

ANEXO 1

PROGRAMACIÓN EN LENGUAJE ASSEMBLER DEL PROGRAMA DE TERMOTERAPIA COMPLETO EN EL MICROCONTROLADOR ATMEGA8

```

.include "C:\VMLAB\includelm8def.inc"
;
; DEFINICIÓN DE VARIABLES:
.def isr =r16
.DEF dtr =r17
.def bandera_zcd =r20
.def cuenta_lcd =r21
.def cruces = r22
.def flag_timer = r23
.def time_1seg = r24
.def fin_del_tratamiento = r25
; ASIGNACIÓN DE PINES DEL LCD AL
ATMEGA:
; RS RW E D7 D6 D5 D4 D3 D2 D1 D0
; PB0 PB1 PB2 PB7 PB6 PB5 PB4 VSS VSS VSS vSS
; reti ; Addr $04
; reti ; Addr $05
.org $06
; rjmp INT_CTC ; Addr $06 Use 'rjmp
myVector'
reti ; Addr $07 to define a interrupt vector
reti ; Addr $08
reti ; Addr $09
reti ; Addr $0A
reti ; Addr $0B This is just an example
reti ; Addr $0C Not all MCUs have the same
reti ; Addr $0D number of interrupt vectors
reti ; Addr $0E
reti ; Addr $0F
reti ; Addr $10

.dseg
.org $61
temperatura: .BYTE 3
setpoint: .BYTE 3
temp_formateada_low: .BYTE 1
temp_formateada_high: .BYTE 1
error_2percent: .BYTE 3
flag_timer: .BYTE 1
smbus_byte_sent: .BYTE 1
temp_centena: .byte 1
temp_decena: .byte 1
temp_unidad: .byte 1
temp_decimal0: .byte 1
temp_decimal1: .byte 1

sp_centena: .byte 1
sp_decena: .byte 1
sp_unidad: .byte 1
sp_decimal: .byte 1
tiempo_tratamiento: .byte 2

bandera_ON: .byte 1

.cseg
.org $00
reset:
rjmp inicio
rjmp ZCD_routine ; Addr $01

; reti ; Addr $02
; reti ; Addr $03

;*INICIO*DEL PROGRAMA*PRINCIPAL
inicio:
; Inicializando la Pila
ldi r18,high(RAMEND)
out sph,r18
ldi r18,low(RAMEND)
out spl,r18
; SALIDA PARA EL DRIVER (PD0) PARA EL
CONTROL ON-OFF
ldi r18,0b00000001
out ddrd,r18
; PD3,PD4,PD5 configurados como entradas
cbi ddr,0
cbi ddr,1
cbi ddr,2
; bandera para el detector de cruce por cero
clr bandera_zcd
; contador de cruces por cero
clr cruces
; bandera para el inicio del temporizador
clr flag_timer
; variable par intercalar el estado ON-OFF
clr time_1seg
; variable para finalizar el tratamiento
clr fin_del_tratamiento
; Valor Inicial del Set Point = 40°
ldi r16, low(4000)
sts setpoint, r16
ldi r16, high(4000)
sts setpoint+1, r16
; Valor Inicial del Error = 2%
ldi r16, low(200)

```

```

sts    error_2percent, r16
ldi    r16, high(200)
sts    error_2percent+1, r16
CLR    R16
sts    bandera_ON,R16

;      CONFIGURACION DE LAS
INTERRUPCIONES EXTERNAS
in r18,MCUCR ; Solicitud de Interrupción en flanco de
ori r18,1 ; subida
out MCUCR,r18
in r18,GICR ; Habilitamos la interrupción externa 0
ori r18,$40
out GICR,r18
;      Configuración del TWI a 27.8KHZ
ldi    r18,10
out    TWBR,r18
;      Apunta el puntero X a la variable
"temperatura"
ldi    xh,high(temperatura)
ldi    xl,low(temperatura)

;      INICIALIZAMOS EL MENU PRINCIPAL
rcall  lcd_ini
Configuracion:
rcall  settings
rcall  sensor_display
rcall  formatear_tsensor_bcd_a_dec
ldi    isr,0b11000000
rcall  write_ins
rcall  setpoint_display
lds    dtr,sp_centena
rcall  write_dtr
lds    dtr,sp_decena
rcall  write_dtr
lds    dtr,sp_unidad
rcall  write_dtr
ldi    dtr,'.'
