 EQU 0x20.7

EV9 EQU 0x21.0

EV10 EQU 0x21.1

EV11 EQU 0x21.2

EV12 EQU 0x21.3

EV13 EQU 0x21.4

EV14 EQU 0x21.5

EV15 EQU 0x21.6

EV16 EQU 0x21.7

EV17 EQU 0x22.0

EV18 EQU 0x22.1

EV19 EQU 0x22.2

EV20 EQU 0x22.3

EV21 EQU 0x22.4

EV22 EQU 0x22.5

EV23 EQU 0x22.6

EV24 EQU 0x22.7

EV25 EQU 0x23.0

EV26 EQU 0x23.1

RH\_40cm EQU 0x25.2

RV\_40cm EQU 0x25.3

;PMW

PWM0 EQU 0xFC

PWM1 EQU 0xFD

PWMP EQU 0xFE

;Puerto P0

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 P1

BMOV\_FRONT EQU P1.0

BMOV\_BACK EQU P1.1

FC\_RH\_TOP EQU P1.2

RH\_UPTO EQU P1.3

RH\_DOWNTO EQU P1.4

FC\_RV\_BORDE EQU P1.5

RV\_OUTTO EQU P1.6

RV\_INTO EQU P1.7

;Puerto P2

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

;Timer

tick\_100ms EQU 0x24.0

tick\_1s EQU 0x24.1

tick\_4s EQU 0x24.2

tick\_5s EQU 0x24.3

tick\_30s EQU 0x24.4

cont\_50ms EQU R2

cont\_100ms EQU R3

cont\_5s EQU R4

cont\_100ms\_1s EQU R5

;ADC

EAD EQU 0xA8.6

ADCON EQU 0xC5

ADCH EQU 0xC6

tick\_ADCR0 EQU 0x25.0

tick\_ADCR1 EQU 0x25.1

VALOR\_ADC\_0 EQU R6

VALOR\_ADC\_1 EQU R7

;#########################################################################

; PROGRAMA PRINCIPAL

;#########################################################################

ORG 0x0000

ACALL INICIO

INICIO:

ACALL INICIALIZAR

BUCLE:

ACALL GEN\_EVE

ACALL MAQ\_ESTADOS

AJMP BUCLE

;#########################################################################

; Interrupcion\_Timer

;#########################################################################

ORG 0x000B

MOV TH0, #0x3C ;Valor de 13536 --> para contar 50ms

MOV TL0, #0xB0

CLR TF0 ; Limpia el flag de interrupci n

DEC cont\_50ms

CJNE cont\_50ms, #0x00, FIN\_Int\_Timer\_0 ; Si cont\_50ms se ha hecho 2 veces, NO salta

MOV cont\_50ms, #0x02 ; Reinicia el contador

SETB tick\_100ms ; Activa el flag de 100ms

DEC cont\_100ms

DEC cont\_100ms\_1s

CJNE cont\_100ms, #0x0A, NO\_4s ; Si cont\_50ms se ha hecho 40 veces, NO salta

SETB tick\_4s ; Activa el flag de 4s

NO\_4s:

CJNE cont\_100ms\_1s, #0x00, NO\_1s ; Si cont\_100ms\_1s se ha hecho 10 veces, NO salta

SETB tick\_1s ; Activa el flag de 1s

MOV cont\_100ms\_1s, #0x0A

NO\_1s:

CJNE cont\_100ms, #0x00, FIN\_Int\_Timer\_0 ; Si cont\_50ms se ha hecho 50 veces, NO salta

MOV cont\_100ms, #0x32 ; Reinicia el contador

SETB tick\_5s ; Activa el flag de 5s

DEC cont\_5s

CJNE cont\_5s, #0x00, FIN\_Int\_Timer\_0 ; Si cont\_5s se ha hecho 6 veces, NO salta

MOV cont\_5s, #0x06 ; Reinicia el contador

SETB tick\_30s ; Activa el flag de 30s

FIN\_Int\_Timer\_0:

RETI

;#########################################################################

; Interrupcion\_ADC

;#########################################################################

ORG 0x0053

PUSH 0xE0 ; Guardar el acumulador en el stack

MOV A, ADCON ; Lee el valor del ADCON para determinar el canal

ANL A, #0x0B ; Aisla los bits AADR2:AADR0

CJNE A, #0x01, CANAL0\_ADC ; Comprueba si el canal le do es el 1

SETB tick\_ADCR1

SJMP CANAL1\_ADC

CANAL0\_ADC:

SETB tick\_ADCR0

MOV VALOR\_ADC\_0, ADCH

MOV ADCON, #0x09 ; activa adcs y empieza la conversion en P5.1

SJMP FIN\_ADC

CANAL1\_ADC:

MOV VALOR\_ADC\_1, ADCH

SETB tick\_ADCR1

FIN\_ADC:

POP 0xE0 ; Recupera el valor del acumulador

RETI

;#########################################################################

; TIMERS

;#########################################################################

ACTIVAR\_TIMER\_0:

MOV TMOD, #00000001 ;Poner el timer\_0 en modo 1 (16 bits)

MOV cont\_50ms, #0x02 ;llega a 0 al hacerse 2 veces --> 100ms

MOV cont\_100ms, #0x32 ;llega a 0 al hacerse 50 veces --> 5s

MOV cont\_100ms\_1s, #0x0A;llega a 0 al hacerse 10 veces --> 1s

MOV cont\_5s, #0x06 ;llega a 0 al hacerse 6 veces --> 30s

MOV TH0, #0x3C ;Valor de 13536 --> para contar 50ms

MOV TL0, #0xB0

SETB EA ;activar las interrupciones

SETB ET0 ;Habilitar interrupciones del timer\_0

SETB TCON.4 ;Enciende el timer

RET

DESACTIVAR\_TIMER\_0:

CLR EA ;desactivar las interrupciones

CLR ET0 ;Inhabilitar interrupciones del timer\_0

CLR TCON.4 ;Apaga el timer

RET

;#########################################################################

; INICIALIZACIONES

;#########################################################################

INICIALIZAR:

MOV PWM0, #0xFF

MOV PWM1, #0xFF

CLR EA ;desactivar las interrupciones

MOV ESTADO, #0x0A

MOV EVENTO, #0x00

MOV ANT0, #0x00

MOV ACT0, #0x00

MOV ANT1, #0x00

MOV ACT1, #0x00

MOV ANT2, #0x00

MOV ACT2, #0x00

; ENTRADAS:

SETB S\_PLAT

SETB FICHA\_TIPO

SETB S\_FICHA

SETB P\_START

SETB BPOS\_START

SETB FC\_RH\_TOP

SETB FC\_RV\_BORDE

SETB S\_CAR

SETB FC\_RV\_CENTRO

; SALIDAS

CLR SEM\_ROJO

CLR AL\_START

CLR BMOV\_FRONT

CLR BMOV\_BACK

CLR RH\_UPTO

CLR RH\_DOWNTO

CLR RV\_OUTTO

CLR RV\_INTO

CLR EV\_AGUA

CLR EV\_JABON

CLR LED\_LNOR

CLR LED\_LINT

CLR SEM\_VERDE

; NO UTILIZADOS

CLR P2.2

CLR P2.7

;FLAGS

CLR EV1

CLR EV2

CLR EV3

CLR EV4

CLR EV5

CLR EV6

CLR EV7

CLR EV8

CLR EV9

CLR EV10

CLR EV11

CLR EV12

CLR EV13

CLR EV14

CLR EV15

CLR EV16

CLR EV17

CLR EV18

CLR EV19

CLR EV20

CLR EV21

CLR EV22

CLR EV23

CLR EV24

CLR EV25

CLR EV26

CLR tick\_100ms

CLR tick\_4s

CLR tick\_5s

CLR tick\_30s

CLR RH\_40cm

CLR RV\_40cm

;PWMP

MOV PWMP, #0x17

SETB SEM\_VERDE

RET

;#########################################################################

; M QUINA DE ESTADOS

;#########################################################################

MAQ\_ESTADOS:

MOV A, ESTADO

RL A

MOV DPTR, #LISTA\_EST

JMP @A+DPTR

LISTA\_EST:

AJMP ESPERA\_COCHE; Estado0

AJMP COCHE\_PUESTO; Estado1

AJMP ALARMA; Estado2

AJMP PONER\_PUENTE; Estado3

AJMP ENJABONADO; Estado4

AJMP LIMPIAR\_ATRAS; Estado5

AJMP PONER\_RODILLOS; Estado6

AJMP RODILLOS; Estado7

AJMP LIMPIAR\_DELANTE;Estado8

AJMP ACLARADO; Estado9

AJMP SECADO; Estado10

AJMP RETIRAR\_COCHE; Estado11

;#########################################################################

; GENERADOR DE EVENTOS

;#########################################################################

GEN\_EVE:

; VERIFICACION DE LOS BITS DEL PUERTO 0

MOV A, ACT0

MOV ANT0, A ; Dar valores a ACT y ANT

MOV ACT0, P0

MOV A, ACT0

XRL A, ANT0 ; XOR con el valor anterior para detectar cambios

JZ PUERTO1\_GV

MOV B, ANT0

; Verificaci n para bit 0.2 (S\_PLAT)

JNB ACC.2, S\_FICHA\_GV

JNB B.2, ASCENDENTE\_SPLAT ; Si ANT.2 era 0 y cambi a 1, es un flanco ascendente

JB B.2, DESCENDENTE\_SPLAT ; Si ANT.2 era 1 y cambi a 0, es un flanco descendente

ASCENDENTE\_SPLAT:

SETB EV2 ; EVENTO 2

SJMP S\_FICHA\_GV

DESCENDENTE\_SPLAT:

SETB EV3 ; EVENTO 3

; Verificaci n para bit 0.4 (S\_FICHA)

S\_FICHA\_GV:

JNB ACC.4, P\_START\_GV

JNB B.4, ASCENDENTE\_SFICHA ; Si ANT.2 era 0 y cambi a 1, es un flanco ascendente

SJMP P\_START\_GV

ASCENDENTE\_SFICHA:

SETB EV5 ; EVENTO 5

; Verificaci n para bit 0.5 (P\_START) y S\_FICHA

P\_START\_GV:

JNB ACC.5, BPOS\_START\_GV

JNB B.5, ASCENDENTE\_PSTART ; Si ANT.2 era 0 y cambi a 1, es un flanco ascendente

SJMP BPOS\_START\_GV

ASCENDENTE\_PSTART:

JNB S\_FICHA, BPOS\_START\_GV ; Para que S\_ficha tambien tenga que estar activado

SETB EV6 ; EVENTO 6

; Verificaci n para bit 0.7 BPOS\_START

BPOS\_START\_GV:

JNB ACC.7, PUERTO1\_GV

JNB B.7, ASCENDENTE\_BPOS\_START ; Si ANT.2 era 0 y cambi a 1, es un flanco ascendente

SJMP PUERTO1\_GV

ASCENDENTE\_BPOS\_START:

SETB EV20 ; EVENTO 20

; VERIFICACION DE LOS BITS DEL PUERTO 1

PUERTO1\_GV:

MOV A, ACT1

MOV ANT1, A ; Dar valores a ACT y ANT

MOV ACT1, P1

MOV A, ACT1

XRL A, ANT1 ; XOR con el valor anterior para detectar cambios

JZ PUERTO2\_GV

MOV B, ANT1

; Verificaci n para bit 1.2 (FC\_RH\_TOP)

JNB ACC.2, FC\_RV\_BORDE\_GV

JNB B.2, ASCENDENTE\_FC\_RH\_TOP\_GV ; Si ANT.2 era 0 y cambi a 1, es un flanco ascendente

SJMP FC\_RV\_BORDE\_GV

ASCENDENTE\_FC\_RH\_TOP\_GV:

SETB EV19 ; EVENTO 19

; Verificaci n para bit 1.5 (FC\_RV\_BORDE)

FC\_RV\_BORDE\_GV:

JNB ACC.5, PUERTO2\_GV

JNB B.5, ASCENDENTE\_FC\_RV\_BORDE\_GV ; Si ANT.2 era 0 y cambi a 1, es un flanco ascendente

SJMP PUERTO2\_GV

ASCENDENTE\_FC\_RV\_BORDE\_GV:

SETB EV16 ; EVENTO 16

; VERIFICACION DE LOS BITS DEL PUERTO 2

PUERTO2\_GV:

MOV A, ACT2

MOV ANT2, A ; Dar valores a ACT y ANT

MOV ACT2, P2

MOV A, ACT2

XRL A, ANT2 ; XOR con el valor anterior para detectar cambios

JZ FIN\_PUERTOS

MOV B, ANT2

; Verificaci n para bit 2.3 (S\_CAR)

JNB ACC.3, FC\_RV\_CENTRO\_GV

JNB B.3, ASCENDENTE\_SCAR ; Si ANT.2 era 0 y cambi a 1, es un flanco ascendente

JB B.3, DESCENDENTE\_SCAR ; Si ANT.2 era 1 y cambi a 0, es un flanco descendente

ASCENDENTE\_SCAR:

SETB EV8 ; EVENTO 8

SJMP FC\_RV\_CENTRO\_GV

DESCENDENTE\_SCAR:

SETB EV9 ; EVENTO 9

; Verificaci n para bit 2.4 (FC\_RV\_CENTRO)

FC\_RV\_CENTRO\_GV:

JNB ACC.4, FIN\_PUERTOS

JNB B.4, ASCENDENTE\_FC\_RV\_CENTRO\_GV ; Si ANT.2 era 0 y cambi a 1, es un flanco ascendente

SJMP FIN\_PUERTOS

ASCENDENTE\_FC\_RV\_CENTRO\_GV:

SETB EV17 ; EVENTO 17

FIN\_PUERTOS:

; VERIFICACION DE LOS FLAGS DEL TIMER

; Verificaci n para flag tick\_4s

JNB tick\_4s, tick\_5s\_GV

CLR tick\_4s

SETB EV10 ; EVENTO 10

; Verificaci n para flag tick\_5s

tick\_5s\_GV:

JNB tick\_5s, tick\_30s\_GV

CLR tick\_5s

SETB EV1 ; EVENTO 1

; Verificaci n para flag tick\_30s

tick\_30s\_GV:

JNB tick\_30s, tick\_30s\_S\_FICHA

SETB EV4 ; EVENTO 4

; Verificaci n para flag tick\_30s y S\_FICHA

tick\_30s\_S\_FICHA:

JNB tick\_30s, tick\_100ms\_GV

JNB S\_FICHA, tick\_100ms\_GV

SETB EV7 ; EVENTO 7

; Verificaci n para flag tick\_100ms

tick\_100ms\_GV:

CLR tick\_30s

JNB tick\_100ms, tick\_1s\_GV

SETB EV26 ; EVENTO 26

CLR tick\_100ms

; Verificaci n para flag tick\_1s

tick\_1s\_GV:

JNB tick\_1s, FIN\_FLAGS\_TIMER

SETB EV21 ; EVENTO 21

CLR tick\_1s

FIN\_FLAGS\_TIMER:

; VERIFICACION DE LOS FLAGS DEL ADC

; Verificaci n para flag ADCR0: 37.5 <= SPOS\_RH <= 40, SPOS\_RH < 37.5 y SPOS\_RH > 42.5

CLR C

JNB tick\_ADCR0, ADCR1\_GV

CJNE VALOR\_ADC\_0, #0x60, NO\_IGUAL\_ADC\_00

SJMP EN\_RANGO\_0

NO\_IGUAL\_ADC\_00:

JC MENOR\_0

CJNE VALOR\_ADC\_0, #0x6C, NO\_IGUAL\_ADC\_01

SJMP EN\_RANGO\_0

NO\_IGUAL\_ADC\_01:

JC EN\_RANGO\_0

; SPOS\_RH > 42.5

SETB EV15 ; EVENTO 15

SJMP FIN\_ADRO

EN\_RANGO\_0: ; 37.5 <= SPOS\_RH <= 40

SETB EV24 ; EVENTO 24

SJMP FIN\_ADRO

MENOR\_0: ; SPOS\_RH < 37.5

SETB EV14 ; EVENTO 14

SJMP FIN\_ADRO

FIN\_ADRO:

CLR tick\_ADCR0

; Verificaci n para flag ADCR1: 37.5 <= SPOS\_RV <= 40, SPOS\_RV < 37.5 y SPOS\_RV > 42.5

ADCR1\_GV:

CLR C

JNB tick\_ADCR1, RODIILLOS\_POSICION

CJNE VALOR\_ADC\_1, #0x60, NO\_IGUAL\_ADC\_10

SJMP EN\_RANGO\_1

NO\_IGUAL\_ADC\_10:

JC MENOR\_1

CJNE VALOR\_ADC\_1, #0x6C, NO\_IGUAL\_ADC\_11

SJMP EN\_RANGO\_1

NO\_IGUAL\_ADC\_11:

JC EN\_RANGO\_1

; SPOS\_RV > 42.5

SETB EV12 ; EVENTO 12

SJMP FIN\_ADR1

EN\_RANGO\_1: ; 37.5 <= SPOS\_RV <= 40

SETB EV25 ; EVENTO 25

SJMP FIN\_ADR1

MENOR\_1: ; SPOS\_RV < 37.5

SETB EV13 ; EVENTO 13

SJMP FIN\_ADR1

FIN\_ADR1:

CLR tick\_ADCR1

RODIILLOS\_POSICION:

; Verificaci n para flag RH\_40cm y RV\_40cm

JNB RH\_40cm, SELECTOR\_DE\_PRIORIDAD

JNB RV\_40cm, SELECTOR\_DE\_PRIORIDAD

SETB EV11

CLR RH\_40cm

CLR RV\_40cm

SELECTOR\_DE\_PRIORIDAD:

; SELECTOR DE PRIORIDAD DE LOS FLAGS DE EVENTOS

JNB EV1, EV2\_GV ; EVENTO 1 ACTIVADO

MOV EVENTO, #0x01

CLR EV1

LJMP FIN\_GV

EV2\_GV:

JNB EV2, EV3\_GV ; EVENTO 2 ACTIVADO

MOV EVENTO, #0x02

CLR EV2

LJMP FIN\_GV

EV3\_GV:

JNB EV3, EV4\_GV ; EVENTO 3 ACTIVADO

MOV EVENTO, #0x03

CLR EV3

LJMP FIN\_GV

EV4\_GV:

JNB EV4, EV5\_GV ; EVENTO 4 ACTIVADO

MOV EVENTO, #0x04

CLR EV4

LJMP FIN\_GV

EV5\_GV:

JNB EV5, EV6\_GV ; EVENTO 5 ACTIVADO

MOV EVENTO, #0x05

CLR EV5

LJMP FIN\_GV

EV6\_GV:

JNB EV6, EV7\_GV ; EVENTO 6 ACTIVADO

MOV EVENTO, #0x06

CLR EV6

AJMP FIN\_GV

EV7\_GV:

JNB EV7, EV8\_GV ; EVENTO 7 ACTIVADO

MOV EVENTO, #0x07

CLR EV7

AJMP FIN\_GV

EV8\_GV:

JNB EV8, EV9\_GV ; EVENTO 8 ACTIVADO

MOV EVENTO, #0x08

CLR EV8

AJMP FIN\_GV

EV9\_GV:

JNB EV9, EV10\_GV ; EVENTO 9 ACTIVADO

MOV EVENTO, #0x09

CLR EV9

AJMP FIN\_GV

EV10\_GV:

JNB EV10, EV11\_GV ; EVENTO 10 ACTIVADO

MOV EVENTO, #0x0A

CLR EV10

AJMP FIN\_GV

EV11\_GV:

JNB EV11, EV12\_GV ; EVENTO 11 ACTIVADO

MOV EVENTO, #0x0B

CLR EV11

AJMP FIN\_GV

EV12\_GV:

JNB EV12, EV13\_GV ; EVENTO 12 ACTIVADO

MOV EVENTO, #0x0C

CLR EV12

AJMP FIN\_GV

EV13\_GV:

JNB EV13, EV14\_GV ; EVENTO 13 ACTIVADO

MOV EVENTO, #0x0D

CLR EV13

AJMP FIN\_GV

EV14\_GV:

JNB EV14, EV15\_GV ; EVENTO 14 ACTIVADO

MOV EVENTO, #0x0E

CLR EV14

SJMP FIN\_GV

EV15\_GV:

JNB EV15, EV16\_GV ; EVENTO 15 ACTIVADO

MOV EVENTO, #0x0F

CLR EV15

SJMP FIN\_GV

EV16\_GV:

JNB EV16, EV17\_GV ; EVENTO 16 ACTIVADO

MOV EVENTO, #0x10

CLR EV16

SJMP FIN\_GV

EV17\_GV:

JNB EV17, EV18\_GV ; EVENTO 17 ACTIVADO

MOV EVENTO, #0x11

CLR EV17

SJMP FIN\_GV

EV18\_GV:

