

Arquitectura de Computadores

Práctica Final de Laboratorio 2021-2022

Departamento:

Tecnología Electrónica

Titulación:

Grado en Informática de Gestión y Sistemas de Información

2º Curso (1º Cuatrimestre)

Septiembre de 2021

Velasco Prieto, Álvaro

Elorza Gabilondo, David

Contenido

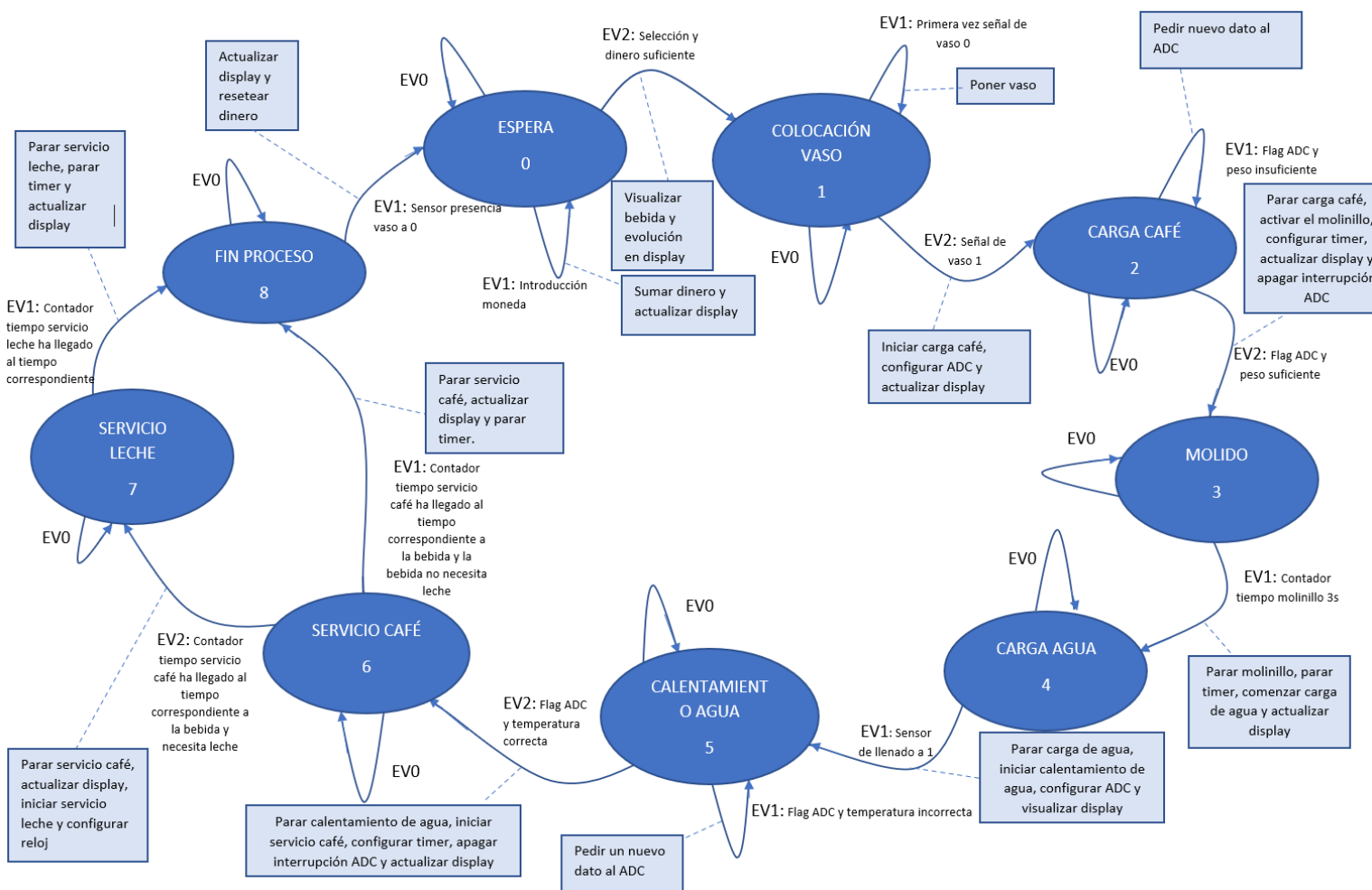
Contenido	1
1.Introducción	2
2.Diagramas de estados/eventos/acciones y de flujo	2
3.Cálculos y Comentarios	10
4.Código	11

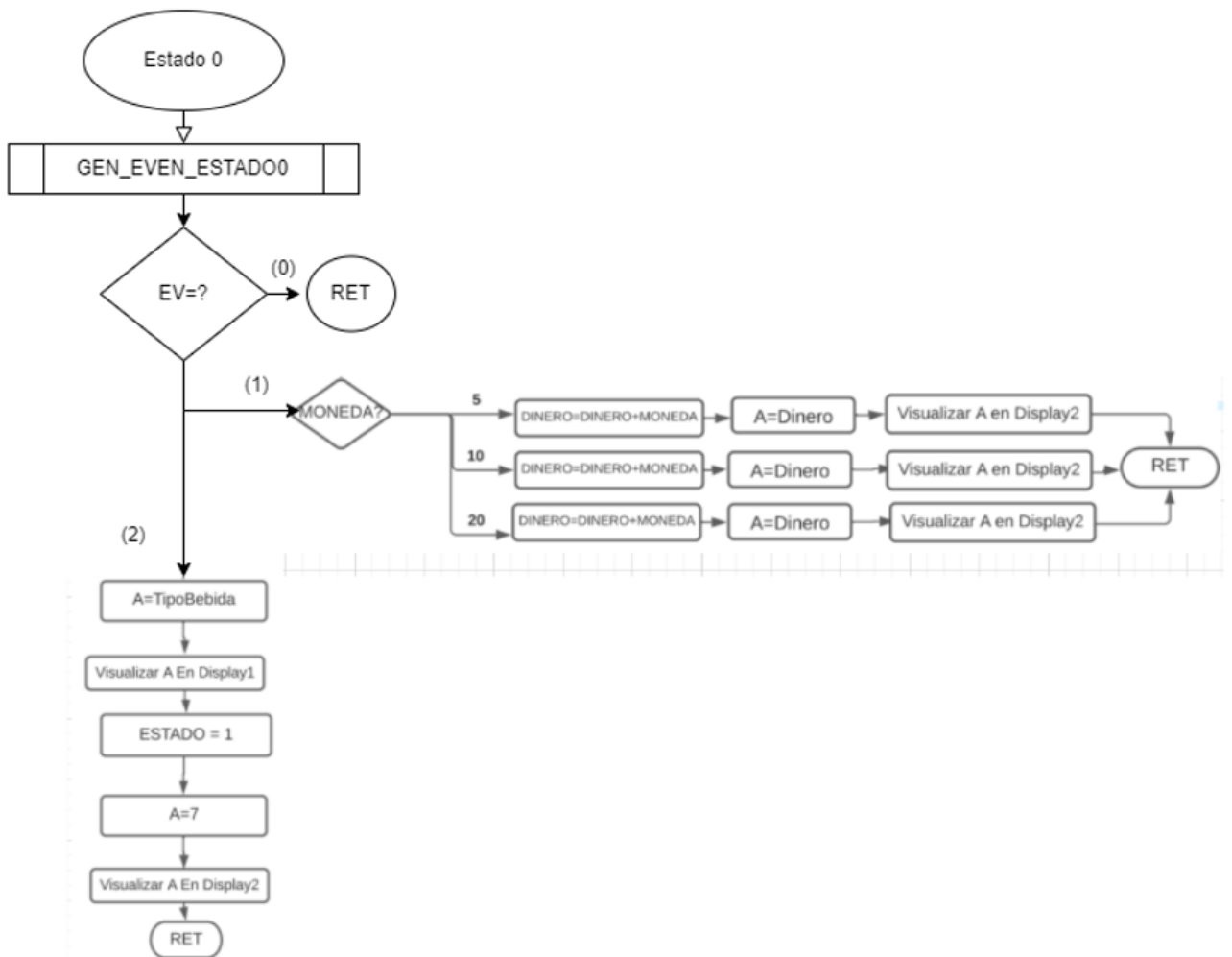
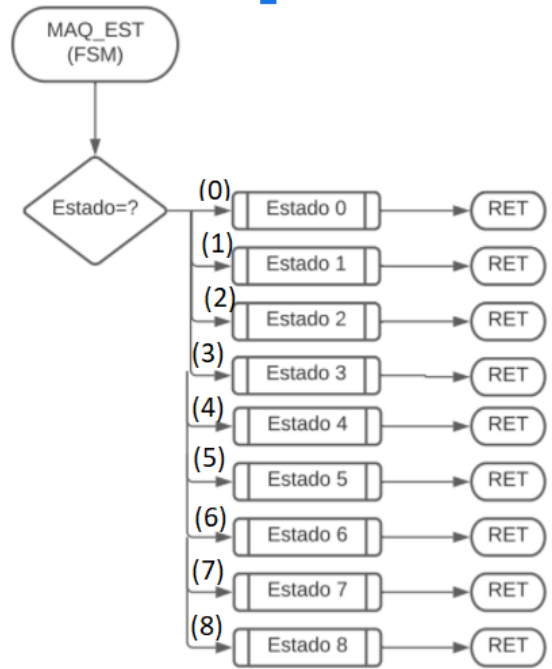
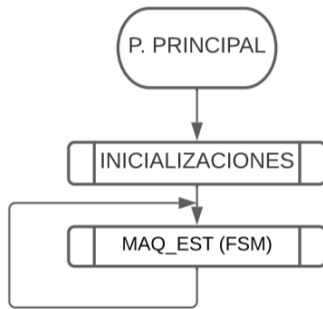
1.Introducción