rcall  write_dtr
lds    dtr,sp_decimal
rcall  write_dtr
ldi    dtr,'0'
rcall  write_dtr
ldi    dtr,0b11011111
rcall  write_dtr
;rcall  refrescar_lectura_sensor
;      ACTIVAMOS LAS INTERRUPCIONES
SEI

;      CONFIGURACION DEL TWI A 27.8KHZ
ldi    r18,10
out    TWBR,r18
ldi    xh,high(temperatura)
ldi    xl,low(temperatura)
clr    cuenta_lcd

;      EMPIEZA EL SISTEMA DE CONTROL
comparación_bandera_zcd:
cpi    bandera_zcd,1
brne   comparación_bandera_zcd
clr    bandera_zcd
rcall  leer_temperatura
rcall  formatear_centigrados
rcall  formatear_temperatura_lcd
rcall  mostrar_temp_sensor

;      VERIFICACION DE LA FINALIZACION DEL
TIMEPO DE TRATAMIENTO
cpi    fin_del_tratamiento,1
brne   Comparo_setpoint_temperatura
clr    bandera_zcd
clr    cruces
clr    flag_timer
clr    time_1seg
clr    fin_del_tratamiento
cbi    portd,0
CLI
LDI    isr,1
rcall  write_ins
rjmp   Configuracion

;      COMPARACIÓN DEL SET POINT Y LA
TEMPERATURA
Comparo_setpoint_temperatura:
LDS    R16, setpoint+1
LDS    R17, temp_formateada_high
cp     R17, R16 ;COMPARO LOS MSB
brlo   MENOR ; TEMPERATURA <
SET POINT
breq   comparo_en_LSB ; TEMPERATURA
= SET POINT
rjmp   MAYOR ; TEMPERATURA >
SET POINT

comparo_en_LSB:
LDS    R16, setpoint
LDS    R17, temp_formateada_low
cp     R17, R16 ;COMPARO LOS LSB
brlo   MENOR ; TEMPERATURA <
SET POINT
breq   IGUAL
; TEMPERATURA = SET POINT
rjmp   MAYOR ; TEMPERATURA >
SET POINT

MENOR:
cpi    flag_timer,1
breq   comparo_error_menor
sbi    portd,0
rjmp   comparación_bandera_zcd
comparo_error_menor:
LDS    R16, setpoint

```

```
LDS R17, setpoint+1
LDS R18, temp_formateada_low
LDS R19, temp_formateada_high

; RESTA HEXADECIMAL DE 2 BYTES (32 BITS)
SBC R16,R18 ; ERROR '-' LSB
SBC R17,R19 ; ERROR '-' MSB

LDS R18, error_2percent
LDS R19, error_2percent+1
cp R17, R19 ;COMPARO LOS MSB
brlo AUMENTAR ; ERROR < 2.0
%
breq comparo_errormenor_LSB ; ERROR =
2.0 %
rjmp AUMENTAR ; ERROR > 2.0 %
comparo_errormenor_LSB:
cp R16, R18 ;COMPARO LOS LSB
brlo AUMENTAR ; ERROR < 2.0 %
breq AUMENTAR ; ERROR = 2.0 %
rjmp AUMENTAR ; ERROR > 2.0 %

MAYOR:
cpi flag_timer,1
breq comparo_error_mayor
ldi flag_timer,1 ;
comparo_error_mayor:
LDS R16, setpoint
LDS R17, setpoint+1
LDS R18, temp_formateada_low
LDS R19, temp_formateada_high

; RESTA HEXADECIMAL DE 2 BYTES (32 BITS)
SUB R18,R16 ; ERROR '+' LSB
SBC R19,R17 ; ERROR '+' MSB

LDS R16, error_2percent
LDS R17, error_2percent+1
cp R19, R17 ;COMPARO LOS MSB
brlo REDUCIR ; ERROR <
2.0 %
breq comparo_errormayor_LSB ; ERROR = 2.0
%
rjmp REDUCIR ; ERROR > 2.0 %
comparo_errormayor_LSB:
cp R18, R16 ;COMPARO LOS LSB
brlo REDUCIR ; ERROR < 2.0 %
breq REDUCIR ; ERROR = 2.0 %
rjmp REDUCIR ; ERROR > 2.0 %

IGUAL:
sbi portd,0
rjmp comparación_bandera_zcd
```

```
REDUCIR:
clr cruces