JNB EV18, EV19\_GV ; EVENTO 18 ACTIVADO

MOV EVENTO, #0x12

CLR EV18

SJMP FIN\_GV

EV19\_GV:

JNB EV19, EV20\_GV ; EVENTO 19 ACTIVADO

MOV EVENTO, #0x13

CLR EV19

SJMP FIN\_GV

EV20\_GV:

JNB EV20, EV21\_GV ; EVENTO 20 ACTIVADO

MOV EVENTO, #0x14

CLR EV20

SJMP FIN\_GV

EV21\_GV:

JNB EV21, EV22\_GV ; EVENTO 21 ACTIVADO

MOV EVENTO, #0x15

CLR EV21

SJMP FIN\_GV

EV22\_GV:

JNB EV22, EV23\_GV ; EVENTO 22 ACTIVADO

MOV EVENTO, #0x16

CLR EV22

SJMP FIN\_GV

EV23\_GV:

JNB EV23, EV24\_GV ; EVENTO 23 ACTIVADO

MOV EVENTO, #0x17

CLR EV23

SJMP FIN\_GV

EV24\_GV:

JNB EV24, EV25\_GV ; EVENTO 24 ACTIVADO

MOV EVENTO, #0x18

CLR EV24

SJMP FIN\_GV

EV25\_GV:

JNB EV25, EV26\_GV ; EVENTO 25 ACTIVADO

MOV EVENTO, #0x19

CLR EV25

SJMP FIN\_GV

EV26\_GV:

JNB EV26, EV0\_GV ; EVENTO 26 ACTIVADO

MOV EVENTO, #0x1A

CLR EV26

SJMP FIN\_GV

EV0\_GV:

MOV EVENTO, #0x00

FIN\_GV:

RET

;#########################################################################

; ESTADO 0

;#########################################################################

ESPERA\_COCHE:

;MAQ\_EVENTOS\_0:

MOV A, EVENTO

RL A

MOV DPTR, #LISTA\_EV\_0

JMP @A+DPTR

LISTA\_EV\_0:

AJMP EVENTOS0\_0

AJMP EVENTOS0\_1

AJMP EVENTOS0\_2

AJMP EVENTOS0\_3

AJMP EVENTOS0\_4

AJMP EVENTOS0\_5

AJMP EVENTOS0\_6

AJMP EVENTOS0\_7

AJMP EVENTOS0\_8

AJMP EVENTOS0\_9

AJMP EVENTOS0\_10

AJMP EVENTOS0\_11

AJMP EVENTOS0\_12

AJMP EVENTOS0\_13

AJMP EVENTOS0\_14

AJMP EVENTOS0\_15

AJMP EVENTOS0\_16

AJMP EVENTOS0\_17

AJMP EVENTOS0\_18

AJMP EVENTOS0\_19

AJMP EVENTOS0\_20

AJMP EVENTOS0\_21

AJMP EVENTOS0\_22

AJMP EVENTOS0\_23

AJMP EVENTOS0\_24

AJMP EVENTOS0\_25

AJMP EVENTOS0\_26

EVENTOS0\_0:

RET

EVENTOS0\_1:

MOV ESTADO, #0x01

ACALL ACTIVAR\_TIMER\_0 ; AQUI SE INICIA EL TIMER DE 30s (SI PASAN 30s --> IR AL EVENTO 4)

RET

EVENTOS0\_2:

SETB SEM\_ROJO

CLR SEM\_VERDE

ACALL ACTIVAR\_TIMER\_0 ; AQUI SE INICIA EL TIMER DE 5s (SI PASAN 5s --> IR AL EVENTO 1)

RET

EVENTOS0\_3:

SETB SEM\_VERDE

CLR SEM\_ROJO

ACALL DESACTIVAR\_TIMER\_0 ; Reinicio de los valores del timer

RET

EVENTOS0\_4:

RET

EVENTOS0\_5:

RET

EVENTOS0\_6:

RET

EVENTOS0\_7:

RET

EVENTOS0\_8:

RET

EVENTOS0\_9:

RET

EVENTOS0\_10:

RET

EVENTOS0\_11:

RET

EVENTOS0\_12:

RET

EVENTOS0\_13:

RET

EVENTOS0\_14:

RET

EVENTOS0\_15:

RET

EVENTOS0\_16:

RET

EVENTOS0\_17:

RET

EVENTOS0\_18:

RET

EVENTOS0\_19:

RET

EVENTOS0\_20:

RET

EVENTOS0\_21:

RET

EVENTOS0\_22:

RET

EVENTOS0\_23:

RET

EVENTOS0\_24:

RET

EVENTOS0\_25:

RET

EVENTOS0\_26:

RET

;#########################################################################

; ESTADO 1

;#########################################################################

COCHE\_PUESTO:

;MAQ\_EVENTOS\_1:

MOV A, EVENTO

RL A

MOV DPTR, #LISTA\_EV\_1

JMP @A+DPTR

LISTA\_EV\_1:

AJMP EVENTOS1\_0

AJMP EVENTOS1\_1

AJMP EVENTOS1\_2

AJMP EVENTOS1\_3

AJMP EVENTOS1\_4

AJMP EVENTOS1\_5

AJMP EVENTOS1\_6

AJMP EVENTOS1\_7

AJMP EVENTOS1\_8

AJMP EVENTOS1\_9

AJMP EVENTOS1\_10

AJMP EVENTOS1\_11

AJMP EVENTOS1\_12

AJMP EVENTOS1\_13

AJMP EVENTOS1\_14

AJMP EVENTOS1\_15

AJMP EVENTOS1\_16

AJMP EVENTOS1\_17

AJMP EVENTOS1\_18

AJMP EVENTOS1\_19

AJMP EVENTOS1\_20

AJMP EVENTOS1\_21

AJMP EVENTOS1\_22

AJMP EVENTOS1\_23

AJMP EVENTOS1\_24

AJMP EVENTOS1\_25

AJMP EVENTOS1\_26

EVENTOS1\_0:

RET

EVENTOS1\_1:

RET

EVENTOS1\_2:

RET

EVENTOS1\_3:

RET

EVENTOS1\_4:

MOV ESTADO, #0x02

SETB AL\_START

ACALL ACTIVAR\_TIMER\_0

;AQUI HAY QUE EMPEZAR OTRO TIMER DE 30s (SI PASAN 30s y S\_FICHA --> IR AL EVENTO 7)

RET

EVENTOS1\_5:

JNB FICHA\_TIPO, NORMAL\_1

SETB LED\_LINT

SJMP INTENSIVO\_1

NORMAL\_1:

SETB LED\_LNOR

INTENSIVO\_1:

RET

EVENTOS1\_6:

ACALL DESACTIVAR\_TIMER\_0

MOV ESTADO, #0x03

SETB BMOV\_FRONT

RET

EVENTOS1\_7:

RET

EVENTOS1\_8:

RET

EVENTOS1\_9:

RET

EVENTOS1\_10:

RET

EVENTOS1\_11:

RET

EVENTOS1\_12:

RET

EVENTOS1\_13:

RET

EVENTOS1\_14:

RET

EVENTOS1\_15:

RET

EVENTOS1\_16:

RET

EVENTOS1\_17:

RET

EVENTOS1\_18:

RET

EVENTOS1\_19:

RET

EVENTOS1\_20:

RET

EVENTOS1\_21:

RET

EVENTOS1\_22:

RET

EVENTOS1\_23:

RET

EVENTOS1\_24:

RET

EVENTOS1\_25:

RET

EVENTOS1\_26:

RET

;#########################################################################

; ESTADO 2

;#########################################################################

ALARMA:

;MAQ\_EVENTOS\_2:

MOV A, EVENTO

RL A

MOV DPTR, #LISTA\_EV\_2

JMP @A+DPTR

LISTA\_EV\_2:

AJMP EVENTOS2\_0

AJMP EVENTOS2\_1

AJMP EVENTOS2\_2

AJMP EVENTOS2\_3

AJMP EVENTOS2\_4

AJMP EVENTOS2\_5

AJMP EVENTOS2\_6

AJMP EVENTOS2\_7

AJMP EVENTOS2\_8

AJMP EVENTOS2\_9

AJMP EVENTOS2\_10

AJMP EVENTOS2\_11

AJMP EVENTOS2\_12

AJMP EVENTOS2\_13

AJMP EVENTOS2\_14

AJMP EVENTOS2\_15

AJMP EVENTOS2\_16

AJMP EVENTOS2\_17

AJMP EVENTOS2\_18

AJMP EVENTOS2\_19

AJMP EVENTOS2\_20

AJMP EVENTOS2\_21

AJMP EVENTOS2\_22

AJMP EVENTOS2\_23

AJMP EVENTOS2\_24

AJMP EVENTOS2\_25

AJMP EVENTOS2\_26

EVENTOS2\_0:

RET

EVENTOS2\_1:

RET

EVENTOS2\_2:

RET

EVENTOS2\_3:

MOV ESTADO, #0x00

CLR AL\_START

SETB SEM\_VERDE

CLR SEM\_ROJO

RET

EVENTOS2\_4:

RET

EVENTOS2\_5:

JNB FICHA\_TIPO, NORMAL\_2

SETB LED\_LNOR

SJMP INTENSIVO\_2

NORMAL\_2:

SETB LED\_LINT

INTENSIVO\_2:

RET

EVENTOS2\_6:

ACALL DESACTIVAR\_TIMER\_0

MOV ESTADO, #0x03

SETB BMOV\_FRONT

RET

EVENTOS2\_7:

ACALL DESACTIVAR\_TIMER\_0

MOV ESTADO, #0x03

SETB BMOV\_FRONT

RET

EVENTOS2\_8:

RET

EVENTOS2\_9:

RET

EVENTOS2\_10:

RET

EVENTOS2\_11:

RET

EVENTOS2\_12:

RET

EVENTOS2\_13:

RET

EVENTOS2\_14:

RET

EVENTOS2\_15:

RET

EVENTOS2\_16:

RET

EVENTOS2\_17:

RET

EVENTOS2\_18:

RET

EVENTOS2\_19:

RET

EVENTOS2\_20:

RET

EVENTOS2\_21:

RET

EVENTOS2\_22:

RET

EVENTOS2\_23:

RET

EVENTOS2\_24:

RET

EVENTOS2\_25:

RET

EVENTOS2\_26:

RET

;#########################################################################

; ESTADO 3

;#########################################################################

PONER\_PUENTE:

;MAQ\_EVENTOS\_3:

MOV A, EVENTO

RL A

MOV DPTR, #LISTA\_EV\_3

JMP @A+DPTR

LISTA\_EV\_3:

AJMP EVENTOS3\_0

AJMP EVENTOS3\_1

AJMP EVENTOS3\_2

AJMP EVENTOS3\_3

AJMP EVENTOS3\_4

AJMP EVENTOS3\_5

AJMP EVENTOS3\_6

AJMP EVENTOS3\_7

AJMP EVENTOS3\_8

AJMP EVENTOS3\_9

AJMP EVENTOS3\_10

AJMP EVENTOS3\_11

AJMP EVENTOS3\_12

AJMP EVENTOS3\_13

AJMP EVENTOS3\_14

AJMP EVENTOS3\_15

AJMP EVENTOS3\_16

AJMP EVENTOS3\_17

AJMP EVENTOS3\_18

AJMP EVENTOS3\_19

AJMP EVENTOS3\_20

AJMP EVENTOS3\_21

AJMP EVENTOS3\_22

AJMP EVENTOS3\_23

AJMP EVENTOS3\_24

AJMP EVENTOS3\_25

AJMP EVENTOS3\_26

EVENTOS3\_0:

RET

EVENTOS3\_1:

RET

EVENTOS3\_2:

RET

EVENTOS3\_3:

RET

EVENTOS3\_4:

RET

EVENTOS3\_5:

RET

EVENTOS3\_6:

RET

EVENTOS3\_7:

RET

EVENTOS3\_8:

MOV ESTADO, #0x04

SETB EV\_AGUA

SETB EV\_JABON

RET

EVENTOS3\_9:

RET

EVENTOS3\_10:

RET

EVENTOS3\_11:

RET

EVENTOS3\_12:

RET

EVENTOS3\_13:

RET

EVENTOS3\_14:

RET

EVENTOS3\_15:

RET

EVENTOS3\_16:

RET

EVENTOS3\_17:

RET

EVENTOS3\_18:

RET

EVENTOS3\_19:

RET

EVENTOS3\_20:

RET

EVENTOS3\_21:

RET

EVENTOS3\_22:

RET

EVENTOS3\_23:

RET

EVENTOS3\_24:

RET

EVENTOS3\_25:

RET

EVENTOS3\_26:

RET

;#########################################################################

; ESTADO 4

;#########################################################################

ENJABONADO:

ACALL MAQ\_EVENTOS\_4

RET

MAQ\_EVENTOS\_4:

MOV A, EVENTO

RL A

MOV DPTR, #LISTA\_EV\_4

JMP @A+DPTR

LISTA\_EV\_4:

AJMP EVENTOS4\_0

AJMP EVENTOS4\_1

AJMP EVENTOS4\_2

AJMP EVENTOS4\_3

AJMP EVENTOS4\_4

AJMP EVENTOS4\_5

AJMP EVENTOS4\_6

AJMP EVENTOS4\_7

AJMP EVENTOS4\_8

AJMP EVENTOS4\_9

AJMP EVENTOS4\_10

AJMP EVENTOS4\_11

AJMP EVENTOS4\_12

AJMP EVENTOS4\_13

AJMP EVENTOS4\_14

AJMP EVENTOS4\_15

AJMP EVENTOS4\_16

AJMP EVENTOS4\_17

AJMP EVENTOS4\_18

AJMP EVENTOS4\_19

AJMP EVENTOS4\_20

AJMP EVENTOS4\_21

AJMP EVENTOS4\_22

AJMP EVENTOS4\_23

AJMP EVENTOS4\_24

AJMP EVENTOS4\_25

AJMP EVENTOS4\_26

EVENTOS4\_0:

RET

EVENTOS4\_1:

RET

EVENTOS4\_2:

RET

EVENTOS4\_3:

RET

EVENTOS4\_4:

RET

EVENTOS4\_5:

RET

EVENTOS4\_6:

RET

EVENTOS4\_7:

RET

EVENTOS4\_8:

RET

EVENTOS4\_9:

CLR EV\_AGUA

CLR EV\_JABON

ACALL ACTIVAR\_TIMER\_0; avanza 40cm mas: AQUI SE INICIA UNA TIMER DE 4s (CUANDO PASEN 4s --> IR AL EVENTO 10)

RET

EVENTOS4\_10:

ACALL DESACTIVAR\_TIMER\_0

MOV ESTADO, #0x05

JB LED\_LNOR, LAV\_NORMAL\_R

SJMP LAV\_INTENSIVO\_R

LAV\_NORMAL\_R:

MOV PWM0, #0x80 ; %50 del PWM0

SJMP RODILLO\_ACTIVADO

LAV\_INTENSIVO\_R:

MOV PWM0, #0x00 ; %100 del PWM0

RODILLO\_ACTIVADO:

SETB RV\_INTO

RET

EVENTOS4\_11:

RET

EVENTOS4\_12:

RET

EVENTOS4\_13:

RET

EVENTOS4\_14:

RET

EVENTOS4\_15:

RET

EVENTOS4\_16:

RET

EVENTOS4\_17:

RET

EVENTOS4\_18:

RET

EVENTOS4\_19:

RET

EVENTOS4\_20:

RET

EVENTOS4\_21:

RET

EVENTOS4\_22:

RET

EVENTOS4\_23:

RET

EVENTOS4\_24:

RET

EVENTOS4\_25:

RET

EVENTOS4\_26:

RET

;#########################################################################

; ESTADO 5

;#########################################################################

LIMPIAR\_ATRAS:

;MAQ\_EVENTOS\_5:

MOV A, EVENTO

RL A

MOV DPTR, #LISTA\_EV\_5

JMP @A+DPTR

LISTA\_EV\_5:

AJMP EVENTOS5\_0

AJMP EVENTOS5\_1

AJMP EVENTOS5\_2

AJMP EVENTOS5\_3

AJMP EVENTOS5\_4

AJMP EVENTOS5\_5

AJMP EVENTOS5\_6

AJMP EVENTOS5\_7

AJMP EVENTOS5\_8

AJMP EVENTOS5\_9

AJMP EVENTOS5\_10

AJMP EVENTOS5\_11

AJMP EVENTOS5\_12

AJMP EVENTOS5\_13

AJMP EVENTOS5\_14

AJMP EVENTOS5\_15

AJMP EVENTOS5\_16

AJMP EVENTOS5\_17

AJMP EVENTOS5\_18

AJMP EVENTOS5\_19

AJMP EVENTOS5\_20

AJMP EVENTOS5\_21

AJMP EVENTOS5\_22

AJMP EVENTOS5\_23

AJMP EVENTOS5\_24

AJMP EVENTOS5\_25

AJMP EVENTOS5\_26

EVENTOS5\_0:

RET

EVENTOS5\_1:

RET

EVENTOS5\_2:

RET

EVENTOS5\_3:

RET

EVENTOS5\_4:

RET

EVENTOS5\_5:

RET

EVENTOS5\_6:

RET

EVENTOS5\_7:

RET

EVENTOS5\_8:

RET

EVENTOS5\_9:

RET

EVENTOS5\_10:

RET

EVENTOS5\_11:

RET

EVENTOS5\_12:

RET

EVENTOS5\_13:

RET

EVENTOS5\_14:

RET

EVENTOS5\_15:

RET

EVENTOS5\_16:

CLR RV\_OUTTO

MOV ESTADO, #0x06

SETB BMOV\_BACK

ACALL ACTIVAR\_TIMER\_0

SETB EAD ; HABILITAR INTERRUPCIONES DEL ADC

RET

EVENTOS5\_17:

CLR RV\_INTO

SETB RV\_OUTTO

RET

EVENTOS5\_18:

RET

EVENTOS5\_19:

RET

EVENTOS5\_20:

RET

EVENTOS5\_21:

RET

EVENTOS5\_22:

RET

EVENTOS5\_23:

RET

EVENTOS5\_24:

RET

EVENTOS5\_25:

RET

EVENTOS5\_26:

RET

;#########################################################################

; ESTADO 6

;#########################################################################

PONER\_RODILLOS:

;MAQ\_EVENTOS\_6:

MOV A, EVENTO

RL A

MOV DPTR, #LISTA\_EV\_6

JMP @A+DPTR

LISTA\_EV\_6:

AJMP EVENTOS6\_0

AJMP EVENTOS6\_1

AJMP EVENTOS6\_2

AJMP EVENTOS6\_3

AJMP EVENTOS6\_4

AJMP EVENTOS6\_5

AJMP EVENTOS6\_6

AJMP EVENTOS6\_7

AJMP EVENTOS6\_8

AJMP EVENTOS6\_9

AJMP EVENTOS6\_10

AJMP EVENTOS6\_11

AJMP EVENTOS6\_12

AJMP EVENTOS6\_13

AJMP EVENTOS6\_14

AJMP EVENTOS6\_15

AJMP EVENTOS6\_16

AJMP EVENTOS6\_17

AJMP EVENTOS6\_18

AJMP EVENTOS6\_19

AJMP EVENTOS6\_20

AJMP EVENTOS6\_21

AJMP EVENTOS6\_22

AJMP EVENTOS6\_23

AJMP EVENTOS6\_24

AJMP EVENTOS6\_25

AJMP EVENTOS6\_26

EVENTOS6\_0:

RET

EVENTOS6\_1:

RET

EVENTOS6\_2:

RET

EVENTOS6\_3:

RET

EVENTOS6\_4:

RET

EVENTOS6\_5:

RET

EVENTOS6\_6:

RET

EVENTOS6\_7:

RET

EVENTOS6\_8:

SETB RH\_DOWNTO

SETB RV\_INTO

CLR BMOV\_BACK

RET

EVENTOS6\_9:

RET

EVENTOS6\_10:

RET

EVENTOS6\_11:

SETB BMOV\_BACK

MOV ESTADO, #0x07

RET

EVENTOS6\_12:

RET

EVENTOS6\_13:

RET

EVENTOS6\_14:

RET

EVENTOS6\_15:

RET

EVENTOS6\_16:

RET

EVENTOS6\_17:

RET

EVENTOS6\_18:

RET

EVENTOS6\_19:

RET

EVENTOS6\_20:

RET

EVENTOS6\_21:

RET

EVENTOS6\_22:

RET

EVENTOS6\_23:

RET

EVENTOS6\_24:

CLR RH\_DOWNTO

SETB RH\_40cm

RET

EVENTOS6\_25:

CLR RV\_INTO

SETB RV\_40CM

RET

EVENTOS6\_26:

MOV ADCON, #0x08 ; activa adcs y empieza la conversion en P5.0

RET

;#########################################################################

; ESTADO 7

;#########################################################################

RODILLOS:

;MAQ\_EVENTOS\_7:

MOV A, EVENTO

RL A

MOV DPTR, #LISTA\_EV\_7

JMP @A+DPTR

LISTA\_EV\_7:

AJMP EVENTOS7\_0

AJMP EVENTOS7\_1

AJMP EVENTOS7\_2

AJMP EVENTOS7\_3

AJMP EVENTOS7\_4

AJMP EVENTOS7\_5

AJMP EVENTOS7\_6

AJMP EVENTOS7\_7

AJMP EVENTOS7\_8

AJMP EVENTOS7\_9

AJMP EVENTOS7\_10

AJMP EVENTOS7\_11

AJMP EVENTOS7\_12

AJMP EVENTOS7\_13

AJMP EVENTOS7\_14

AJMP EVENTOS7\_15

AJMP EVENTOS7\_16

AJMP EVENTOS7\_17

AJMP EVENTOS7\_18

AJMP EVENTOS7\_19

AJMP EVENTOS7\_20

AJMP EVENTOS7\_21

AJMP EVENTOS7\_22

AJMP EVENTOS7\_23

AJMP EVENTOS7\_24

AJMP EVENTOS7\_25

AJMP EVENTOS7\_26

EVENTOS7\_0:

RET

EVENTOS7\_1:

RET

EVENTOS7\_2:

RET

EVENTOS7\_3:

RET

EVENTOS7\_4:

RET

EVENTOS7\_5:

RET

EVENTOS7\_6:

RET

EVENTOS7\_7:

RET

EVENTOS7\_8:

RET

EVENTOS7\_9:

CLR EAD

ACALL DESACTIVAR\_TIMER\_0

CLR BMOV\_BACK

CLR RH\_DOWNTO

CLR RV\_INTO

CLR RV\_OUTTO

SETB RH\_UPTO

RET

EVENTOS7\_10:

RET

EVENTOS7\_11:

RET

EVENTOS7\_12:

CLR RV\_OUTTO

SETB RV\_INTO

RET

EVENTOS7\_13:

CLR RV\_INTO

SETB RV\_OUTTO

RET

EVENTOS7\_14:

CLR RH\_DOWNTO

SETB RH\_UPTO

RET

EVENTOS7\_15:

CLR RH\_UPTO

SETB RH\_DOWNTO

RET

EVENTOS7\_16:

RET

EVENTOS7\_17:

RET

EVENTOS7\_18:

RET

EVENTOS7\_19:

CLR RH\_UPTO

SETB BMOV\_BACK

ACALL ACTIVAR\_TIMER\_0 ; AQUI SE INICIA UNA TIMER DE 4s (CUANDO PASEN 4s --> IR AL EVENTO 10)

MOV ESTADO, #0x08

RET

EVENTOS7\_20:

RET

EVENTOS7\_21:

RET

EVENTOS7\_22:

RET

EVENTOS7\_23:

RET

EVENTOS7\_24:

CLR RH\_UPTO

CLR RH\_DOWNTO

RET

EVENTOS7\_25:

CLR RV\_OUTTO

CLR RV\_INTO

RET

EVENTOS7\_26:

MOV ADCON, #0x08 ; activa adcs y empieza la conversion en P5.0

RET

;#########################################################################

; ESTADO 8

;#########################################################################

LIMPIAR\_DELANTE:

;MAQ\_EVENTOS\_8:

MOV A, EVENTO

RL A

MOV DPTR, #LISTA\_EV\_8

JMP @A+DPTR

LISTA\_EV\_8:

AJMP EVENTOS8\_0

AJMP EVENTOS8\_1

AJMP EVENTOS8\_2

AJMP EVENTOS8\_3

AJMP EVENTOS8\_4

AJMP EVENTOS8\_5

AJMP EVENTOS8\_6

AJMP EVENTOS8\_7

AJMP EVENTOS8\_8

AJMP EVENTOS8\_9

AJMP EVENTOS8\_10

AJMP EVENTOS8\_11

AJMP EVENTOS8\_12

AJMP EVENTOS8\_13

AJMP EVENTOS8\_14

AJMP EVENTOS8\_15

AJMP EVENTOS8\_16

AJMP EVENTOS8\_17

AJMP EVENTOS8\_18

AJMP EVENTOS8\_19

AJMP EVENTOS8\_20

AJMP EVENTOS8\_21

AJMP EVENTOS8\_22

AJMP EVENTOS8\_23

AJMP EVENTOS8\_24

AJMP EVENTOS8\_25

AJMP EVENTOS8\_26

EVENTOS8\_0:

RET

EVENTOS8\_1:

RET

EVENTOS8\_2:

RET

EVENTOS8\_3:

RET

EVENTOS8\_4:

RET

EVENTOS8\_5:

RET

EVENTOS8\_6:

RET

EVENTOS8\_7:

RET

EVENTOS8\_8:

RET

EVENTOS8\_9:

RET

EVENTOS8\_10:

CLR BMOV\_BACK

ACALL DESACTIVAR\_TIMER\_0

SETB RV\_INTO

RET

EVENTOS8\_11:

RET

EVENTOS8\_12:

RET

EVENTOS8\_13:

RET

EVENTOS8\_14:

RET

EVENTOS8\_15:

RET

EVENTOS8\_16:

CLR RV\_OUTTO

MOV PWM0, #0xFF

MOV ESTADO, #0x09

SETB BMOV\_FRONT

SETB EV\_AGUA

RET

EVENTOS8\_17:

CLR RV\_INTO

SETB RV\_OUTTO

RET

EVENTOS8\_18:

RET

EVENTOS8\_19:

RET

EVENTOS8\_20:

RET

EVENTOS8\_21:

RET

EVENTOS8\_22:

RET

EVENTOS8\_23:

RET

EVENTOS8\_24:

RET

EVENTOS8\_25:

RET

EVENTOS8\_26:

RET

;#########################################################################

; ESTADO 9

;#########################################################################

ACLARADO:

;MAQ\_EVENTOS\_9:

MOV A, EVENTO

RL A

MOV DPTR, #LISTA\_EV\_9

JMP @A+DPTR

LISTA\_EV\_9:

AJMP EVENTOS9\_0

AJMP EVENTOS9\_1

AJMP EVENTOS9\_2

AJMP EVENTOS9\_3

AJMP EVENTOS9\_4

AJMP EVENTOS9\_5

AJMP EVENTOS9\_6

AJMP EVENTOS9\_7

AJMP EVENTOS9\_8

AJMP EVENTOS9\_9

AJMP EVENTOS9\_10

AJMP EVENTOS9\_11

AJMP EVENTOS9\_12

AJMP EVENTOS9\_13

AJMP EVENTOS9\_14

AJMP EVENTOS9\_15

AJMP EVENTOS9\_16

AJMP EVENTOS9\_17

AJMP EVENTOS9\_18

AJMP EVENTOS9\_19

AJMP EVENTOS9\_20

AJMP EVENTOS9\_21

AJMP EVENTOS9\_22

AJMP EVENTOS9\_23

AJMP EVENTOS9\_24

AJMP EVENTOS9\_25

AJMP EVENTOS9\_26

EVENTOS9\_0:

RET

EVENTOS9\_1:

RET

EVENTOS9\_2:

RET

EVENTOS9\_3:

RET

EVENTOS9\_4:

RET

EVENTOS9\_5:

RET

EVENTOS9\_6:

RET

EVENTOS9\_7:

RET

EVENTOS9\_8:

SETB EV\_AGUA

RET

EVENTOS9\_9:

CLR EV\_AGUA

ACALL ACTIVAR\_TIMER\_0 ; AQUI SE INICIA UNA TIMER DE 4s (CUANDO PASEN 4s --> IR AL EVENTO 10)

RET

EVENTOS9\_10:

ACALL DESACTIVAR\_TIMER\_0

CLR BMOV\_FRONT

SETB BMOV\_BACK

MOV ESTADO, #0x0A

RET

EVENTOS9\_11:

RET

EVENTOS9\_12:

RET

EVENTOS9\_13:

RET

EVENTOS9\_14:

RET

EVENTOS9\_15:

RET

EVENTOS9\_16:

RET

EVENTOS9\_17:

RET

EVENTOS9\_18:

RET

EVENTOS9\_19:

RET

EVENTOS9\_20:

RET

EVENTOS9\_21:

RET

EVENTOS9\_22:

RET

EVENTOS9\_23:

RET

EVENTOS9\_24:

RET

EVENTOS9\_25:

RET

EVENTOS9\_26:

RET

;#########################################################################

; ESTADO 10

;#########################################################################

SECADO:

;MAQ\_EVENTOS\_10:

MOV A, EVENTO

RL A

MOV DPTR, #LISTA\_EV\_10

JMP @A+DPTR

LISTA\_EV\_10:

AJMP EVENTOS10\_0

AJMP EVENTOS10\_1

AJMP EVENTOS10\_2

AJMP EVENTOS10\_3

AJMP EVENTOS10\_4

AJMP EVENTOS10\_5

AJMP EVENTOS10\_6

AJMP EVENTOS10\_7

AJMP EVENTOS10\_8

AJMP EVENTOS10\_9

AJMP EVENTOS10\_10

AJMP EVENTOS10\_11

AJMP EVENTOS10\_12

AJMP EVENTOS10\_13

AJMP EVENTOS10\_14

AJMP EVENTOS10\_15

AJMP EVENTOS10\_16

AJMP EVENTOS10\_17

AJMP EVENTOS10\_18

AJMP EVENTOS10\_19

AJMP EVENTOS10\_20

AJMP EVENTOS10\_21

AJMP EVENTOS10\_22

AJMP EVENTOS10\_23

AJMP EVENTOS10\_24

AJMP EVENTOS10\_25

AJMP EVENTOS10\_26

EVENTOS10\_0:

RET

EVENTOS10\_1:

RET

EVENTOS10\_2:

RET

EVENTOS10\_3:

RET

EVENTOS10\_4:

RET

EVENTOS10\_5:

RET

EVENTOS10\_6:

RET

EVENTOS10\_7:

RET

EVENTOS10\_8:

SJMP LAV\_INTENSIVO\_V

LAV\_NORMAL\_V:

MOV PWM1, #0x80 ; %50 del PWM1

SJMP VENTILADOR\_ACTIVADO

LAV\_INTENSIVO\_V:

MOV PWM1, #0xD9 ; %85 del PWM1

VENTILADOR\_ACTIVADO:

RET

EVENTOS10\_9:

MOV PWM1, #0xFF

RET

EVENTOS10\_10:

RET

EVENTOS10\_11:

RET

EVENTOS10\_12:

RET

EVENTOS10\_13:

RET

EVENTOS10\_14:

RET

EVENTOS10\_15:

RET

EVENTOS10\_16:

RET

EVENTOS10\_17:

RET

EVENTOS10\_18:

RET

EVENTOS10\_19:

RET

EVENTOS10\_20:

CLR SEM\_ROJO

SETB SEM\_VERDE

MOV ESTADO, #0x0B

ACALL ACTIVAR\_TIMER\_0

RET

EVENTOS10\_21:

RET

EVENTOS10\_22:

RET

EVENTOS10\_23:

RET

EVENTOS10\_24:

RET

EVENTOS10\_25:

RET

EVENTOS10\_26:

RET

;#########################################################################

; ESTADO 11

;#########################################################################

RETIRAR\_COCHE:

;MAQ\_EVENTOS\_11:

MOV A, EVENTO

RL A

MOV DPTR, #LISTA\_EV\_11

JMP @A+DPTR

LISTA\_EV\_11:

SJMP EVENTOS11\_0

SJMP EVENTOS11\_1

SJMP EVENTOS11\_2

SJMP EVENTOS11\_3

SJMP EVENTOS11\_4

SJMP EVENTOS11\_5

SJMP EVENTOS11\_6

SJMP EVENTOS11\_7

SJMP EVENTOS11\_8

SJMP EVENTOS11\_9

SJMP EVENTOS11\_10

SJMP EVENTOS11\_11

SJMP EVENTOS11\_12

SJMP EVENTOS11\_13

SJMP EVENTOS11\_14

SJMP EVENTOS11\_15

SJMP EVENTOS11\_16

SJMP EVENTOS11\_17

SJMP EVENTOS11\_18

SJMP EVENTOS11\_19

SJMP EVENTOS11\_20

SJMP EVENTOS11\_21

SJMP EVENTOS11\_22

SJMP EVENTOS11\_23

SJMP EVENTOS11\_24

SJMP EVENTOS11\_25

SJMP EVENTOS11\_26

EVENTOS11\_0:

RET

EVENTOS11\_1:

RET

EVENTOS11\_2:

RET

EVENTOS11\_3:

SETB SEM\_VERDE

MOV ESTADO, #0x00

RET

EVENTOS11\_4:

RET

EVENTOS11\_5:

RET

EVENTOS11\_6:

RET

EVENTOS11\_7:

RET

EVENTOS11\_8:

RET

EVENTOS11\_9:

RET

EVENTOS11\_10:

RET

EVENTOS11\_11:

RET

EVENTOS11\_12:

RET

EVENTOS11\_13:

RET

EVENTOS11\_14:

RET

EVENTOS11\_15:

RET

EVENTOS11\_16:

RET

EVENTOS11\_17:

RET

EVENTOS11\_18:

RET

EVENTOS11\_19:

RET

EVENTOS11\_20:

RET

EVENTOS11\_21:

JB SEM\_VERDE, EN\_VERDE

SETB SEM\_VERDE

SJMP NO\_VERDE

EN\_VERDE:

CLR SEM\_VERDE

NO\_VERDE:

RET

EVENTOS11\_22:

RET

EVENTOS11\_23:

;acciones

RET

EVENTOS11\_24:

RET

EVENTOS11\_25:

RET

EVENTOS11\_26:

RET

END