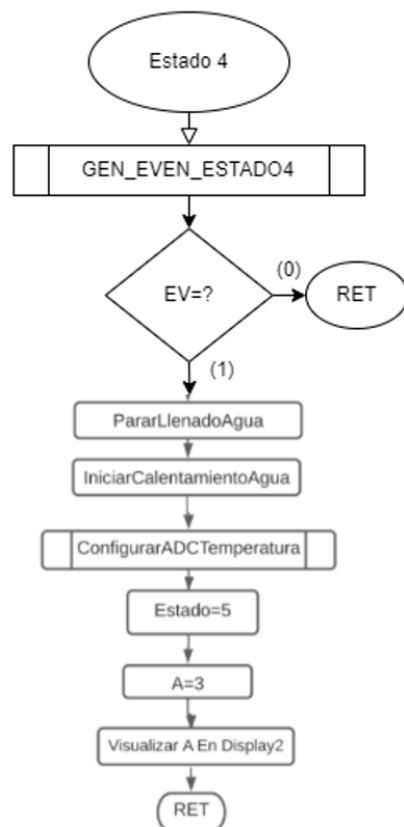
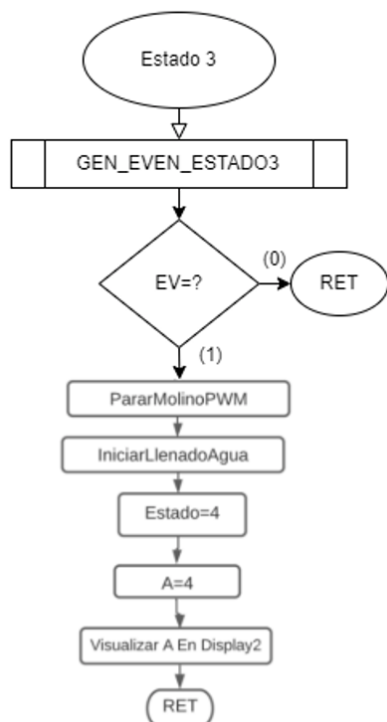
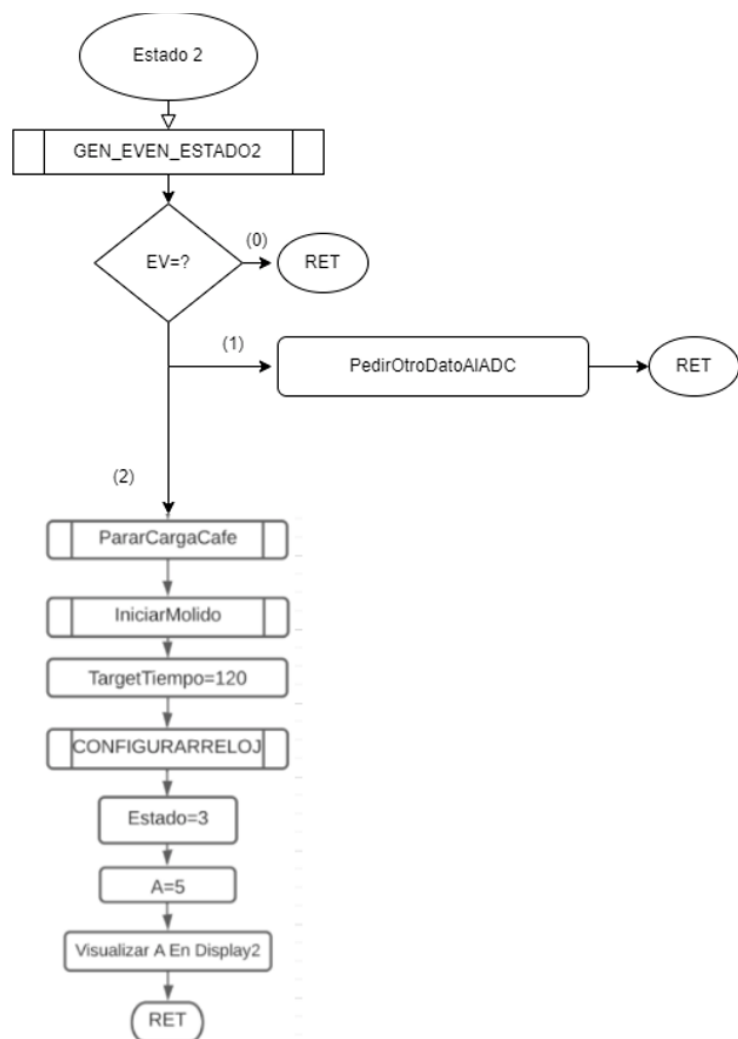
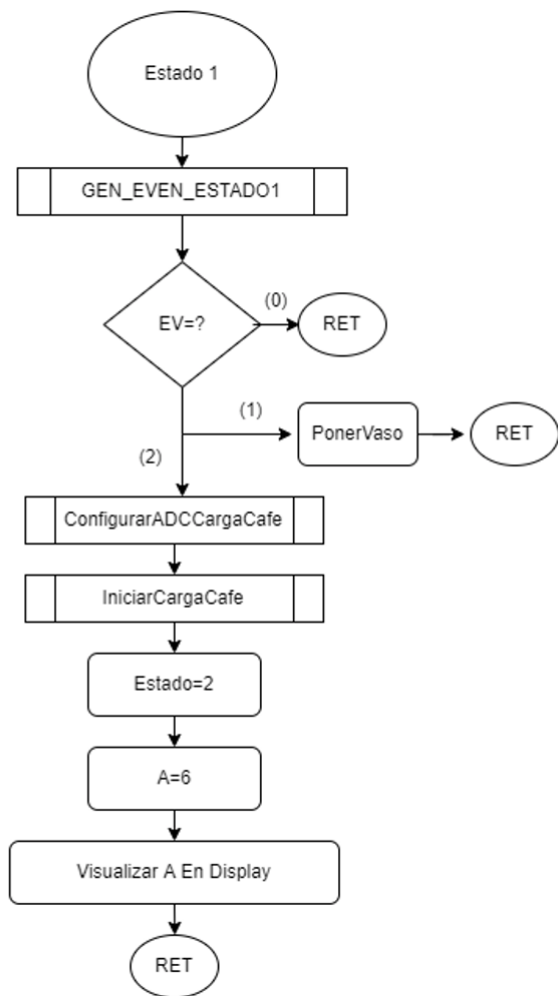
En este proyecto hemos diseñado el funcionamiento y control de una máquina de café mediante un diagrama de estados, eventos y acciones, y un diagrama de flujo. Además, tras la fase de diseño de dicha máquina, la hemos programado en código de ensamblador para el microcontrolador 80C552 de la marca Philips. El código está estructurado empleando subrutinas para mejorar la claridad del mismo. Para controlar la máquina de café se realizan lecturas y escrituras en puertos, conversiones a través del conversor analógico/digital (ADC), configuración de puertos PWM y configuración de timers para controlar el tiempo.