cbi portd,0
rjmp comparación_bandera_zcd
AUMENTAR:
sbi portd,0
rjmp comparación_bandera_zcd
; TERMINO DEL PROGRAMA PRINCIPAL
fin:rjmp fin

; **FIN*DEL PROGRAMA*PRINCIPAL

cuenta_tiempo_tratamiento:
push r18
push r19
ldi isr,0b10001100
rcall write_ins
ldi dtr,'t'
rcall write_dtr
ldi dtr,'='
rcall write_dtr
lds dtr,tiempo_tratamiento
rcall write_dtr
lds dtr,tiempo_tratamiento+1
rcall write_dtr
lds r18,tiempo_tratamiento
lds r19,tiempo_tratamiento+1

comparar_digitos:
cpi r19,'0'
breq segundos_unidad_cero
dec r19
sts tiempo_tratamiento+1,r19
rjmp fin_tiempo_tratamiento

segundos_unidad_cero:
cpi r18,'0'
breq fin_tiempo_tratamiento
ldi r19,'9'
sts tiempo_tratamiento+1,r19
dec r18
sts tiempo_tratamiento,r18

fin_tiempo_tratamiento:
pop r19
pop r18
ret

; Subrutina para dar formato decimal al valor
BCD del Set Point
formatear_tsensor_bcd_a_dec:
push r18
push yl
push yh
ldi yh,high(5000)
ldi yl,low(5000)
```

```

lds      r18,sp_unidad
subi     r18,48
clr      dtr

sumar_centenas_sp:
cp       r18,dtr
breq     fin_sumar_centenas_sp
adiw     yh:yl,50
adiw     yh:yl,50
inc      dtr
rjmp     sumar_centenas_sp

fin_sumar_centenas_sp:
lds      r18,sp_decimal
subi     r18,48
ldi      dtr,10
mul      r18,dtr
mov      r18,r0
clc
add      yl,r18
in       dtr,sreg
andi     dtr,1
add      yh,dtr
sts      setpoint,yl
sts      setpoint+1,yh
pop      yh
pop      yl
pop      r18
ret

;      Subrutina Para el Refrescamiento de la
lectura del sensor
refrescar_lectura_sensor:
rcall    leer_temperatura
rcall    formatear_centigrados
rcall    formatear_temperatura_lcd
rcall    mostrar_temp_sensor
ret

mensaje_proyecto1a:
.db "INICIANDO",0

mensaje_proyecto1b:
.db " ESPERE",0

mensaje_terapia:
.db "TERAPIA",0

mensaje_termica:
.db "TERMICA",0

proyecto_electronico1_display:
push     r18
rjmp     men_proy1b_fin ;ERICK CAMBIO ESTO
ldi      isr,0b10000100

rcall    write_ins
ldi      zh,high(mensaje_proyecto1a*2)
ldi      zl,low(mensaje_proyecto1a*2)
clr      r18

comp_menproy1a:
lpm      dtr,z+
cpi      dtr,0
breq     men_proy1a_fin
rcall    write_dtr
rjmp     comp_menproy1a

men_proy1a_fin:
ldi      isr,0b11000010
rcall    write_ins
ldi      zh,high(mensaje_proyecto1b*2)
ldi      zl,low(mensaje_proyecto1b*2)
clr      r18

comp_menproy1b:
lpm      dtr,z+
cpi      dtr,0
breq     men_proy1b_fin
rcall    write_dtr
rjmp     comp_menproy1b

men_proy1b_fin:
;rcall   espera_1seg
;rcall   espera_1seg
ldi      isr,1
rcall    write_ins
ldi      isr,0b10000100
rcall    write_ins
ldi      zh,high(mensaje_terapia*2)
ldi      zl,low(mensaje_terapia*2)
clr      r18

comp_menterapia:
lpm      dtr,z+
cpi      dtr,0
breq     men_terapia_fin
rcall    write_dtr
rjmp     comp_menterapia

men_terapia_fin:
ldi      isr,0b11000100
rcall    write_ins
