



MICROPROCESADORES 2003/2004

INVERNADERO

Jaime Pérez Crespo
Rubén Seijas Valverde

Introducción:

El objetivo de esta práctica es el diseño y programación de un sistema de control de invernadero basado en el microprocesador 8051. Para ello usamos un sistema formado por un sensor de temperatura (LM35) que nos proporciona un rango de medidas de temperatura entre -50 y 150 grados centígrados. Este valor es devuelto por el sensor como una diferencia de potencial que el conversor analógico-digital (ADC0804) se encarga de traducir a un valor digital definido por un rango entre 0 y 255, siendo el valor intermedio el que corresponde a 0 grados ($128 \text{ A/D} = 0$ grados).

El programa invernadero se encarga de leer la temperatura ambiente del sensor, y según unos umbrales definidos por el usuario, se encargará de activar una señal correspondiente a un sistema calefactor cuando la temperatura baje de ese umbral y la apagará cuando supere otra temperatura umbral también definida por el usuario. Así mismo también controla un sistema de apagado y encendido de luces. Para ello lleva el control de un reloj que muestra la hora actual, previamente introducida por el usuario, y que encenderá las luces cuando supere una hora límite y las apagará cuando alcance otra, ambas también introducidas por el usuario.

Para la implementar el programa se ha usado una rutina que provoca una interrupción del `Timer0` cada centésima de segundo. De esta forma tenemos el control del tiempo transcurrido, pudiendo reproducir así el reloj del sistema (la hora de comienzo se introduce al inicio de la ejecución). Para controlar el sistema de iluminación almacenamos la hora de encendido y apagado en registros del microprocesador, pidiendo al usuario hora y minutos.

A continuación se pide al usuario que introduzca los umbrales de encendido y apagado de la calefacción. Una vez almacenados todos los valores, comienza la ejecución del programa, que básicamente se mantiene en un bucle a la espera de las interrupciones del `Timer0`, y una vez haya transcurrido un segundo mostramos el estado actual del sistema por el puerto serie en el formato indicado por el enunciado de la práctica.

Cada vez que transcurre un segundo, realizamos una comprobación de la temperatura mediante el sensor. Debido a las posibles oscilaciones del mismo, en realidad la lectura se ejecuta en ocho ocasiones y posteriormente se calcula la media aritmética de todas las medidas. De este modo el sensor actualiza la temperatura cada segundo, mostrando el valor medio de todas las medidas realizadas. El estado de las calefacción y las luces se mostrará indicando ON/OFF en el puerto serie y se activarán o desactivarán los pines `P4.0` en el caso de la calefacción y el `P4.1` para las luces.

Listado:

A continuación se muestra el listado en ensamblador del programa construido para el control del invernadero propuesto.


```

56
---- 57          RSEG  STACK
0000 58          DS    30H          ; 16 bytes de pila
A51 MACRO ASSEMBLER INVERN
18:11:57 PAGE 2
25/10/03

59
---- 60          CSEG  AT      0
61          USING  0          ; Banco 0 de registros
62
0003 63          ORG   03h
0003 020000 F 64          LJMP  LEER_AD      ; Rutina de la interrupcion de
conversion lista
65
000B 66          ORG   0Bh
000B 020000 F 67          LJMP  DESBORD      ; Rutina de la interrupcion de
desbordamiento del timer
68
0000 69          ;Tras reset se comienza a ejecutar sobre la direccion 0
0000 70          ORG   0H
0000 020000 F 71          JMP   START      ; Saltamos al comienzo del programa
---- 72          RSEG  PROG
73
74
75          ;;;;;;;;;;;;;;
76          ;Provoca un retardo;
77          ;;;;;;;;;;;;;;
0000 78          RETARDO:
0000 F5F0 79          MOV   B,A
0002 C0F0 80          PUSH  B          ; Salvamos R0
0004 7400 81          MOV   A,#0D      ; Iniciamos el contador
0006 2401 82          BUCLE: ADD  A,#1D      ; Sumamos 1
0008 B40AFB 83          CJNE  A,#10D,BUCLE  ; Si el resultado es 0, salimos
000B D0F0 84          POP   B
000D E5F0 85          MOV   A,B          ; Restauramos A
000F 22 86          RET
87
88          ;;;;;;;;;;;;;;
89          ;Normaliza la temperatura leida;
90          ;;;;;;;;;;;;;;
128)+(N-128)*2/12)
0010 91          ; Normalizamos el dato leido del conversor a grados centigrados. (T=(N-
NORM_T:
0010 92          MOV   A,TEMP      ; Obtenemos el valor del conversor
0010 E51D 93          MOV   B,#128D      ; Preparamos la resta
0012 75F080 94          SUBB  A,B          ; Restamos
0015 95F0 95          MOV   R0,A          ; Guardamos la primera resta en R0
0017 F8 96          MOV   B,#2D          ; Introducimos 2 en B
0018 75F002 97          MUL  AB          ; Lo multiplicamos por 2
001B A4 98          MOV   B,#12D      ; Introducimos en B 12
001C 75F00C 99          DIV  AB          ; Lo dividimos por 12
001F 84 100          MOV   B,A          ; Movemos el resultado a B
0020 F5F0 101          MOV   A,R0          ; Movemos la primera resta a A
0022 E8 102          ADD  A,B          ; Hacemos la suma final
0023 25F0 103          MOV   TEMP,A      ; Guardamos el resultado en R7
0025 F51D 104          RET
0027 22 105
106
107          ;;;;;;;;;;;;;;
108          ;Rutina que lee la temperatura actual;
109          ;;;;;;;;;;;;;;
0028 110          LEER_TEMP:
111          ; Inicia una lectura de temperatura en el conversor, la salida se
indicara por interrupcion
0028 D2FE 112          SETB  CS_AD      ; Activamos las señales a 1 para
bloquear el conversor
002A D2FD 113          SETB  WR_AD
002C C2FE 114          CLR   CS_AD      ; Iniciamos la conversion
CS=0,RD=0,WR=1
002E C2FD 115          CLR   WR_AD
0030 120000 F 116          CALL  RETARDO      ; Esperamos a que se estabilicen las
señales
0033 D2FD 117          SETB  WR_AD
0035 D2FE 118          SETB  CS_AD
0037 22 119          RET
120
121          ;;;;;;;;;;;;;;
;;;
122          ;Rutina de tratamiento de la interrupcion del conversor analogico