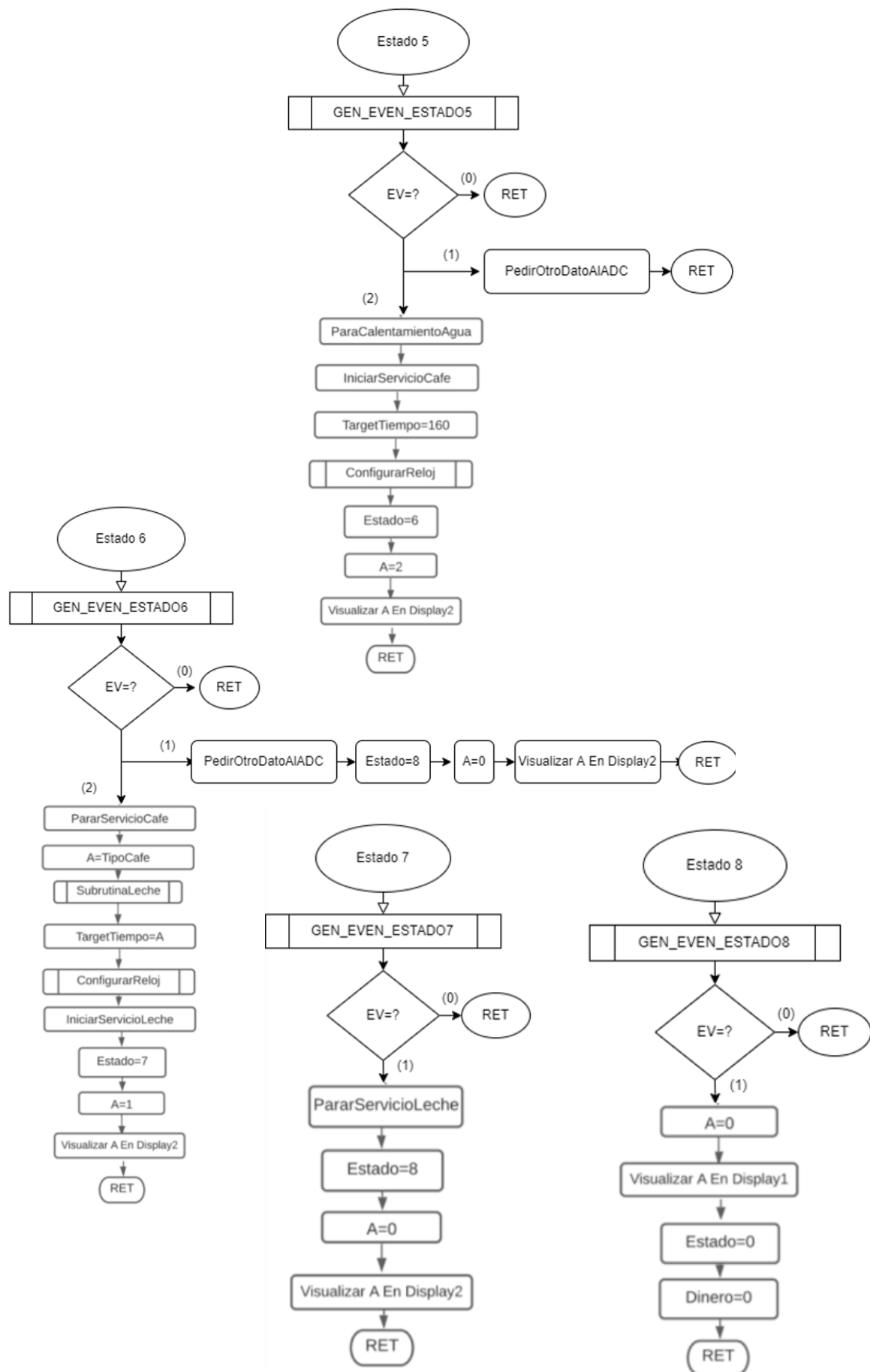
2.Diagramas de estados/eventos/acciones y de flujo

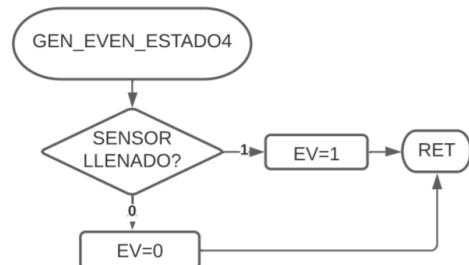
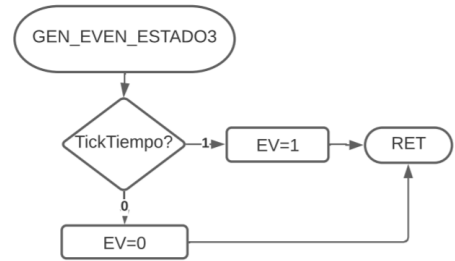
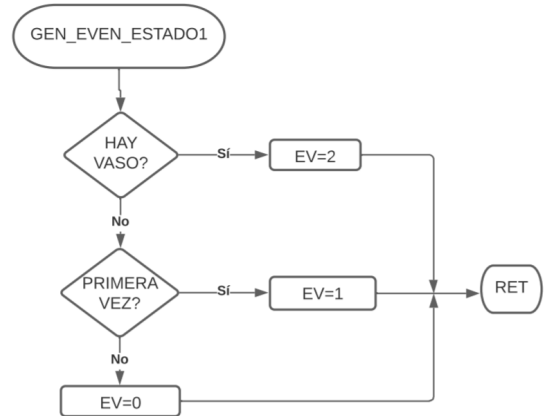
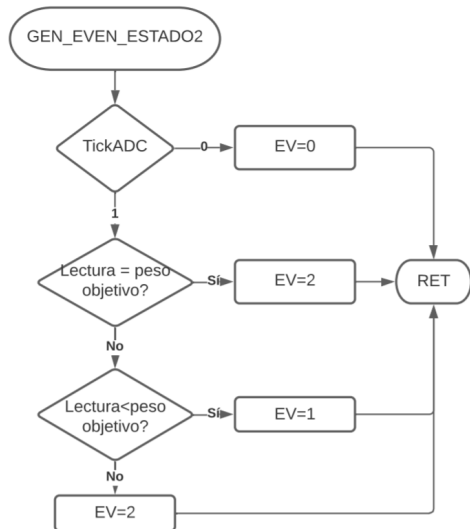
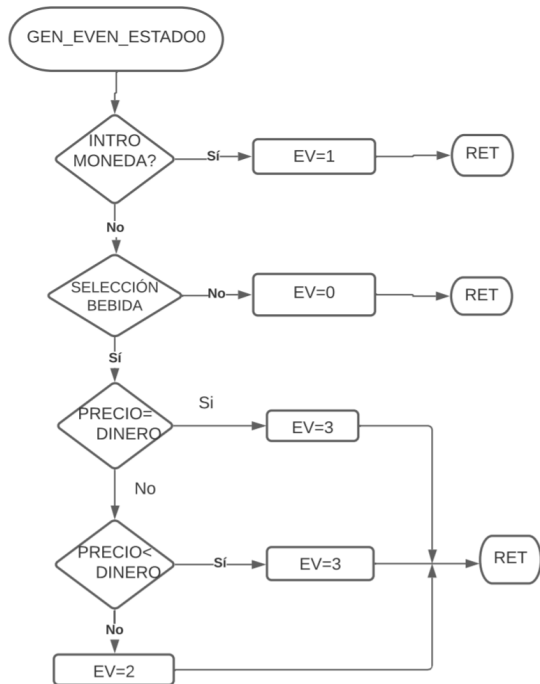
En este apartado se presentan 2 diagramas que representan el funcionamiento de la máquina de café. En primer lugar el diagrama de estados/eventos/acciones y seguidamente el diagrama de flujo. Los diagramas consisten de 9 estados con sus respectivos eventos cada uno. Cada evento tiene una acción asociada al mismo. Además en el diagrama de flujo se incluyen las subrutinas de atención a las interrupciones tanto del Timer como del ADC.