ldi      zh,high(mensaje_termica*2)
ldi      zl,low(mensaje_termica*2)
clr      r18

comp_mentermica:
lpm      dtr,z+
cpi      dtr,0
breq     men_termica_fin
rcall    write_dtr

```

```

rjmp    comp_mentermica

men_termica_fin:
rcall   espera_1seg
;rcall  espera_1seg
;rcall  espera_1seg
ldi     isr,1
rcall   write_ins
clr     zl
clr     zh
pop     r18
ret

mensaje_opcionesa:
.db " Presione boton",0

mensaje_opcionesb:
.db " opciones "O"",0

menu_opciones_display:
push   r18
ldi    zh,high(mensaje_opcionesa*2)
ldi    zl,low(mensaje_opcionesa*2)
clr    r18

comp_menopcionesa:
lpm    dtr,z+
cpi    dtr,0
breq   men_opcionesa_fin
rcall  write_dtr
rjmp   comp_menopcionesa

men_opcionesa_fin:
ldi    isr,0b11000000
rcall  write_ins
ldi    zh,high(mensaje_opcionesb*2)
ldi    zl,low(mensaje_opcionesb*2)
clr    r18

comp_menopcionesb:
lpm    dtr,z+
cpi    dtr,0
breq   men_opcionesb_fin
rcall  write_dtr
rjmp   comp_menopcionesb

men_opcionesb_fin:
comparacion_tecla_O1:
in     r18,pinc
andi   r18,0b00000100
cpi    r18,0b00000100
brne   comparacion_tecla_O1

comparacion_tecla_O2:
in     r18,pinc
andi   r18,0b00000100

cpi    r18,0b00000100
breq   comparacion_tecla_O2

mensaje_tiempo_tratamiento:
.db "t terapia=30 seg ",0

tiempo_tratamiento_display:
push   r18
ldi    isr,0b10000000
rcall  write_ins
ldi    zh,high(mensaje_tiempo_tratamiento*2)
ldi    zl,low(mensaje_tiempo_tratamiento*2)
clr    r18

comp_tratamiento:
lpm    dtr,z+
cpi    dtr,0
breq   men_tratamiento_fin
rcall  write_dtr
rjmp   comp_tratamiento

men_tratamiento_fin:
clr    zl
clr    zh
pop    r18
ret

clear_segunda_linea:
push   r18
ldi    isr,0b11000000
rcall  write_ins
clr    r18

comp_clear_segunda:
ldi    dtr,32
rcall  write_dtr
inc    r18
cpi    r18,16
brne   comp_clear_segunda
pop    r18
ret

mensaje_continuar:
.db "<continuar con O>",0

mensaje_continuar_display:
push   r18
ldi    isr,0b11000000
rcall  write_ins

```

```

ldi    zh,high(mensaje_continuar*2)
ldi    zl,low(mensaje_continuar*2)
clr    r18

comp_continuar:
lpm    dtr,z+
cpi    dtr,0
breq   men_continuar_fin
rcall  write_dtr
rjmp   comp_continuar

men_continuar_fin:
clr    zl
clr    zh
pop    r18
ret

mensaje_regresar:
.db "para salir: O",0

mensaje_regresar_display:
push  r18
ldi   isr,0b11000000
rcall write_ins
ldi   zh,high(mensaje_continuar*2)
ldi   zl,low(mensaje_continuar*2)
clr   r18

comp_regresar:
lpm    dtr,z+
cpi    dtr,0
breq   men_regresar_fin
rcall  write_dtr
rjmp   comp_regresar

men_regresar_fin:
clr    zl
clr    zh
pop    r18
ret

mensaje_reconfigurar:
.db "reconfigurar? ",0

mensaje_si_no:
.db "- :no + :si ",0

mensaje_reconfigurar_display:
push  r18
ldi   isr,0b10000000
rcall write_ins
ldi   zh,high(mensaje_reconfigurar*2)
ldi   zl,low(mensaje_reconfigurar*2)