```

```

digital;
123  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
124  ; El dato de la temperatura se lee de P6 y se guarda en TEMP
normalizado
A51 MACRO ASSEMBLER INVERN 25/10/03
18:11:57 PAGE 3

0038 125 LEER_AD:
0038 C0D0 126 PUSH PSW ; Metemos dentro de la pila la palabra
de estado
003A C083 127 PUSH DPH
003C C082 128 PUSH DPL
003E C2FE 129 CLR CS_AD ; Activamos la salida
0040 C2FF 130 CLR RD_AD ; CS=0,RD=0,WR=0
0042 120000 F 131 CALL RETARDO ; Esperamos
0045 E5FA 132 MOV A,TEMPD ; Leemos la temperatura del conversor
0047 F51D 133 MOV TEMP,A ; La almacenamos en su variable
0049 D2FF 134 SETB RD_AD
004B D2FE 135 SETB CS_AD
004D 120000 F 136 CALL NORM_T ; Normalizamos la temperatura
0050 D082 137 POP DPL
0052 D083 138 POP DPH
0054 D0D0 139 POP PSW ; Sacamos de la pila la palabra de
estado
0056 32 140 RETI
141
142 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
143 ;Rutina que realiza varias lecturas de la temperatura y calcula su
media;
144 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
0057 145 LEER_MEDIA:
146 ;Realiza 8 medidas y calcula su media para evitar oscilaciones del
sensor
0057 75F008 147 MOV B,#8D ;Cargamos el divisor en B
005A 7801 148 MOV R0,#1D ;Inicializamos el contador de la media
005C 120000 F 149 CALL LEER_TEMP ;Leemos la temperatura
005F 120000 F 150 CALL RETARDO ;Esperamos a que el dato este
disponible
0062 120000 F 151 CALL RETARDO
0065 8F01 152 MOV 01H,R7 ;Movemos la temperatura a R1
0067 153
CONT:
0067 E9 154 MOV A,R1 ;Guardamos la suma parcial en A
0068 120000 F 155 CALL LEER_TEMP ;Realizamos otra lectura
006B 120000 F 156 CALL RETARDO ;Esperamos
006E 120000 F 157 CALL RETARDO
0071 251D 158 ADD A,TEMP ;Sumamos la nueva lectura al total
0073 08 159 INC R0 ;Incrementamos el contador del numero
de medidas
0074 F9 160 MOV R1,A ;Almacenamos el resultado parcial en R1
0075 E8 161 MOV A,R0 ;Cargamos el contador en A
0076 B408EE 162 CJNE A,#8D,CONT ;Comprobamos si el numero de medidas
realizadas
0079 E9 163 MOV A,R1 ;Restauramos las sumas
007A 84 164 DIV AB ;Calculamos la media
007B E51D 165 MOV A,TEMP ;Movemos el resutado al registro de la
temperatura
007D 22 166 RET
167
168
169 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
170 ;Rutina que escribe un caracter en el puerto serie;
171 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
172 ;El caracter esta almacenado en A
007E 173 PUTCHAR:
007E C299 174 CLR TI ; TI = 0
0080 F599 175 MOV SBUF,A ; Enviamos el caracter
0082 3099FD 176 JNB TI,$ ; Si TI == 0 entonces linea ocupada,
esperamos a TI == 1
0085 22 177 RET
178
179 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
180 ;Rutina que escribe una cadena por el puerto serie;
181 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
182 ; Escribe la cadena apuntada en DPTR, y MODIFICA dicho puntero
0086 183 PUTSTRING:

```

```

0086 E4          184          CLR      A
0087 93          185          MOVC    A,@A+DPTR      ; Lee del segmento de codigo
0088 6006        186          JZ      EXIT          ; Sale si el caracter es 00H
008A 120000     F 187          CALL   PUTCHAR        ; Escribimos el caracter en el serie
008D A3          188          INC    DPTR           ; Incrementamos el contador
008E 80F6        189          SJMP   PUTSTRING
0090 22          190          EXIT:   RET
A51 MACRO ASSEMBLER INVERN
18:11:57 PAGE 4
25/10/03

191
192          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
193          ;Rutina que escribe una nueva linea por el puerto serie;
194          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0091          195          PUTCRLF:
0091 900000     F 196          MOV    DPTR,#MSG7
0094 120000     F 197          CALL   PUTSTRING
0097 22          198          RET
199
200          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
201          ;Rutina que escribe un caracter ':' por el puerto serie;
202          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0098          203          PUTSEPARATOR:
0098 743A        204          MOV    A,#58D          ; Cargamos un : en A
009A 120000     F 205          CALL   PUTCHAR        ; Lo escribimos
009D 22          206          RET
207
208          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
209          ;Rutina que escribe el estado del sistema por el puerto serie;
210          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
211          ; Escribe la hora actual guardada de R4 a R6
009E          212          PRINTSYS:
009E 900000     F 213          MOV    DPTR,#MSG14      ; "Son las hh:mm:ss"
00A1 120000     F 214          CALL   PUTSTRING
00A4 EC          215          MOV    A,R4            ; Cargamos las horas
00A5 120000     F 216          CALL   GETASCII       ; Las convertimos a digitos
00A8 120000     F 217          CALL   PUTCHAR        ; Escribimos el primer digito
00AB E5F0        218          MOV    A,B
00AD 120000     F 219          CALL   PUTCHAR        ; Escribimos el segundo digito
00B0 120000     F 220          CALL   PUTSEPARATOR   ; Escribimos un :
00B3 ED          221          MOV    A,R5            ; Cargamos los minutos
00B4 120000     F 222          CALL   GETASCII       ; Los convertimos a digitos
00B7 120000     F 223          CALL   PUTCHAR        ; Escribimos el primer digito
00BA E5F0        224          MOV    A,B
00BC 120000     F 225          CALL   PUTCHAR        ; Escribimos el segundo digito
00BF 120000     F 226          CALL   PUTSEPARATOR   ; Escribimos un :
00C2 EE          227          MOV    A,R6            ; Cargamos los segundos
00C3 120000     F 228          CALL   GETASCII       ; Los convertimos a digitos
00C6 120000     F 229          CALL   PUTCHAR        ; Escribimos el primer digito
00C9 E5F0        230          MOV    A,B
00CB 120000     F 231          CALL   PUTCHAR        ; Escribimos el segundo digito
232
00CE 900000     F 233          MOV    DPTR,#MSG15      ; ". La temperatura es "
00D1 120000     F 234          CALL   PUTSTRING
00D4 E51D        235          MOV    A,TEMP
00D6 120000     F 236          CALL   GETASCII       ; Los convertimos a digitos
00D9 120000     F 237          CALL   PUTCHAR        ; Escribimos el primer digito
00DC E5F0        238          MOV    A,B
00DE 120000     F 239          CALL   PUTCHAR        ; Escribimos el segundo digito
240
00E1 900000     F 241          MOV    DPTR,#MSG16      ; "°C. Calefaccion "
00E4 120000     F 242          CALL   PUTSTRING
00E7 20E808     F 243          JB    BCALEF,CALEF_OK
00EA 900000     F 244          MOV    DPTR,#MSG13
00ED 120000     F 245          CALL   PUTSTRING
00F0 8006        246          SJMP   LUCES
00F2          247          CALEF_OK:
00F2 900000     F 248          MOV    DPTR,#MSG12
00F5 120000     F 249          CALL   PUTSTRING
250
00F8          251          LUCES:
00F8 900000     F 252          MOV    DPTR,#MSG17      ; ". Luces "
00FB 120000     F 253          CALL   PUTSTRING
00FE 200008     F 254          JB    BLUCES,LUCES_OK
0101 900000     F 255          MOV    DPTR,#MSG13
0104 120000     F 256          CALL   PUTSTRING
A51 MACRO ASSEMBLER INVERN
18:11:57 PAGE 5
25/10/03