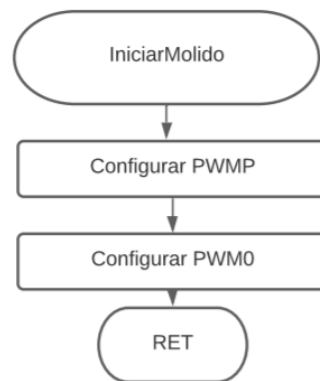
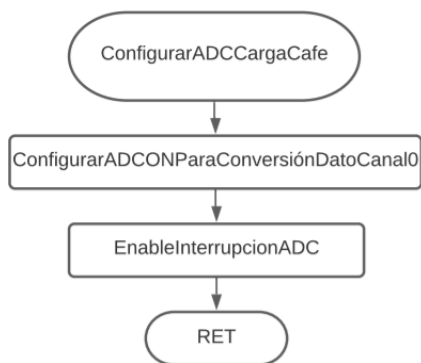
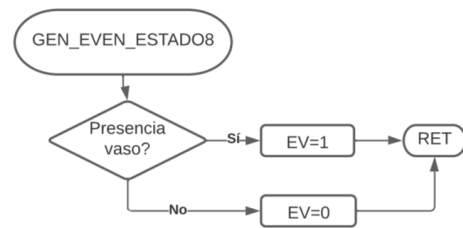
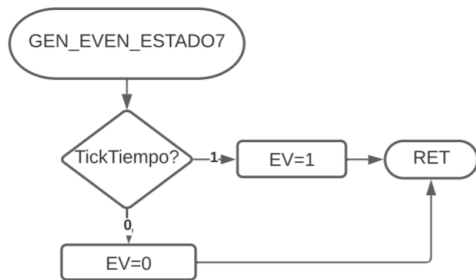
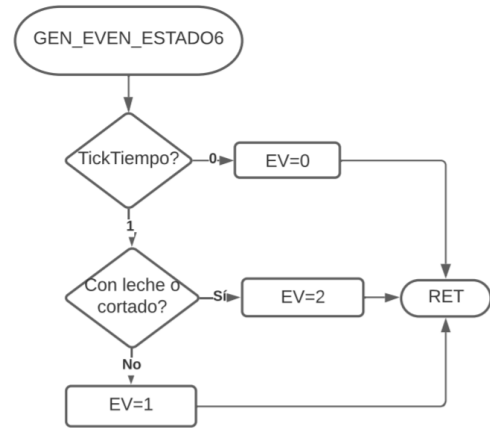
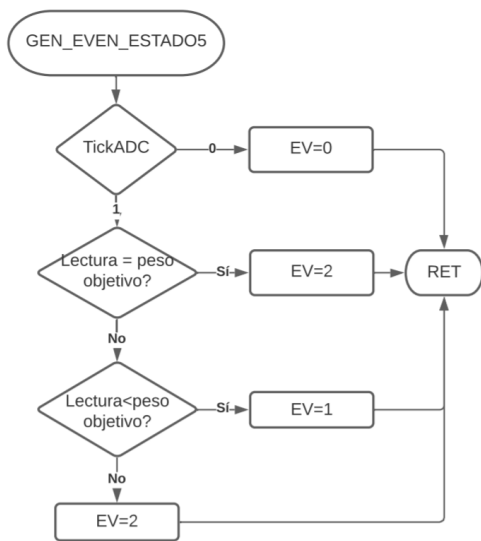


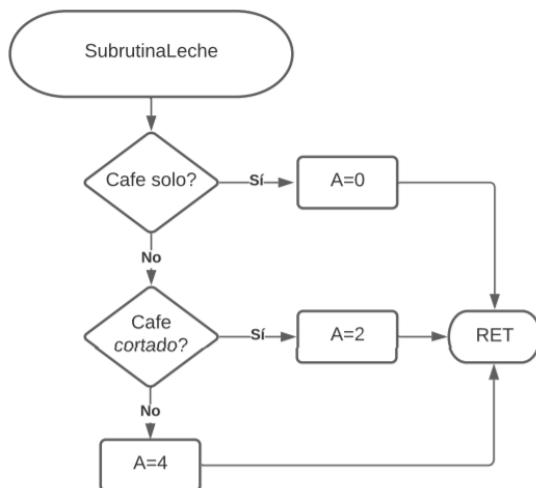
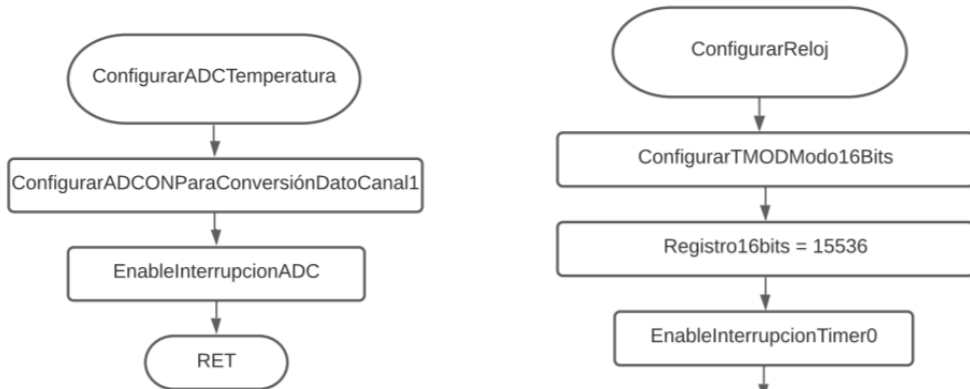
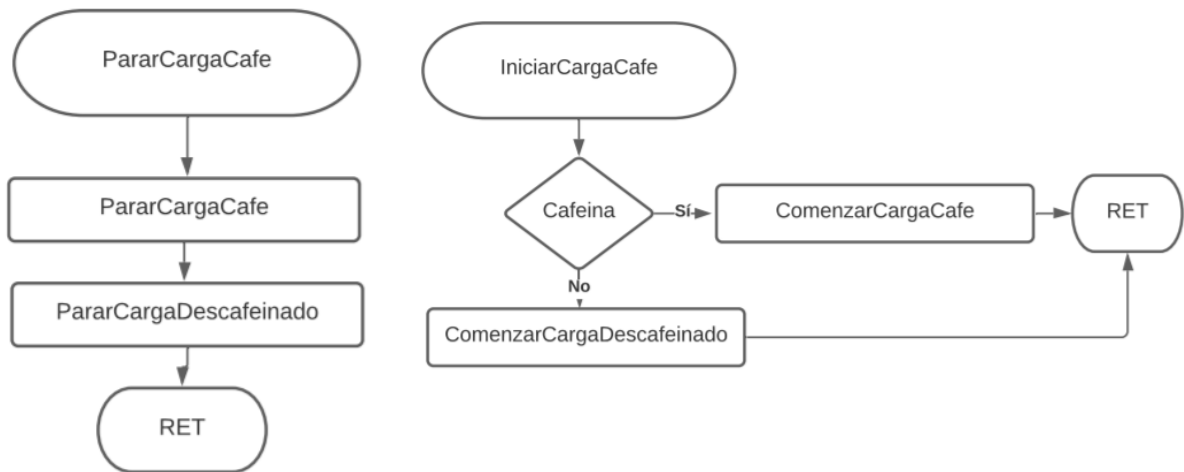