```

```

0107 8006          257          SJMP      PUNTO
0109              258          LUCES_OK:
0109 900000        F          259          MOV       DPTR,#MSG12
010C 120000        F          260          CALL      PUTSTRING
                                261
010F              262          PUNTO:
010F 742E          263          MOV       A,#'.'
0111 120000        F          264          CALL      PUTCHAR
                                265
0114 900000        F          266          MOV       DPTR,#MSG9          ; Iniciamos el cursor
0117 120000        F          267          CALL      PUTSTRING
011A 22           268          RET
                                269
                                270
                                271          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                272          ;Rutina que habilita la interrupcion del timer 0 y el evento externo 0;
                                273          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
011B              274          INTCONF:
011B D2A8          275          SETB     EX0          ; Habilitamos la interrupcion externa 0
011D D288          276          SETB     IT0
011F D2A9          277          SETB     ET0          ; Habilitamos la interrupcion de del
timer 0
0121 D2AF          278          SETB     EA
0123 22           279          RET
                                280
                                281
                                282          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                283          ;Rutina que lee un caracter por el puerto serie;
                                284          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0124              285          ; El caracter leido se almacena en A
0124 3098FD        286          GETCHAR:
recibido, esperamos a RI == 287          JNB      RI,$          ; Si RI == 0 entonces no se ha
                                288
                                289          1
0127 E599          288          MOV       A,SBUF          ; Leemos el caracter por el puerto
0129 C298          289          CLR       RI
                                290          ; Como mon51 envia 11H de vez en cuando lo ignoramos
012B F5F0          291          MOV       B,A          ; Salvaguardamos A
012D 24EF          292          ADD      A,#-11H          ; Restamos 11H a A
012F 60F3          293          JZ       GETCHAR          ; Si es cero leemos otro caracter
0131 E5F0          294          MOV       A,B          ; Recuperamos A desde B
0133 22           295          RET
                                296
                                297          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                298          ;Rutina que obtiene la equivalencia entre digitos ASCII y numeros;
                                299          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0134              300          ; #'0' == #48D, ... , #'9' == #57D
0134 F5F0          301          CONVERT:
0136 B43002        302          MOV       B,A          ; Guardamos A en B
Si no, saltamos a X          303          CJNE    A,#48D,UNO          ; Si es X, no saltamos, ponemos X en A.
                                304          +1
0139 7400          304          MOV       A,#00D
013B B43102        305          UNO:     CJNE    A,#49D,DOS
013E 7401          306          MOV       A,#01D
0140 B43202        307          DOS:     CJNE    A,#50D,TRES
0143 7402          308          MOV       A,#02D
0145 B43302        309          TRES:    CJNE    A,#51D,CUATRO
0148 7403          310          MOV       A,#03D
014A B43402        311          CUATRO: CJNE    A,#52D,CINCO
014D 7404          312          MOV       A,#04D
014F B43502        313          CINCO:   CJNE    A,#53D,SEIS
0152 7405          314          MOV       A,#05D
0154 B43602        315          SEIS:    CJNE    A,#54D,SIETE
0157 7406          316          MOV       A,#06D
0159 B43702        317          SIETE:   CJNE    A,#55D,OCHO
015C 7407          318          MOV       A,#07D
015E B43802        319          OCHO:    CJNE    A,#56D,NUEVE
0161 7408          320          MOV       A,#08D
A51 MACRO ASSEMBLER INVERN
18:11:57 PAGE      6
                                25/10/03

0163 B43902        321          NUEVE:   CJNE    A,#57D,FAIL          ; Si no es 9, comprobamos si fue alguno
de los anteriores
0166 7409          322          MOV       A,#09D
0168 B5F006        323          FAIL:    CJNE    A,B,OK          ; Si A es distinto de B, se encontro un
digito

```

```

016B 75F00A      324          MOV     B,#10D          ; Si no es un digito, B == 10D
016E 020000      325          JMP     OK2             ; Salimos directamente
0171 F5F0        326      OK:    MOV     B,A          ; Devolvemos el digito en A y B
0173 22          327      OK2:   RET
328
329
330      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
331      ;Rutina que pasa de numerico a ASCII;
332      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
333      ; Soporta un numero de dos digitos almacenado en A.
334      ; El primer digito se guarda en A, el segundo en B.
0174      GETASCII:
0174 F8          335          MOV     R0,A          ; Guardamos el numero en R0
0175 75F00A      336          MOV     B,#10D       ; Cargamos 10 en B
0178 84          337          DIV     AB           ; Dividimos el numero por 10
0179 F5F0        338          MOV     B,A          ; Guardamos en B una copia de las
decenas
017B 2430        339          ADD     A,#48D       ; Le sumamos #48 (ASCII '0') al primer
digito
017D F9          340          MOV     R1,A          ; Guardamos ese primer digito
017E E5F0        341          MOV     A,B          ; Restauramos las decenas
0180 75F00A      342          MOV     B,#10D       ; Cargamos un 10 en B
0183 A4          343          MUL     AB           ; Obtenemos las decenas
0184 F5F0        344          MOV     B,A          ; Guardamos las decenas en B
0186 E8          345          MOV     A,R0         ; Restauramos el numero original en A
0187 95F0        346          SUBB   A,B           ; Le restamos las decenas al original
0189 2430        347          ADD     A,#48D       ; Le sumamos #48 (ASCII '0') al segundo
digito
018B F5F0        348          MOV     B,A          ; Dejamos en B el segundo digito
018D E9          349          MOV     A,R1         ; y en A el primero
018E 22          350          RET
351
352      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
353      ;Rutina que configura el timer0 para que se desborde cada centesima;
354      ;de segundo
355      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
018F      TIMECONF:
018F 758901      357      ; Configuramos el Timer0:
0192 758CD8      358          MOV     TMOD,#00000001B ; C/T = 0, Mode = 1
desborde cada 359          MOV     TH0,#11011000B ; Configuramos el timer0 para que se
0195 758AF0      360          MOV     TL0,#11110000B ; centesima de segundo
0198 D28C        361          SETB   TR0           ; Arrancamos el timer 0
019A 7E00        362          MOV     R6,#0D      ; Segundos = 0
019C 7F00        363          MOV     R7,#0D      ; Centesimas = 0
019E 22          364          RET
365
366      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
367      ;Rutina de inicializacion del timer0;
368      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
019F      BLANK_TIMER:
019F 758CD8      370          MOV     TH0,#11011000B ; Configuramos el timer0 para que se
desborde cada 371
01A2 758AF0      371          MOV     TL0,#11110000B ; centesima de segundo
01A5 22          372          RET
373
374      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
375      ;Rutina que trata las interrupciones del timer;
376      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
01A6      DESBORD:
01A6 C0D0        378          PUSH   PSW           ; metemos dentro de la pila la palabra
de estado
01A8 C083        379          PUSH   DPH
01AA C082        380          PUSH   DPL
01AC F5F0        381          MOV     B,A
01AE C0F0        382          PUSH   B
01B0 C28D        383          CLR    TF0
01B2 120000      384          CALL  BLANK_TIMER   ; Reiniciamos el timer 0
01B5 120000      385          CALL  INCR_C        ; Incrementamos centesimas
01B8 D0F0        386          POP    B
A51 MACRO ASSEMBLER INVERN
18:11:57 PAGE 7
25/10/03

01BA E5F0        387          MOV     A,B
01BC D082        388          POP    DPL
01BE D083        389          POP    DPH
01C0 D0D0        390          POP    PSW          ; sacamos de la pila la palabra de
estado
01C2 32          391          RETI

```



```

392
393 ;Rutinas para manejo del reloj
394 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
395 ;Rutina para incremento de centesimas de segundo;
396 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
01C3 INCR_C:
01C3 C0F0 398 PUSH B ; Guardamos B
01C5 EF 399 MOV A,R7 ; Guardamos el valor para operar
01C6 2401 400 ADD A,#1D ; Incrementamos en 1 las centesimas
01C8 F5F0 401 MOV B,A ; Salvaguardamos el nuevo numero
01CA 249C 402 ADD A,#-100D ; Le restamos 100
01CC B40006 403 CJNE A,#0D,OK_C ; Si A == 0, reiniciamos cuenta
01CF 75F000 404 MOV B,#0D ; Reiniciamos las centesimas
01D2 120000 F 405 CALL INCR_S ; Incrementamos los segundos
01D5 AFF0 406 OK_C: MOV R7,B ; Restauramos las centesimas
01D7 D0F0 407 POP B ; Recuperamos B
01D9 22 408 RET
409
410 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
411 ;Rutina para incremento de segundos;
412 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
01DA INCR_S:
01DA C0F0 414 PUSH B ; Guardamos B
01DC D2E9 415 SETB FLAG ; Ponemos el flag a 1 para indicar que
han cambiado los seg
undos
01DE EE 416 MOV A,R6 ; Guardamos el valor para operar
01DF 2401 417 ADD A,#1D ; Incrementamos en 1 los segundos
01E1 F5F0 418 MOV B,A ; Salvaguardamos el nuevo numero
01E3 24C4 419 ADD A,#-60D ; Le restamos 60
01E5 B40006 420 CJNE A,#0D,OK_S ; Si A == 0, reiniciamos cuenta
01E8 75F000 421 MOV B,#0D ; Reiniciamos las centesimas
01EB 120000 F 422 CALL INCR_M ; Incrementamos los segundos
01EE AEF0 423 OK_S: MOV R6,B ; Restauramos los segundos
01F0 D0F0 424 POP B ; Recuperamos B
01F2 22 425 RET
426
427 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
428 ;Rutina para incremento de minutos;
429 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
01F3 INCR_M:
01F3 C0F0 431 PUSH B ; Guardamos B
01F5 ED 432 MOV A,R5 ; Guardamos el valor para operar
01F6 2401 433 ADD A,#1D ; Incrementamos en 1 los segundos
01F8 F5F0 434 MOV B,A ; Salvaguardamos el nuevo numero
01FA 24C4 435 ADD A,#-60D ; Le restamos 60
01FC B40006 436 CJNE A,#0D,OK_M ; Si A == 0, reiniciamos cuenta
01FF 75F000 437 MOV B,#0D ; Reiniciamos las centesimas
0202 120000 F 438 CALL INCR_H ; Incrementamos los segundos
0205 ADF0 439 OK_M: MOV R5,B ; Restauramos los segundos
0207 D0F0 440 POP B ; Recuperamos B
0209 22 441 RET
442
443 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
444 ;Rutina para incremento de horas;
445 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
020A INCR_H:
020A C0F0 447 PUSH B ; Guardamos B
020C EC 448 MOV A,R4 ; Guardamos el valor para operar
020D 2401 449 ADD A,#1D ; Incrementamos en 1 los segundos
020F F5F0 450 MOV B,A ; Salvaguardamos el nuevo numero
0211 24E8 451 ADD A,#-24D ; Le restamos 60
A51 MACRO ASSEMBLER INVERN 25/10/03
18:11:57 PAGE 8

0213 B40003 452 CJNE A,#0D,OK_H ; Si A == 0, reiniciamos cuenta
0216 75F000 453 MOV B,#0D ; Reiniciamos las centesimas
0219 ACF0 454 OK_H: MOV R4,B ; Restauramos los segundos
021B D0F0 455 POP B ; Recuperamos B
021D 22 456 RET
457
458 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
459 ;Rutina que comprueba si un numero sobrepasa cierto limite;
460 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
461 ; En B debe encontrarse el dato a comparar.
462 ; En A se encontrara el numero con el que comparar.
463 ; Si B > A entonces A = 1
464 ; Si B <= A entonces A = 0

```

```

021E          465      ISGT:
021E C0F0     466          PUSH   B           ; Guardamos B
0220 C3       467          CLR     C
0221 95F0     468          SUBB   A,B         ; Restamos B a A
0223 5006     469          JNC    CNS         ; Si no hay acarreo, es menor
0225 7401     470          MOV    A,#1D       ; Suponemos B mayor que A
0227 C3       471          CLR     C
0228 D0F0     472          POP    B           ; Restauramos B
022A 22       473          RET
022B          474      CNS:
022B 7400     475          MOV    A,#0D       ;
022D D0F0     476          POP    B           ; Restauramos B
022F 22       477          RET
478
479          ;
480          ;Rutina que lee un numero del puerto serie y lo valida;
481          ;
482          ; En B se encuentra el numero maximo a obtener
483          ; En A se devuelve 0 si no hubo exito, 1 en caso contrario
484          ; Si hay exito, en B se encuentra el numero
0230          485      GETNUM:
0230 C0F0     486          PUSH   B
0232 120000   F 487          CALL  GETCHAR     ; Leemos el primer numero
0235 FB       488          MOV    R3,A       ; Guardamos el caracter
0236 120000   F 489          CALL  CONVERT     ; Lo convertimos a numero
0239 A8F0     490          MOV    R0,B       ; Cargamos en R0 el resultado de la
conversion
023B B80A05   491          CJNE  R0,#10D,D1OK ; Si no es un digito, volvemos a
pedirlo
023E 7400     492          MOV    A,#0D       ; Error!!
0240 D0F0     493          POP    B
0242 22       494          RET
0243 75F00A   495      D1OK:  MOV    B,#10D     ; Cargamos el 10 en B
0246 A4       496          MUL   AB         ; Multiplicamos el primer digito por 10
0247 F9       497          MOV    R1,A       ; Cargamos el primer digito en R1
0248 EB       498          MOV    A,R3       ; Cargamos el digito para escribirlo
0249 120000   F 499          CALL  PUTCHAR     ; Escribimos el digito leido
500
024C 120000   F 501          CALL  GETCHAR     ; Leemos el segundo numero
024F FB       502          MOV    R3,A       ; Guardamos el caracter
0250 120000   F 503          CALL  CONVERT     ; Lo convertimos a numero
0253 FA       504          MOV    R2,A       ; Cargamos el segundo digito en R1
0254 BA0A05   505          CJNE  R2,#10D,D2OK ;
0257 7400     506          MOV    A,#0D       ; Error!!
0259 D0F0     507          POP    B
025B 22       508          RET
025C          509      D2OK:
025C EB       510          MOV    A,R3       ; Cargamos el digito para escribirlo
025D 120000   F 511          CALL  PUTCHAR     ; Escribimos el segundo digito de las
horas
0260 900000   F 512          MOV    DPTR,#MSG9  ; Saltamos de linea
0263 120000   F 513          CALL  PUTSTRING
514
0266 EA       515          MOV    A,R2       ; Restauramos el digito en A
0267 29       516          ADD   A,R1       ; Sumamos ambas cantidades, hemos
obtenido las horas
0268 F8       517          MOV    R0,A       ; Guardamos en R0 la suma final
A51 MACRO ASSEMBLER INVERN
18:11:57 PAGE 9
25/10/03

0269 D0F0     518          POP    B
026B E5F0     519          MOV    A,B
026D 88F0     520          MOV    B,R0
026F 120000   F 521          CALL  ISGT       ; Verificamos que sea correcta
0272 B40103   522          CJNE  A,#1D,SAVENUM
0275 7400     523          MOV    A,#0D       ; Error!!
0277 22       524          RET
0278          525      SAVENUM:
0278 88F0     526          MOV    B,R0
027A 7401     527          MOV    A,#1D       ; Exito!!
027C 22       528          RET
529
530          ;
531          ;Comprueba y cambia el estado de la calefaccion;
532          ;
533      CHECK_CALEF:
027D          534          MOV    B,TEMP     ; Cargamos la temperatura en B
027D 851DF0   534          MOV    A,TMAX     ; Cargamos la temperatura maxima en A
0280 E51F     535