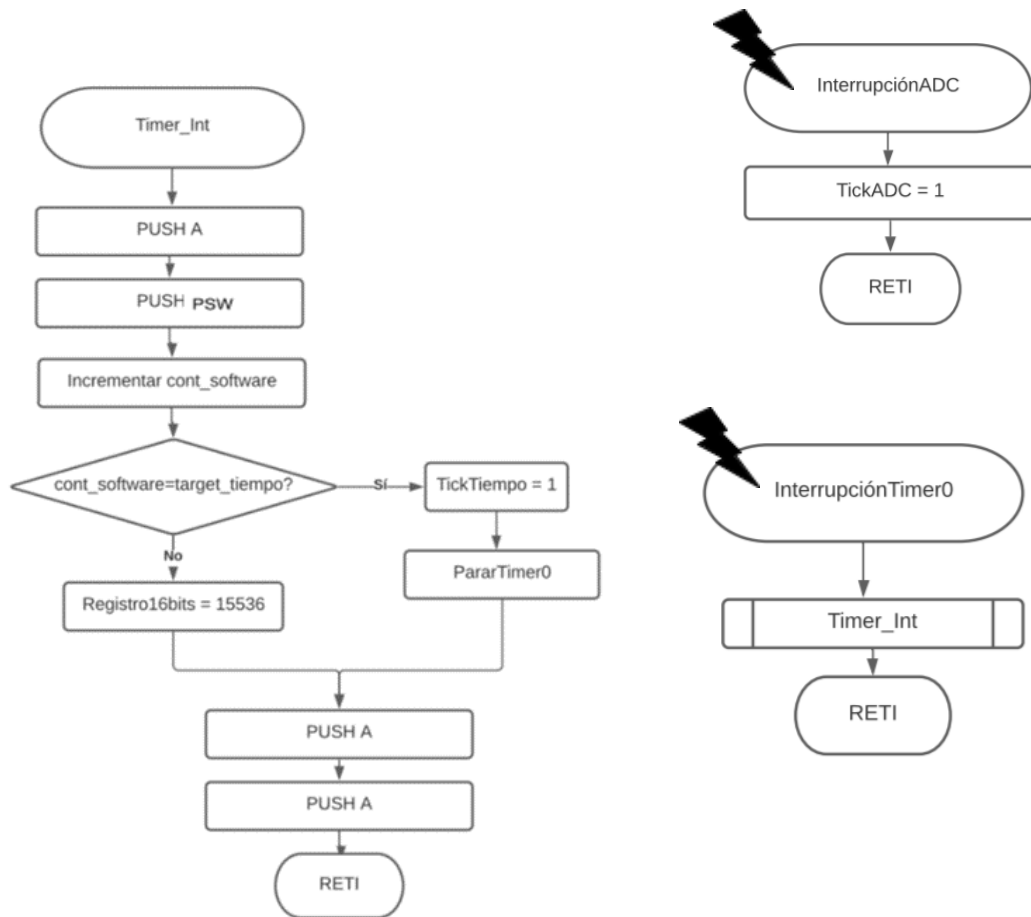












3.Cálculos y Comentarios

ADC

Para saber que valor tiene que tomar la conversión del ADC para conseguir el peso deseado debemos emplear la siguiente fórmula:

$$N_{v_{in}}^q = 256 \cdot \frac{V_{in} - V_{reg-}}{V_{reg+} - V_{reg-}}$$

Sabiendo que para el peso deseado de 99g la tensión de entrada es 3,86V, además sabiendo que $V_{reg+} = 5V$ y $V_{reg-} = 0V$:

$$N_{v_{in}}^q = 256 \cdot \frac{3,86V - 0V}{5V - 0V} = 197,63 \approx 198$$

Se realiza el redondeo del resultado hacia arriba para asegurar que la cantidad de café sea suficiente. Por lo tanto para que el peso sea 99g el resultado de la conversión ADC debe ser 198, este resultado de la conversión es equivalente a una tensión algo mayor que 3,86V debido al redondeo. El valor en hexadecimal es C6.

Sabiendo que para la temperatura deseada de 90°C la tensión de entrada es 4,5V, además sabiendo que $V_{reg+} = 5V$ y $V_{reg-} = 0V$:

$$N_{v_{in}}^q = 256 \cdot \frac{4,5V - 0V}{5V - 0V} = 230,4 \approx 231$$

Se realiza el redondeo del resultado hacia arriba para asegurar que la temperatura no es inferior a los 90°C. Por lo tanto para que la temperatura sea 90°C el resultado de la conversión ADC debe ser 231, este resultado de la conversión es equivalente a una tensión algo mayor que 4,5V debido al redondeo. El valor en hexadecimal es E7.

Timers

Para conseguir un tiempo T debemos aplicar la siguiente fórmula:

$$T = T_{osc} * PRESCALER * POSTSCALER * CONT_TIMER * CONT_SW$$

Donde T_{osc} es 1 entre la frecuencia del oscilador.

Sabiendo que la frecuencia del oscilador es 24MHz, el valor del prescaler es 12 y el postscaler es 1. Si queremos contar 1 segundo:

$$1s = \frac{1}{24MHz} * 12 * 1 * cont \rightarrow cont = 2 * 10^6$$

En el microcontrolador 80C552 disponemos de varias maneras en las que un timer puede operar, en nuestro caso el timer emplea un registro de 16 bits y, por lo tanto, el contador del timer contará 50000 ciclos de reloj y el contador de software contará 40 overflows del contador del timer para conseguir 1 segundo.

Para ello, pre-cargamos el contador de 16 bit con: $2^{16} - 50000 = 15536$

Para pre-cargar el contador con este valor se cargan los siguientes valores en hexadecimal: TH0=3C y TL0=B0.

PWM

En el proyecto se nos pide configurar el PWM a la mitad de la frecuencia máxima permitida por el PWMP, para ello debemos saber que con el oscilador interno del 80C552 de 24 MHz la frecuencia máxima permitida por el PWM es de 47,1 KHz. Teniendo en cuenta la siguiente formula podemos calcular el valor de PWMP:

$$f_{PWM} = \frac{f_{osc}}{2*(1+PWMP)*255} \rightarrow 23,53 \text{ kHz} = \frac{24 \text{ MHz}}{2*(1+PWMP)*255} \rightarrow PWMP = 1$$

Además, se nos pide que la velocidad de giro del molinillo sea un 30% del ciclo de trabajo (duty-cycle) del PWM

$$\text{Duty-cycle (D)} = \frac{t_{on}}{T} \quad D=0,3$$

$$\begin{aligned} \text{L/H ratio } PWM0 &= \frac{PWM0}{255-PWM0} \rightarrow \frac{t_{off}}{t_{on}} = \frac{PWM0}{255-PWM0} \rightarrow \frac{T}{t_{on}} = \frac{PWM0}{255-PWM0} + 1 \rightarrow \\ \frac{T}{t_{on}} &= \frac{PWM0}{255-PWM0} + 1 \rightarrow \frac{t_{on}}{T} = \frac{255-PWM0}{255} \rightarrow 0,3 = \frac{255-PWM0}{255} \\ PWM0 &= 178,5 \approx 179 \end{aligned}$$

4.Código

Evento EQU R6
Estado EQU R2
GranoCafe EQU P3.2
GranoDescafeinado EQU P3.7
LlenarAgua EQU P0.7
Calentar EQU P2.7
ServirCafe EQU P3.3
PonerVaso EQU P1.3
LlenoAgua EQU P3.0
Vaso EQU P3.1
ServirLeche EQU P3.6
PrimeraVezVaso EQU 0x20.0
TickADC EQU 0x20.1
TickTiempo EQU 0x20.2
TargetPeso EQU 0x21
TargetTemp EQU 0x23
TargetTiempo EQU 0x24
PWMP EQU 0xFE
PWM0 EQU 0xFC
CONTSW EQU 0x22
TipoCafe EQU 0x25
Dinero EQU 0x26
CuentaAtras EQU 0x27
Moneda EQU 0x28

ORG 0X00
AJMP PPRINCIPAL

ORG 0X0B
ACALL TimerInt

```

    RETI

ORG 0X0053
    ACALL ADCInt
    RETI

ORG 0X7B

PPRINCIPAL:
    ACALL INICIALIZACIONES
BUCLE:
    ACALL MAQUEST
    AJMP BUCLE

INICIALIZACIONES:
    MOV Estado,#0x00          ;Estado inicial 0
    MOV Dinero,#0x00          ;Dinero inicial 0
    MOV TargetPeso, #0xC6      ;Cantidad de cafe necesaria
    MOV TargetTemp, #0xE7      ;Temperatura del agua para hacer cafe
    ;Para hacer las pruebas se ponen los puertos a 0 en las inicializaciones
    RET

;*****MAQUINA DE ESTADOS*****
MAQUEST:
    MOV A, Estado              ;Se mueve al acumulador el estado al que hay que saltar
    RL A
    MOV DPTR,#JMP_TBL
    JMP @A + DPTR              ;Se salta al estado correspondiente teniendo en cuenta la tabla
JMP_TBL:
    AJMP ESPERA                ;Estado 0
    AJMP COLOCACIONVASO        ;Estado 1
    AJMP CARGACAFE             ;Estado 2
    AJMP MOLIDO                ;Estado 3
    AJMP CARGAAGUA              ;Estado 4
    AJMP CALENTAMIENTOAGUA     ;Estado 5
    AJMP SERVICIOCAFE           ;Estado 6
    AJMP SERVICIOLECHE          ;Estado 7
    AJMP FINPROCESO             ;Estado 8

;*****Estado 0: ESPERA*****
ESPERA:
    ACALL GENEVENESTADO0        ;Llamada al generador de eventos
    MOV A, Evento               ;Se mueve el numero de evento al acumulador
    RL A
    MOV DPTR,#JMP_TBL_EST0
    JMP @A + DPTR               ;Se salta al evento correspondiente
JMP_TBL_EST0:
    AJMP E0_EVENTO0
    AJMP E0_EVENTO1
    AJMP E0_EVENTO2

GENEVENESTADO0:                ;Generador de eventos
    MOV A, P1
    ANL A, #00000111B
    ANL P1, #11111000B
    JNZ SETESTADO0EVENTO1
    MOV A, P1
    ANL A, #11100000B
    ANL P1, #00011111B
    JNZ SETESTADO0EVENTO002
    MOV Evento, #0x00           ;No introducción, no selección --> Evento 0
    RET
SETESTADO0EVENTO1:             ;Introducción moneda --> Evento 1
    MOV Moneda, A
    MOV Evento, #0x01
    RET
SETESTADO0EVENTO002:           ;Se comprueba que bebida se ha seleccionado
    RL A
    RL A