```

```

0282 120000 F 536 CALL ISGT ; Verificamos que sea menor
0285 B40102 537 CJNE A,#1D,CALEF_OFF ; Comprobamos si la temperatura es
mayor que el maximo
0288 C2E8 538 CLR BCALEF ; Es mayor, apagamos
028A 539 CALEF_OFF:
028A E51E 540 MOV A,TMIN ; Cargamos la temperatura minima en A
028C 120000 F 541 CALL ISGT ; Verificamos que sea menor
028F B40002 542 CJNE A,#0D,CALEFRET ; Comprobamos si la temperatura es
menor que el minimo
0292 D2E8 543 SETB BCALEF ; Es menor, encendemos
0294 544 CALEFRET:
0294 22 545 RET
546
547 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
548 ;Comprueba y cambia el estado de la iluminacion;
549 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0295 550 CHECK_LUCES:
551 ; comprobamos que la hora actual sea MAYOR O IGUAL que la de encendido
0295 EC 552 MOV A,R4
0296 04 553 INC A ; Incrementamos la hora actual
0297 F5F0 554 MOV B,A ; Cargamos esa hora en B
0299 E519 555 MOV A,HON ; Cargamos la hora de encendido en A
029B 120000 F 556 CALL ISGT ; Comprobamos si la hora es mayor O
IGUAL
029E B40126 557 CJNE A,#1d,LUCESRET
02A1 ED 558 MOV A,R5
02A2 04 559 INC A ; Incrementamos los minutos actuales
02A3 F5F0 560 MOV B,A ; Cargamos los minutos en B
02A5 E51A 561 MOV A,MON ; Cargamos los minutos de encendido en
A
02A7 120000 F 562 CALL ISGT ; Comprobamos si los minutos son
mayores O IGUALES
02AA B4011A 563 CJNE A,#1d,LUCESRET
564 ; comprobamos que la hora actual sea MENOR O IGUAL que la de apagado
02AD 8CF0 565 MOV B,R4 ; Cargamos la hora en B
02AF E51B 566 MOV A,HOFF ; Cargamos la hora de apagado en A
02B1 120000 F 567 CALL ISGT ; Comprobamos si la hora es MENOR O
IGUAL
02B4 B40010 568 CJNE A,#0h,LUCESRET
02B7 ED 569 MOV A,R5
02B8 04 570 INC A ; Incrementamos los minutos actuales
02B9 F5F0 571 MOV B,A
572 ; MOV B,R5 ; Hora posterior, comprobamos minutos
02BB E51C 573 MOV A,MOFF ; Cargamos los minutos de apagado en A
02BD 120000 F 574 CALL ISGT ; Comprobamos si los minutos son
MENORES
02C0 B40004 575 CJNE A,#0h,LUCESRET ;
02C3 D2E9 576 SETB BLUCES ; La hora era posterior, ENCENDEMOS
LUCES
02C5 8002 577 SJMP RETURN
02C7 578 LUCESRET:
02C7 C2E9 579 CLR BLUCES ; La hora era posterior, APAGAMOS LUCES
02C9 580 RETURN:
02C9 22 581 RET
582
583 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

A51 MACRO ASSEMBLER INVERN 25/10/03
18:11:57 PAGE 10

584 ;Comienzo del programa;
585 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
02CA 586 START:
02CA 758100 F 587 MOV SP,#STACK-1 ; Inicializamos el registro SP
588
589 ; Inicializamos el puerto serie
02CD 759850 590 MOV SCON,#01010000B
02D0 758920 591 MOV TMOD,#00100000B ; C/T = 0, Mode = 2
02D3 758DF3 592 MOV TH1,#0F3H
02D6 D28E 593 SETB TR1
02D8 D299 594 SETB TI
595
02DA 900000 F 596 MOV DPTR,#MSG0 ; Cargamos el mensaje en DPTR
02DD 120000 F 597 CALL PUTSTRING ; Escribimos la cadena de texto
02E0 900000 F 598 MOV DPTR,#MSG7 ; Escribimos un CRLF
02E3 120000 F 599 CALL PUTSTRING
600
02E6 7E00 601 MOV R6,#0D
02E8 7F00 602 MOV R7,#0D

```

```

603
604 ; Pedimos al usuario que ponga el reloj en hora
605 ; HORAS
02EA 02EA 900000 F 606 BACKH:
02EA 900000 F 607 MOV DPTR,#MSG5 ; Cargamos el mensaje en DPTR
02ED 120000 F 608 CALL PUTSTRING
02F0 75F017 609 MOV B,#23D ; Numero maximo de horas
02F3 120000 F 610 CALL GETNUM ; Leemos y validamos las horas
02F6 B401F1 611 CJNE A,#1D,BACKH
02F9 ACF0 612 MOV R4,B
02FB 120000 F 613 CALL PUTCRLF
614
615 ; Si la hora es correcta, pasamos a leer los minutos
616 ; MINUTOS
02FE 02FE 900000 F 617 BACKM:
02FE 900000 F 618 MOV DPTR,#MSG6 ; Cargamos el mensaje en DPTR
0301 120000 F 619 CALL PUTSTRING
0304 75F03B 620 MOV B,#59D ; Numero maximo de minutos
0307 120000 F 621 CALL GETNUM ; Leemos y validamos los minutos
030A B401F1 622 CJNE A,#1D,BACKM
030D ADF0 623 MOV R5,B
030F 120000 F 624 CALL PUTCRLF
625
626 ; Si los minutos son correctos, leemos Hon
0312 0312 900000 F 627 BACKHON:
0312 900000 F 628 MOV DPTR,#MSG4
0315 120000 F 629 CALL PUTSTRING
0318 75F017 630 MOV B,#23D ; Hora encendido luz maxima
031B 120000 F 631 CALL GETNUM ; Leemos y validamos Hon
031E B401F1 632 CJNE A,#1D,BACKHON
0321 E5F0 633 MOV A,B
0323 F519 634 MOV HON,A
0325 120000 F 635 CALL PUTCRLF
636
637 ; Si Hon es correcto, leemos Mon
0328 0328 900000 F 638 BACKMON:
0328 900000 F 639 MOV DPTR,#MSG3
032B 120000 F 640 CALL PUTSTRING
032E 75F03B 641 MOV B,#59D ; Minutos encendido luz maximos
0331 120000 F 642 CALL GETNUM ; Leemos y validamos Mon
0334 B401F1 643 CJNE A,#1D,BACKMON
0337 E5F0 644 MOV A,B
0339 F51A 645 MOV MON,A
033B 120000 F 646 CALL PUTCRLF
647
648 ; Si Mon es correcta, leemos Hoff
033E 033E 900000 F 649 BACKHOFF:
A51 MACRO ASSEMBLER INVERN
18:11:57 PAGE 11
033E 900000 F 650 MOV DPTR,#MSG2
0341 120000 F 651 CALL PUTSTRING
0344 75F017 652 MOV B,#23D ; Hora apagado luz maxima
0347 120000 F 653 CALL GETNUM ; Leemos y validamos Hoff
034A B401F1 654 CJNE A,#1D,BACKHOFF
034D E5F0 655 MOV A,B
034F F51B 656 MOV HOFF,A
0351 120000 F 657 CALL PUTCRLF
658
659 ; Si Hoff es correcta, leemos Moff
0354 0354 900000 F 660 BACKMOFF:
0354 900000 F 661 MOV DPTR,#MSG1
0357 120000 F 662 CALL PUTSTRING
035A 75F03B 663 MOV B,#59D ; Minutos apagado luz maximos
035D 120000 F 664 CALL GETNUM ; Leemos y validamos Moff
0360 B401F1 665 CJNE A,#1D,BACKMOFF
0363 E5F0 666 MOV A,B
0365 F51C 667 MOV MOFF,A
0367 120000 F 668 CALL PUTCRLF
669
670 ; Si Hoff es correcta, leemos Tmin
036A 036A 900000 F 671 BACKTMIN:
036A 900000 F 672 MOV DPTR,#MSG10
036D 120000 F 673 CALL PUTSTRING
0370 75F00A 674 MOV B,#10D ; Maxima temperatura para encendido
calefaccion
0373 120000 F 675 CALL GETNUM ; Leemos y validamos Tmin
0376 B401F1 676 CJNE A,#1D,BACKTMIN

```

```

0379 E5F0          677          MOV      A,B
037B F51E          678          MOV      TMIN,A
037D 120000      F      679          CALL     PUTCRLF
680
681          ; Si Tmin es correcta, leemos Tmax
0380          682          BACKTMAX:
0380 900000      F      683          MOV      DPTR,#MSG11
0383 120000      F      684          CALL     PUTSTRING
0386 75F01E      685          MOV      B,#30D          ; Maxima temperatura para apagado
calefaccion
0389 120000      F      686          CALL     GETNUM          ; Leemos y validamos Tmax
038C B401F1      687          CJNE    A,#1D,BACKTMAX
038F E5F0          688          MOV      A,B
0391 F51F          689          MOV      TMAX,A
0393 120000      F      690          CALL     PUTCRLF
691
0396          692          CONFINIT:
0396 120000      F      693          CALL     INTCONF          ; Habilitamos interrupciones
0399 120000      F      694          CALL     TIMECONF          ; Configuramos el timer
039C D220          695          SETB    FLAG          ; Marcamos los segundos como cambiados
039E 120000      F      696          CALL     LEER_MEDIA          ; Leemos la temperatura inicialmente
03A1          697          MAIN:
03A1 102002      698          JBC     FLAG,NEWSEG
03A4 80FB          699          SJMP    MAIN          ; Sino ha cambiado el segundo no lo
mostramos
03A6          700          NEWSEG:
03A6 120000      F      701          CALL     CHECK_CALEF          ; Comprobamos la calefaccion
03A9 120000      F      702          CALL     CHECK_LUCES          ; Comprobamos las luces
03AC 120000      F      703          CALL     PRINTSYS          ; Imprimimos la informacion de estado
03AF C220          704          CLR     FLAG
03B1 120000      F      705          CALL     LEER_MEDIA          ; Leemos la temperatura
03B4 80EB          706          SJMP    MAIN          ; Bucle infinito
707
708          ; Cadenas de caracteres
03B6 53495354    709          MSG0:   DB      'SISTEMA INVERNADERO',CR,LF,00H
03BA 454D4120
03BE 494E5645
03C2 524E4144
03C6 45524F0D
03CA 0A00
03CC 494C554D    710          MSG1:   DB      'ILUMINACION OFF (MM): ',00H
A51 MACRO ASSEMBLER INVERN
18:11:57 PAGE 12
25/10/03

03D0 494E4143
03D4 494F4E20
03D8 4F464620
03DC 284D4D29
03E0 3A2000
03E3 494C554D    711          MSG2:   DB      'ILUMINACION OFF (HH): ',00H
03E7 494E4143
03EB 494F4E20
03EF 4F464620
03F3 28484829
03F7 3A2000
03FA 494C554D    712          MSG3:   DB      'ILUMINACION ON (MM): ',00H
03FE 494E4143
0402 494F4E20
0406 4F4E2028
040A 4D4D293A
040E 2000
0410 494C554D    713          MSG4:   DB      'ILUMINACION ON (HH): ',00H
0414 494E4143
0418 494F4E20
041C 4F4E2028
0420 4848293A
0424 2000
0426 52454C4F    714          MSG5:   DB      'RELOJ (HH): ',00H
042A 4A202848
042E 48293A20
0432 00
0433 52454C4F    715          MSG6:   DB      'RELOJ (MM): ',00H
0437 4A20284D
043B 4D293A20
043F 00
0440 0D0A00      716          MSG7:   DB      CR,LF,00H
0443 484F5241    717          MSG8:   DB      'HORA: ',00H
0447 3A2000

```

```

044A 0D00          718    MSG9:   DB      CR,00H
044C 54454D50     719    MSG10:  DB      'TEMPERATURA MINIMA (TT): ',00H
0450 45524154
0454 55524120
0458 4D494E49
045C 4D412028
0460 5454293A
0464 2000
0466 54454D50     720    MSG11:  DB      'TEMPERATURA MAXIMA (TT): ',00H
046A 45524154
046E 55524120
0472 4D415849
0476 4D412028
047A 5454293A
047E 2000
0480 4F4E00       721    MSG12:  DB      'ON',00H
0483 4F464600     722    MSG13:  DB      'OFF',00H
0487 536F6E20     723    MSG14:  DB      'Son las ',00H
048B 6C617320
048F 00
0490 2E204C61     724    MSG15:  DB      '. La temperatura es ',00H
0494 2074656D
0498 70657261
049C 74757261
04A0 20657320
04A4 00
04A5 BA432E20     725    MSG16:  DB      '°C. Calefaccion ',00H
04A9 43616C65
04AD 66616363
04B1 696F6E20
04B5 00
04B6 2E204C75     726    MSG17:  DB      '. Luces ',00H
A51 MACRO ASSEMBLER INVERN
18:11:57 PAGE      13

```

25/10/03

```

04BA 63657320
04BE 00
727
04BF          728    FIN:
729          END
A51 MACRO ASSEMBLER INVERN
18:11:57 PAGE      14

```

25/10/03

SYMBOL TABLE LISTING

N A M E	T Y P E	V A L U E	A T T R I B U T E S	
B.	D ADDR	00F0H	A	
BACKH.	C ADDR	02EAH	R	SEG=PROG
BACKHOFF.	C ADDR	033EH	R	SEG=PROG
BACKHON.	C ADDR	0312H	R	SEG=PROG
BACKM.	C ADDR	02FEH	R	SEG=PROG
BACKMOFF.	C ADDR	0354H	R	SEG=PROG
BACKMON.	C ADDR	0328H	R	SEG=PROG
BACKTMAX.	C ADDR	0380H	R	SEG=PROG
BACKTMIN.	C ADDR	036AH	R	SEG=PROG
BCALEF.	B ADDR	00E8H.0	A	
BLANK_TIMER.	C ADDR	019FH	R	SEG=PROG
BLUCES.	B ADDR	00E8H.1	A	
BUCLE.	C ADDR	0006H	R	SEG=PROG
CALEFRET.	C ADDR	0294H	R	SEG=PROG
CALEF_OFF.	C ADDR	028AH	R	SEG=PROG
CALEF_OK.	C ADDR	00F2H	R	SEG=PROG
CHECK_CALEF.	C ADDR	027DH	R	SEG=PROG
CHECK_LUCES.	C ADDR	0295H	R	SEG=PROG
CINCO.	C ADDR	014FH	R	SEG=PROG
CNS.	C ADDR	022BH	R	SEG=PROG
CONFINIT.	C ADDR	0396H	R	SEG=PROG
CONT.	C ADDR	0067H	R	SEG=PROG
CONVERT.	C ADDR	0134H	R	SEG=PROG
CR.	N NUMB	000DH	A	
CS_AD.	B ADDR	00F8H.6	A	
CUATRO.	C ADDR	014AH	R	SEG=PROG
D1OK.	C ADDR	0243H	R	SEG=PROG
D2OK.	C ADDR	025CH	R	SEG=PROG
DESBORD.	C ADDR	01A6H	R	SEG=PROG

```

DOS. . . . . C ADDR 0140H R SEG=PROG
DPH. . . . . D ADDR 0083H A
DPL. . . . . D ADDR 0082H A
EA . . . . . B ADDR 00A8H.7 A
ET0. . . . . B ADDR 00A8H.1 A
EX0. . . . . B ADDR 00A8H.0 A
EXIT . . . . . C ADDR 0090H R SEG=PROG
FAIL . . . . . C ADDR 0168H R SEG=PROG
FIN. . . . . C ADDR 04BFH R SEG=PROG
FLAG . . . . . N NUMB 0020H A
GETASCII . . . . . C ADDR 0174H R SEG=PROG
GETCHAR. . . . . C ADDR 0124H R SEG=PROG
GETNUM . . . . . C ADDR 0230H R SEG=PROG
HOFF . . . . . D ADDR 001BH A
HON. . . . . D ADDR 0019H A
INCR_C . . . . . C ADDR 01C3H R SEG=PROG
INCR_H . . . . . C ADDR 020AH R SEG=PROG
INCR_M . . . . . C ADDR 01F3H R SEG=PROG
INCR_S . . . . . C ADDR 01DAH R SEG=PROG
INTCONF. . . . . C ADDR 011BH R SEG=PROG
INVERN . . . . . N NUMB -----
ISGT . . . . . C ADDR 021EH R SEG=PROG
IT0. . . . . B ADDR 0088H.0 A
LEER_AD. . . . . C ADDR 0038H R SEG=PROG
LEER_MEDIA . . . . . C ADDR 0057H R SEG=PROG
LEER_TEMP. . . . . C ADDR 0028H R SEG=PROG
LF . . . . . N NUMB 000AH A
LUCES. . . . . C ADDR 00F8H R SEG=PROG
LUCESRET . . . . . C ADDR 02C7H R SEG=PROG
LUCES_OK . . . . . C ADDR 0109H R SEG=PROG
MAIN . . . . . C ADDR 03A1H R SEG=PROG
A51 MACRO ASSEMBLER INVERN
18:11:57 PAGE 15