```

RL A	
MOV TipoCafe, A	
ACALL SUBROUTINA_PRECIO	;Se pide el precio a una subrutina
CJNE A, Dinero, INSUFICIENTEOSOBRA	;Se comprueba si hay saldo suficiente
MOV Evento, #0x02	;Saldo suficiente --> Evento 2
RET	
INSUFICIENTEOSOBRA:	
JNC INSUFICIENTE	
MOV Evento, #0x02	;Saldo suficiente --> Evento 2
RET	
INSUFICIENTE:	;Saldo insuficiente --> Evento 0
MOV Evento, #0x00	
RET	
E0_EVENTO0:	;Evento 0
RET	
E0_EVENTO1:	;Evento 1: Intro moneda
MOV A, Moneda	
JB ACC.0, CINCOCENT	
JB ACC.1, DIEZCENT	
MOV A, Dinero	
ADD A, #0x14	;Se suma el valor de la moneda al dinero
MOV Dinero, A	
ACALL BCD_7SEG_DINERO_1	;Se visualiza el dinero en la pantalla
ACALL SUBROUTINA_BCD_7_SEGMENTOS	
ACALL VISUALIZAR_DISPLAY_2	
MOV A, Dinero	
ACALL BCD_7SEG_DINERO_2	
ACALL SUBROUTINA_BCD_7_SEGMENTOS	
ACALL VISUALIZAR_DISPLAY_1	
CLR P1.2	
RET	
CINCOCENT:	
MOV A, Dinero	
ADD A, #0x05	;Se suma el valor de la moneda al dinero
MOV Dinero, A	
ACALL BCD_7SEG_DINERO_1	;Se visualiza el dinero en la pantalla
ACALL SUBROUTINA_BCD_7_SEGMENTOS	
ACALL VISUALIZAR_DISPLAY_2	
MOV A, Dinero	
ACALL BCD_7SEG_DINERO_2	
ACALL SUBROUTINA_BCD_7_SEGMENTOS	
ACALL VISUALIZAR_DISPLAY_1	
RET	
DIEZCENT:	
MOV A, Dinero	
ADD A, #0x0A	;Se suma el valor de la moneda al dinero
MOV Dinero, A	
ACALL BCD_7SEG_DINERO_1	;Se visualiza el dinero en la pantalla
ACALL SUBROUTINA_BCD_7_SEGMENTOS	
ACALL VISUALIZAR_DISPLAY_2	
MOV A, Dinero	
ACALL BCD_7SEG_DINERO_2	
ACALL SUBROUTINA_BCD_7_SEGMENTOS	
ACALL VISUALIZAR_DISPLAY_1	
RET	
E0_EVENTO2:	;Evento 2: Selección bebida y dinero suficiente
MOV A, TipoCafe	
ACALL SUBROUTINA_BCD_7_SEGMENTOS	;Se visualiza el numero de la bebida en display 1
ACALL VISUALIZAR_DISPLAY_1	
MOV CuentaAtras, #0x07	;Se visualiza la "cuenta regresiva" del proceso en display 2
MOV A, CuentaAtras	
ACALL SUBROUTINA_BCD_7_SEGMENTOS	
ACALL VISUALIZAR_DISPLAY_2	
MOV Estado, #0x01	;Se pasa al siguiente estado
SETB PrimeraVezVaso	;Se activa este bit para el caso en el que en el siguiente estado
RET	;no se detecte vaso, saber que es la primera vez que esto pasa

```

;*****Estado 1: COLOCACION VASO*****
COLOCACIONVASO:
    ACALL GENEVENESTADO1                ;Llamada al generador de eventos
    MOV A, Evento                        ;Se mueve al acumulador el evento al que hay que saltar
    RL A
    MOV DPTR,#JMP_TBL_EST1              ;Se salta al evento correspondiente
    JMP @A + DPTR

JMP_TBL_EST1:
    AJMP E1_EVENTO0
    AJMP E1_EVENTO1
    AJMP E1_EVENTO2

GENEVENESTADO1:                          ;Generador de eventos
    JNB Vaso, SETESTADO1EVENTO1o0        ;Se comprueba la presencia de vaso
    MOV Evento, #0x02                    ;Vaso presente --> Evento 2
    RET

SETESTADO1EVENTO1o0:                      ;Vaso no presente
    JBC PrimeraVezVaso, SETESTADO1EVENTO1 ;Se comprueba si es la primera vez que no se detecta
    MOV Evento, #0x00                    ;No es la primera vez --> Evento 0
    RET

SETESTADO1EVENTO1:                        ;Es la primera vez -->Evento 1 (en el salto se hace CLR PrimeraVezVaso)
    MOV Evento, #0x01
    RET

E1_EVENTO0:                              ;Evento 0
    RET

E1_EVENTO1:                              ;Evento 1: Primera vez que no se detecta vaso
    SETB PonerVaso                       ;Mandar poner vaso
    RET

E1_EVENTO2:                              ;Evento 2: Vaso presente

    ACALL ConfigurarADCCargaCafe          ;Configurar el ADC para pesarel cafe
    ACALL IniciarCargaCafe                ;Iniciar la carga de cafe
    MOV Estado, #0x02                     ;Moverse al estado 2
    CLR PonerVaso                          ;Parar PonerVaso (para el caso en el que estuviera activo)
    DEC CuentaAtras
    MOV A, CuentaAtras                    ;Decrementar cuenta regresiva en display2
    ACALL SUBROUTINA_BCD_7_SEGMENTOS
    ACALL VISUALIZAR_DISPLAY_2
    RET

IniciarCargaCafe:
    MOV A, TipoCafe                       ;Mover el tipo de cafe al acumulador
    JB 0xE2,Descafeinado                  ;Comprobar si es con o sin cafeina (saltar en el ultimo caso)
    SETB GranoCafe                         ;Encender la carga de cafe (con cafeina)
    RET

Descafeinado:
    SETB GranoDescafeinado                ;Encender la carga de cafe (descafeinado)
    RET

ConfigurarADCCargaCafe:
    ANL 0XC5, #11111100B                  ;Canal 0 ADC
    SETB 0xA8.6                            ;ADC interrupt
    SETB 0xA8.7                            ;Global interrupt
    ORL 0XC5, #00001000B                  ;ADCS 1 --> Inicio conversion
    RET

;*****Estado 2: CARGA CAFE*****
CARGACAFE:
    ACALL GENEVENESTADO2                ;Llamada al generador de eventos
    MOV A, Evento                        ;Se mueve al acumulador el evento al que hay que saltar
    RL A
    MOV DPTR,#JMP_TBL_EST2              ;Se salta al evento correspondiente
    JMP @A + DPTR