```

25/10/03

```

MOFF . . . . . D ADDR 001CH A
MON. . . . . D ADDR 001AH A
MSG0 . . . . . C ADDR 03B6H R SEG=PROG
MSG1 . . . . . C ADDR 03CCH R SEG=PROG
MSG10. . . . . C ADDR 044CH R SEG=PROG
MSG11. . . . . C ADDR 0466H R SEG=PROG
MSG12. . . . . C ADDR 0480H R SEG=PROG
MSG13. . . . . C ADDR 0483H R SEG=PROG
MSG14. . . . . C ADDR 0487H R SEG=PROG
MSG15. . . . . C ADDR 0490H R SEG=PROG
MSG16. . . . . C ADDR 04A5H R SEG=PROG
MSG17. . . . . C ADDR 04B6H R SEG=PROG
MSG2 . . . . . C ADDR 03E3H R SEG=PROG
MSG3 . . . . . C ADDR 03FAH R SEG=PROG
MSG4 . . . . . C ADDR 0410H R SEG=PROG
MSG5 . . . . . C ADDR 0426H R SEG=PROG
MSG6 . . . . . C ADDR 0433H R SEG=PROG
MSG7 . . . . . C ADDR 0440H R SEG=PROG
MSG8 . . . . . C ADDR 0443H R SEG=PROG
MSG9 . . . . . C ADDR 044AH R SEG=PROG
NEWSEG . . . . . C ADDR 03A6H R SEG=PROG
NORM_T . . . . . C ADDR 0010H R SEG=PROG
NUEVE. . . . . C ADDR 0163H R SEG=PROG
OCHO . . . . . C ADDR 015EH R SEG=PROG
OK . . . . . C ADDR 0171H R SEG=PROG
OK2. . . . . C ADDR 0173H R SEG=PROG
OK_C . . . . . C ADDR 01D5H R SEG=PROG
OK_H . . . . . C ADDR 0219H R SEG=PROG
OK_M . . . . . C ADDR 0205H R SEG=PROG
OK_S . . . . . C ADDR 01EEH R SEG=PROG
P4 . . . . . D ADDR 00E8H A
P5 . . . . . D ADDR 00F8H A
P6 . . . . . D ADDR 00FAH A
P7 . . . . . D ADDR 00DBH A
P8 . . . . . D ADDR 00DDH A
PRINTSYS . . . . . C ADDR 009EH R SEG=PROG
PROG . . . . . C SEG 04BFH REL=UNIT
PSW. . . . . D ADDR 00D0H A
PUNTO. . . . . C ADDR 010FH R SEG=PROG
PUTCHAR. . . . . C ADDR 007EH R SEG=PROG
PUTCRLF. . . . . C ADDR 0091H R SEG=PROG
PUTSEPARATOR . . . . . C ADDR 0098H R SEG=PROG
PUTSTRING. . . . . C ADDR 0086H R SEG=PROG
RD_AD. . . . . B ADDR 00F8H.7 A

```

```

RETARDO. . . . . C ADDR 0000H R SEG=PROG
RETURN . . . . . C ADDR 02C9H R SEG=PROG
RI . . . . . B ADDR 0098H.0 A
SAVENUM. . . . . C ADDR 0278H R SEG=PROG
SBUF . . . . . D ADDR 0099H A
SCON . . . . . D ADDR 0098H A
SEIS . . . . . C ADDR 0154H R SEG=PROG
SIETE. . . . . C ADDR 0159H R SEG=PROG
SP . . . . . D ADDR 0081H A
STACK. . . . . I SEG 0030H REL=UNIT
START. . . . . C ADDR 02CAH R SEG=PROG
TEMP . . . . . D ADDR 001DH A
TEMPD. . . . . D ADDR 00FAH A
TF0. . . . . B ADDR 0088H.5 A
TH0. . . . . D ADDR 008CH A
TH1. . . . . D ADDR 008DH A
TI . . . . . B ADDR 0098H.1 A
TIMECONF . . . . . C ADDR 018FH R SEG=PROG
TL0. . . . . D ADDR 008AH A
TMAX . . . . . D ADDR 001FH A
TMIN . . . . . D ADDR 001EH A
TMOD . . . . . D ADDR 0089H A
A51 MACRO ASSEMBLER INVERN
18:11:57 PAGE 16

```

25/10/03

```

TR0. . . . . B ADDR 0088H.4 A
TR1. . . . . B ADDR 0088H.6 A
TRES . . . . . C ADDR 0145H R SEG=PROG
UNO. . . . . C ADDR 013BH R SEG=PROG
WR_AD. . . . . B ADDR 00F8H.5 A

```

REGISTER BANK(S) USED: 0

ASSEMBLY COMPLETE. 0 WARNING(S), 0 ERROR(S)

Rutinas utilizadas:

A continuación se explican detalladamente las rutinas utilizadas para la implementación del sistema:

- **PRINTSYS**: sencilla rutina que imprime en el puerto serie y sobre una misma línea el estado actual del sistema, es decir, hora, temperatura, y estado de las luces y sistema de calefacción. El formato utilizado es el mismo especificado en el enunciado de la práctica: *Son las hh:mm:ss. La temperatura es tt° C. Calefaccion ON/OFF. Luces ON/OFF.*
- **ISGT**: comprueba que el número almacenado en B sea mayor que el número almacenado en A, devolviendo en este último 1 en caso afirmativo, o 0 en caso contrario.
- **GETNUM**: obtiene un número de máximo dos dígitos a través del puerto serie. Permite validar el número para que no sobrepase un valor indicado en el registro B. Una vez se obtenga un número que cumpla dicha condición, en el propio B se devuelve el valor leído. Esta es la rutina utilizada para la toma de datos al inicio del programa.
- **CHECK_CALEF**: rutina que comprueba la temperatura actual y enciende o apaga el sistema de calefacción en consecuencia. Si la temperatura es inferior al mínimo umbral para el encendido, la calefacción será accionada, y al contrario, si la temperatura es superior al máximo umbral de apagado, ésta se parará.
- **CHECK_LUCES**: realiza las comprobaciones relativas al sistema de iluminación. Si la hora actual es posterior o igual a la hora de encendido, activará las luces. Por contra, si la hora es posterior o igual a la hora de apagado, desactivará la iluminación.
- **RETARDO**: provoca un retardo ejecutando un pequeño bucle en el que no realiza ninguna operación. Sirve para esperar a que las señales enviadas al conversor se estabilicen y ejecute correctamente los cronogramas.
- **NORM_T**: normaliza el dato leído del registro referenciado por TEMP, convirtiéndolo a grados centígrados. Debido a que los valores positivos del conversor están entre 128 y 255, y que el rango de grados centígrados del sensor de temperatura esta entre 0 y 150, se utiliza la siguiente función para ajustar de manera proporcional al incremento del numero proporcionado por el conversor la temperatura en grados centígrados:

$$norm(N) = (N - 128) + (N - 128) * 2 / 12.$$

Básicamente, se resta 128 al número obtenido del conversor y se suma la parte proporcional a los grados para poder alcanzar así los 150 grados posibles del sensor, siendo más pequeña esta parte cuando el valor está mas próximo a 0.