```

```

JMP_TBL_EST2:
    AJMP E2_EVENTO0
    AJMP E2_EVENTO1
    AJMP E2_EVENTO2
    RET

GENEVENESTADO2:                                ;Generador de eventos
    JBC TickADC, SETESTADO2EVENTO1O2 ;Se comprueba si el TickADC esta activo (si ha terminado la conversión)
    MOV Evento, #0x00                        ;TickADC no activo --> Evento 0
    RET
SETESTADO2EVENTO1O2:                          ;Tick ADC activo
    MOV A, 0XC6                            ;Se mueve el contenido de la conversión al acumulador (ADCH) (solo los 8 bits MSB)

    CJNE A, TargetPeso, PESOMAYOROMENOR      ;Se comprueba si el peso es igual al objetivo, se salta a si no lo es
    MOV Evento, #0x02                        ;Peso igual al objetivo --> Evento 2
    RET
PESOMAYOROMENOR:                              ;Peso desigual
    JC SETESTADO2EVENTO1                    ;Carry activo --> dato de conversion < target --> salto
    MOV Evento, #0x02                        ;Carry no activo --> dato de conversion > target --> Evento 2
    RET
SETESTADO2EVENTO1:                            ;dato de conversion < target --> Evento 1
    MOV Evento, #0x01
    RET

E2_EVENTO0:                                    ;Evento 0
    RET

E2_EVENTO1:                                    ;Evento 1: dato de conversion < target
    ORL 0XC5, #00001000B                    ;Pedir otro dato
    RET

E2_EVENTO2:                                    ;Evento 2: dato de conversion >= target
    CLR 0xA8.6                              ;ADC interrupt
    ACALL Pararcargacafe                     ;Parar la carga de cafe
    ACALL Iniciarmolido                     ;Iniciar el molido de cafe
    MOV TargetTiempo, #0x78                 ;120d para contar 120 overflows (3s)
    ACALL Configurarreloj0                  ;Configurar el reloj
    MOV Estado, #0x03                        ;Morverse al estado 3
    DEC CuentaAtras
    MOV A, CuentaAtras                      ;Decrementar la cuenta regresiva y visualizar en
display 2
    ACALL SUBROUTINA_BCD_7_SEGMENTOS
    ACALL VISUALIZAR_DISPLAY_2
    RET

Pararcargacafe:
    CLR GranoCafe                           ;Parar carga cafe (con cafeina)
    CLR GranoDescafeinado                   ;Parar carga cafe (descafeinado)
    RET

Iniciarmolido:                                ;Configurar la salida PWM con los valores deseados
    MOV PWMP, #0X01                          ;PWMP = 0x01 para conseguir la mitar de la frecuencia maxima
    MOV PWM0, #0XB3                          ;PWMP0 = 0xB3 para conseguir un 30% del duty-cycle
    RET

;*****Estado 3: MOLIDO*****
MOLIDO:
    ACALL GENEVENESTADO3                    ;Llamada al generador de eventos
    MOV A, Evento                            ;Se mueve al acumulador el evento al que hay que saltar
    RL A
    MOV DPTR, #JMP_TBL_EST3
    JMP @A + DPTR                            ;Se salta al evento correspondiente
JMP_TBL_EST3:
    AJMP E3_EVENTO0
    AJMP E3_EVENTO1
    RET

GENEVENESTADO3:                                ;Generador de eventos
    JBC TickTiempo, SETESTADO3EVENTO1      ;Se comprueba si ha pasado el tiempo correspondiente, se salta
si es asi (haciendo clear)

```


MOV Evento, #0x00	;No ha pasado el tiempo --> Evento 0
RET	
SETESTADO3EVENTO1:	;Ha pasado el tiempo --> Evento 1
MOV Evento, #0x01	
RET	
E3_EVENTO0:	;Evento 0
RET	
E3_EVENTO1:	;Evento 1: Tiempo de molido cumplido
MOV PWM0, #0xFF	;Parar molino
CLR TCON.4	
SETB LlenarAgua	;Iniciar el llenado de agua
MOV Estado, #0x04	;Moverse al estado 4
DEC CuentaAtras	
MOV A, CuentaAtras	;Decrementar la cuenta regresiva y visualizar en el display 2
ACALL SUBROUTINA_BCD_7_SEGMENTOS	
ACALL VISUALIZAR_DISPLAY_2	
RET	
;*****Estado 4: CARGA AGUA*****	
CARGAAGUA:	
ACALL GENEVENESTADO4	;Llamada al generador de eventos
MOV A, Evento	;Se mueve al acumulador el evento al que hay que saltar
RL A	
MOV DPTR,#JMP_TBL_EST4	
JMP @A + DPTR	;Se salta al evento correspondiente
JMP_TBL_EST4:	
AJMP E4_EVENTO0	
AJMP E4_EVENTO1	
RET	
GENEVENESTADO4:	;Generador de eventos
JBC LlenoAgua,SETESTADO4EVENTO1	;Se comprueba el sensor de llenado y se salta si es asi (haciendo clear)
MOV Evento, #0x00	;Sensor de llenado no activo --> Evento 0
RET	
SETESTADO4EVENTO1:	;Sensor de llenado activo --> Evento 1
MOV Evento, #0x01	
RET	
E4_EVENTO0:	;Evento 0
RET	
E4_EVENTO1:	;Evento 1
CLR LlenarAgua	;Parar la carga de agua
SETB Calentar	;Iniciar el calentamiento del agua
ACALL ConfigurarADCTemperatura	;Configurar ADC
MOV Estado, #0x05	;Moverse al siguiente estado
DEC CuentaAtras	
MOV A, CuentaAtras	;Decrementar cuenta regresiva y visualizar en display 2
ACALL SUBROUTINA_BCD_7_SEGMENTOS	
ACALL VISUALIZAR_DISPLAY_2	
RET	
ConfigurarADCTemperatura:	
ORL 0xC5, #00000001B	;Canal 1 ADC
SETB 0xA8.6	;ADC interrupt
SETB 0xA8.7	;Global interrupt
ORL 0XC5, #00001000B	;ADCS 1 --> Inicio conversion
RET	
;*****Estado 5: CALENTAMIENTO AGUA*****	
CALENTAMIENTOAGUA:	
ACALL GENEVENESTADOS	;Llamada al generador de eventos
MOV A, Evento	;Se mueve al acumulador el evento al que hay que saltar
RL A	
MOV DPTR,#JMP_TBL_EST5	
JMP @A + DPTR	;Se salta al evento correspondiente
JMP_TBL_EST5:	
AJMP E5_EVENTO0	

```

    AJMP E5_EVENTO1
    AJMP E5_EVENTO2
    RET

GENEVENESTADO5:                                ;Generador de eventos
    JBC TickADC, SETESTADO5EVENTO1O2            ;Se comprueba el TickADC para ver si la
conversion ha terminado, se salta si es asi
    MOV Evento, #0x00                            ;El TickADC no esta encendido, la conversion sigue en curso --> Evento 0
    RET
SETESTADO5EVENTO1O2:                            ;TickADC a 1, la conversión ha terminado
    MOV A, 0XC6                                ;Se mueve el valor de la conversión (ADCH) al acumulador para compararlo
    CJNE A, TargetTemp, TEMPMAYOROMENOR        ;Se comprueba si la temperatura es igual al objetivo, se salta a
si no lo es
    MOV Evento, #0x02                            ;Temperatura igual al objetivo --> Evento 2
    RET
TEMPMAYOROMENOR:                                ;Temperatura diferente al objetivo
    JC SETESTADO5EVENTO1                        ;Carry activo --> dato de conversión < target --> salto
    MOV Evento, #0x02                            ;Carry no activo --> dato de conversión > target --> Evento 2
    RET
SETESTADO5EVENTO1:
    MOV Evento, #0x01                            ;dato de conversión < target --> Evento 1
    RET

E5_EVENTO0:                                    ;Evento 0
    RET

E5_EVENTO1:                                    ;Evento 1: conversión finalizada dato de conversión < objetivo
    ORL 0XC5, #00001000B                        ;Pedir otro dato
    RET

E5_EVENTO2:                                    ;Evento 2: conversión finalizada dato de conversión >= objetivo
    CLR Calentar                                ;Parar el calentamiento del agua
    CLR 0xA8.6                                  ;Parar interrupcion ADC
    SETB ServirCafe                            ;Iniciar servicio cafe
    MOV TargetTiempo, #0xA0                    ;160d para contar 160 overflows (4s)
    ACALL ConfigurarReloj0                      ;Configurar reloj
    MOV Estado, #0x06                          ;Moverse al siguiente estado
    DEC CuentaAtras                            ;Decrementar la cuenta regresiva y visualizarla en display 2
    MOV A, CuentaAtras
    ACALL SUBROUTINA_BCD_7_SEGMENTOS
    ACALL VISUALIZAR_DISPLAY_2
    RET

;*****Estado 6: SERVICIO DE CAFE*****
SERVICIOCAFE:
    ACALL GENEVENESTADO6                        ;Llamada al generador de eventos
    MOV A, Evento                                ;Se mueve al acumulador el evento al que hay que saltar
    RL A
    MOV DPTR, #JMP_TBL_EST6
    JMP @A + DPTR                                ;Se salta al evento correspondiente
JMP_TBL_EST6:
    AJMP E6_EVENTO0
    AJMP E6_EVENTO1
    AJMP E6_EVENTO2
    RET

GENEVENESTADO6:                                ;Generador de eventos
    JBC TickTiempo, SETESTADO6EVENTO1o2        ;Comprueba el TickTiempo, salta si es asi (haciendo clear)
    MOV Evento, #0x00                            ;TickTiempo no activo --> Estado 0
    RET
SETESTADO6EVENTO1o2:                            ;TickTiempo activo
    MOV A, TipoCafe
    ACALL SUBROUTINA_LECHE                      ;Se pide el tiempo de servicio de leche
    MOV TargetTiempo, A
    JNZ SETESTADO6EVENTO2                      ;Se comprueba si el tiempo de servicio de leche no es 0
    MOV Evento, #0x01                            ;Tiempo de servicio de leche igual a 0 --> Evento 1
    RET
SETESTADO6EVENTO2:                            ;Tiempo de servicio de leche diferente de 0 --> Evento 2
    MOV Evento, #0x02

```

```

RET

E6_EVENTO0:                                ;Evento 0
RET

E6_EVENTO1:                                ;Evento 1: Cafe solo
CLR ServirCafe                             ;Parar servicio cafe
CLR TCON.4
MOV Estado, #0x08                          ;Moverse al estado 8
DEC CuentaAtras
DEC CuentaAtras
MOV A, CuentaAtras                         ;Decrementar (por 2) cuenta regresiva y visualizar en display 2
ACALL SUBROUTINA_BCD_7_SEGMENTOS
ACALL VISUALIZAR_DISPLAY_2
RET

E6_EVENTO2:                                ;Evento 2: Cafe con leche o cortado
CLR ServirCafe
ACALL ConfigurarReloj0                     ;Se configura el reloj
SETB ServirLeche                           ;Se inicia el servicio de leche
MOV Estado, #0x07                          ;Moverse al estado 7
DEC CuentaAtras
MOV A, CuentaAtras                         ;Decrementar cuenta regresiva y visualizar en display 2
ACALL SUBROUTINA_BCD_7_SEGMENTOS
ACALL VISUALIZAR_DISPLAY_2
RET

;*****Estado 7: SERVICIO LECHE*****
SERVICIOLECHE:
ACALL GENEVENESTADO7                       ;Llamada al generador de eventos
MOV A, Evento                             ;Se mueve al acumulador el evento al que hay que saltar
RL A
MOV DPTR, #JMP_TBL_EST7
JMP @A + DPTR                             ;Se salta al evento correspondiente

JMP_TBL_EST7:
AJMP E7_EVENTO0
AJMP E7_EVENTO1
RET

GENEVENESTADO7:                             ;Genreador de eventos
JBC TickTiempo, SETESTADO7EVENTO1         ;Se comprueba el tickTiempo, y se salta si esta activo (haciendo clear)
MOV Evento, #0x00                          ;TickTiempo no activo --> Evento 0
RET

SETESTADO7EVENTO1:                         ;TickTiempo activo --> Evento 1
MOV Evento, #0x01
RET

E7_EVENTO0:                                ;Evento 0
RET

E7_EVENTO1:                                ;Evento 1: Tiempo servicio leche terminado
CLR ServirLeche                             ;Parar servicio leche
CLR TCON.4
MOV Estado, #0x08                          ;Moverse al siguiente estado
DEC CuentaAtras
MOV A, CuentaAtras                         ;Decrementar cuenta regresiva y visualizar en display 2
ACALL SUBROUTINA_BCD_7_SEGMENTOS
ACALL VISUALIZAR_DISPLAY_2
RET

;*****Estado 8: FIN PROCESO*****
FINPROCESO:
ACALL GENEVENESTADO8                       ;Llamada al generador de eventos
MOV A, Evento                             ;Se mueve al acumulador el evento al que hay que saltar
RL A
MOV DPTR, #JMP_TBL_EST8
JMP @A + DPTR                             ;Se salta al evento correspondiente

JMP_TBL_EST8:
AJMP E8_EVENTO0
AJMP E8_EVENTO1

```

```

RET

GENEVENESTADO8:                                ;Generador de eventos
JNB Vaso, SETESTADO8EVENTO1                    ;Se comprueba la presencia de vaso, se salta si no esta
MOV Evento, #0x00                              ;Vaso presente --> Evento 0
RET
SETESTADO8EVENTO1:                            ;Vaso no presente --> Evento 1
MOV Evento, #0x01
RET

E8_EVENTO0:                                    ;Evento 0
RET

E8_EVENTO1:                                    ;Evento 1: Vaso no presente
MOV A, #0x00                                    ;Visualizar 0 en display 1
ACALL SUBROUTINA_BCD_7_SEGMENTOS
ACALL VISUALIZAR_DISPLAY_1
MOV Estado, #0x00                              ;Moverse a estado 0
MOV Dinero, #0x00                              ;Resetear dinero
RET

;*****SUBROUTINAS*****
SUBROUTINA_BCD_7_SEGMENTOS:                    ;Devuelve el valor en binario que han de tomar los puertos del 7 seg
;para representar un numero
INC A
MOVC A,@A+PC
RET
DB      01111110b ;0
DB      00110000b ;1
DB      01101101b ;2
DB      01111001b ;3
DB      00110011b ;4
DB      01011011b ;5
DB      00011111b ;6
DB      01110000b ;7
DB      01111111b ;8
DB      01110011b ;9

BCD_7SEG_DINERO_1:                            ;El display 2 solo tiene dos opciones a la hora de mostrar el dinero (0 o 5)
JB Dinero.0, Cinco
MOV A, #0x00
RET
Cinco:
MOV A, #0x05
RET

BCD_7SEG_DINERO_2:                            ;El display 1 muestra las decenas del precio por lo tanto si dividimos el precio entre 10
obtenemos dicho numero
MOV B, #0x0A
DIV AB
RET

SUBROUTINA_LECHE:                            ;Devuelve el numero de interrupciones por overflow que hay que contar via sw para conseguir el
tiempo
INC A
MOVC A,@A+PC
RET
DB      0x00      ;NADA
DB      0x10      ;Cafe con leche
DB      0x08      ;Cafe cortado
DB      0x00      ;Cafe solo
DB      0x00      ;NADA
DB      0x10      ;Descafeinado con leche
DB      0x08      ;Descafeinado cortado
DB      0x00      ;Descafeinado solo

SUBROUTINA_CAFEINA:
INC A
MOVC A,@A+PC
RET

```

DB	0x00	;NADA
DB	0x01	;Cafe con leche
DB	0x01	;Cafe cortado
DB	0x01	;Cafe solo
DB	0x00	;NADA
DB	0x00	;Descafeinado con leche
DB	0x00	;Descafeinado cortado
DB	0x00	;Descafeinado solo

SUBROUTINA_PRECIO:

```

INC A
MOVC A,@A+PC
RET
DB 0x00 ;NADA
DB 0x28 ;Cafe con leche 40
DB 0x23 ;Cafe cortado 35
DB 0x1E ;Cafe solo 30
DB 0x00 ;NADA
DB 0x28 ;Descafeinado con leche 40
DB 0x23 ;Descafeinado cortado 35
DB 0x1E ;Descafeinado solo 30

```

VISUALIZAR_DISPLAY_1: ;Bit menos significativo al Carry y se mueve el carry al bit del puerto correspondiente

```

RRC A
MOV P0.6, C
RRC A
MOV P0.5, C
RRC A
MOV P0.4, C
RRC A
MOV P0.3, C
RRC A
MOV P0.2, C
RRC A
MOV P0.1, C
RRC A
MOV P0.0, C
RET

```

VISUALIZAR_DISPLAY_2:

```

RRC A
MOV P2.6, C
RRC A
MOV P2.5, C
RRC A
MOV P2.4, C
RRC A
MOV P2.3, C
RRC A
MOV P2.2, C
RRC A
MOV P2.1, C
RRC A
MOV P2.0, C
RET

```

TimerInt:

```

PUSH ACC
PUSH PSW
INC CONTSW
MOV A, CONTSW
SUBB A, TargetTiempo ;120d para contar 120 overflows y asi conseguir 3s
JZ cumpsegs
MOV TL0, #0XB0 ;Precargar el contador del timer para contar 50000 pulsos
MOV TH0, #0X3C
POP PSW
POP ACC
RET

```

cumpsegs:

```
SETB TickTiempo
POP PSW
POP ACC
RET
```

ADCInt:

```
PUSH ACC
PUSH PSW
ANL 0XC5, #11101111b
SETB TickADC
POP PSW
POP ACC
RET
```

Configurarreloj0:

```
MOV TMOD, #0X01 ;Timer 0 control interno 16bit
MOV TL0, #0XB0 ;Precargar el contador del timer para contar 50000 pulsos
MOV TH0, #0X3C
SETB 0xA8.1 ;Timer 0 interrupt
SETB 0xA8.7 ;Global interrupt
MOV CONTSW, #0X00 ;Resetear contador_sw
SETB TCON.4 ;Iniciar timer 0
RET
```

END