- **LEER_TEMP**: rutina que activa las señales necesarias para que se ejecute el cronograma de lectura del dato del sensor de temperatura, desde el conversor analógico-digital. Se utiliza la rutina **RETARDO** para que las señales se estabilicen. Una vez activadas las señales y transcurrido un pequeño periodo de tiempo el conversor activará una interrupción que será tratada por **LEER_AD**, en la que se ejecuta la parte de lectura del cronograma del conversor, y se obtendrá el dato correspondiente.

- `LEER_AD`: rutina de tratamiento de la interrupción del conversor cuando el dato está disponible. Una vez que se ha iniciado la operación de lectura del conversor analógico-digital, éste manda una interrupción una vez que el dato está disponible, que es tratada por esta rutina. En ella se lee el dato devuelto por el conversor (un número entre 0 y 256). Para ello se realiza un cronograma que activa la salida del conversor y posteriormente se lee el dato. Una vez leído lo normalizamos y convertimos a grados centígrados llamando a `NORM_T` y lo almacenamos en el registro referenciado por `TEMP`.
- `LEER_MEDIA`: lee en 8 ocasiones el dato del sensor de temperatura, y calcula su media aritmética para evitar así posibles oscilaciones del sensor que no son debidas a cambios de la temperatura. De este modo, eliminamos los cambios demasiado bruscos en las medidas. El resultado de la operación es almacenado en el registro referenciado por `TEMP`.

Problemas encontrados:

A lo largo de esta práctica hemos tenido numerosos problemas que nos han retrasado en el desarrollo de la misma. Destacamos algunos de los más molestos:

- Problemas relacionados con el LCD. Desafortunadamente, tras cambiar en dos ocasiones el display LCD e incluso la placa con el microprocesador 8050, nos fué por completo imposible conseguir que el LCD funcionase, debido a que ambos eran de la nueva remesa.
- Problemas relativos al montaje del circuito necesario. El sensor de temperaturas proporcionaba temperaturas muy bajas, debido a que el voltaje de referencia V_{ref} obtenido en la patilla correspondiente del conversor analógico-digital era muy superior al valor esperado (1,14-1,25).
- Problemas derivados del malfuncionamiento del 8051. Por alguna razón que desconocemos, el microprocesador utiliza los registros del banco de registros de propósito general, numerados desde el `0BH` hasta el `18H`. De este modo se hace imposible utilizar dichos registros en tiempo de ejecución, ya que son modificados y los datos que alberguen perdidos. Localizar este problema ha supuesto numerosas trazas rutina a rutina, observando el estado de los registros en todo momento gracias a la ventana dedicada del simulador. El problema se pudo solucionar utilizando otros registros del banco 3 que aparentemente no son modificados, aunque no estamos seguros de que el problema no pueda reproducirse en esta situación.

Anexo: rutinas generales.

A continuación se explican detalladamente todas y cada una de las rutinas utilizadas en este programa, provenientes de prácticas anteriores y reutilizadas.

- **PUTCHAR:** rutina que escribe el carácter almacenado en el registro A, en el puerto serie. Para ello movemos el contenido de A, a SBUF y esperamos hasta que la línea no esté ocupada.
- **GETCHAR:** rutina que lee un carácter ASCII del puerto serie, espera a que se escriba algo en el puerto, lee del puerto, comprueba que el carácter no es 11H, ya que a veces es enviado por el `mon51`, y por último lo almacena en A.
- **PUTSTRING:** escribe una cadena de caracteres por el puerto serie. La cadena de caracteres debe estar apuntada por DPTR. La rutina lee carácter a carácter desde DPTR y llama a la rutina **PUTCHAR**, hasta que llegue un carácter nulo (00H).
- **PUTCRLF:** escribe una nueva línea en el puerto serie (CRLF).
- **PUTSEPARATOR:** escribe el carácter “.” en el puerto serie, moviendo el equivalente ASCII de dicho carácter al registro A y llamando a **PUTCHAR**.
- **CONVERT:** rutina que convierte un carácter ASCII a su equivalente decimal, si el carácter no es un número devuelve un 10 en el registro B, en caso contrario, devuelve el valor de dicho carácter, tanto en el registro B, como en el A. Para ello comprobamos el valor del carácter, con el valor ASCII de los dígitos del 0 al 9, y si coincide con alguno, lo devolvemos en A y en B.
- **GETASCII:** rutina que obtiene de A un número de dos dígitos, y obtiene el primer dígito dividiendo A entre 10 y el segundo dígito restando el primer dígito multiplicado por 10, al número total. Devuelve las decenas en A y las unidades en B.
- **TIMECONF:** rutina que configura el `Timer 0`. Elegimos el modo 1 en `TMOD`, configuramos el `Timer 0` para que se desborde cada centésima de segundo, lo arrancamos e inicializamos los segundos y las centésimas.
- **INTCONF:** rutina que habilita la interrupción del `Timer 0` y del evento externo 0, poniendo a 1 los bits `ET0`, `IT0`, `EX0`, y `EA`.
- **BLANK_TIMER:** configura el `Timer 0` para que se desborde cada centésima, del mismo modo que lo hace **TIMECONF**.
- **INCR_H:** rutina que incrementa en una unidad el número de horas almacenadas en R4, suma 1 al número de horas, le resta 24 y si el resultado es cero inicializa a cero las horas.
- **INCR_M:** rutina que incrementa en una unidad el número de minutos almacenados en R5, suma 1 al número de minutos, le resta 60 y si el resultado es cero inicializa a cero los minutos e incrementa las horas.
- **INCR_S:** rutina que incrementa en una unidad el número de segundos almacenados en R6, suma 1 al número de segundos, le resta 60 y si el resultado es cero inicializa a cero los segundos, e incrementa los minutos.
- **INCR_C:** rutina que incrementa en una unidad el número de centésimas almacenadas en R7, suma 1 al número de centésimas, le resta 100 y si el resultado es cero inicializa a cero las centésimas e incrementa los segundos.

- DESBORD: rutina que trata la interrupción de desbordamiento del `Timer 0` cuando estamos en modo cronómetro. La rutina guarda la palabra de estado en la pila y el registro `B` (que puede estar siendo usado por el programa principal), reinicia el `Timer 0` para que se desborde cada centésima, e incrementa las centésimas. A continuación envía al puerto serie el instante del crono llamando a `PRINT_C`, salvo en el caso de que el número de pulsaciones del botón (`R3`) sea igual a 1, ya que estaríamos en el modo `LAPSUS`, en el que el programa no actualizará la hora en el puerto serie, aunque el timer seguirá funcionando y actualizando la hora correctamente. Por último restaura el registro `B` y la palabra de estado.