clr   r18

comp_reconfigurar:
lpm    dtr,z+
cpi    dtr,0
breq   men_reconfigurar_fin
rcall  write_dtr
rjmp   comp_reconfigurar

men_reconfigurar_fin:
ldi   isr,0b11000000
rcall write_ins
ldi   zh,high(mensaje_si_no*2)
ldi   zl,low(mensaje_si_no*2)
clr   r18

comp_si_no:
lpm    dtr,z+
cpi    dtr,0
breq   men_si_no_fin
rcall  write_dtr
rjmp   comp_si_no

men_si_no_fin:
clr    zl
clr    zh
pop    r18
ret

; SUBROUTINA PARA PARA PRESENTACION
DEL EQUIPO
settings:
push  r18
rcall proyecto_electronico1_display
rcall menu_opciones_display

configuracion_set_point:
ldi   isr,0b10000000
rcall write_ins
rcall setpoint_display
ldi   dtr,'0'
sts   sp_centena,dtr
ldi   dtr,'5'
sts   sp_decena,dtr
rcall write_dtr
ldi   dtr,'0'
sts   sp_unidad,dtr
rcall write_dtr
ldi   dtr,','
rcall write_dtr
ldi   dtr,'0'
sts   sp_decimal,dtr
rcall write_dtr
ldi   dtr,0b11011111
rcall write_dtr
ldi   dtr,'3'
sts   tiempo_tratamiento,dtr
rcall mensaje_continuar_display
    
```

```
comparacion_tecla_mas:
in      r18,pinc
andi   r18,0b00000010
cpi    r18,0b00000010
brne   comparacion_tecla_menos
```

```
presiono_tecla_mas:
in      r18,pinc
andi   r18,0b00000010
cpi    r18,0b00000010
breq   presiono_tecla_mas
```

```
solto_tecla_mas:
lds    dtr,sp_unidad
cpi    dtr,'4'
breq   comparacion_tecla_mas
lds    dtr,sp_decimal
cpi    dtr,'0'
breq   sp_decimal_cero_aumentar
ldi    isr,0b10000110
rcall  write_ins
lds    dtr,sp_unidad
inc    dtr
rcall  write_dtr
sts    sp_unidad,dtr
ldi    isr,0b10001000
rcall  write_ins
ldi    dtr,'0'
rcall  write_dtr
sts    sp_decimal,dtr
rjmp  comparacion_tecla_mas
```

```
sp_decimal_cero_aumentar:
ldi    isr,0b10001000
rcall  write_ins
ldi    dtr,'5'
rcall  write_dtr
sts    sp_decimal,dtr
rjmp  comparacion_tecla_mas
```

```
comparacion_tecla_menos:
in      r18,pinc
andi   r18,0b00000001
cpi    r18,0b00000001
brne   comparacion_tecla_opciones
```

```
presiono_tecla_menos:
in      r18,pinc
andi   r18,0b00000001
cpi    r18,0b00000001
breq   presiono_tecla_menos
```

```
solto_tecla_menos:
lds    dtr,sp_unidad
cpi    dtr,'0'
breq   comparacion_tecla_menos1
```

```
rjmp   se_puede_restar
comparacion_tecla_menos1:
lds    dtr,sp_decimal
cpi    dtr,'5'
brne   comparacion_tecla_mas
```

```
se_puede_restar:
lds    dtr,sp_decimal
cpi    dtr,'5'
breq   sp_decimal_cinco_disminuir
ldi    isr,0b10000110
rcall  write_ins
lds    dtr,sp_unidad
dec    dtr
rcall  write_dtr
sts    sp_unidad,dtr
ldi    dtr,'5'
sts    sp_decimal,dtr
ldi    isr,0b10001000
rcall  write_ins
rcall  write_dtr
rjmp  comparacion_tecla_mas
```

```
sp_decimal_cinco_disminuir:
ldi    isr,0b10001000
rcall  write_ins
ldi    dtr,'0'
rcall  write_dtr
sts    sp_decimal,dtr
rjmp  comparacion_tecla_mas
```

```
comparacion_tecla_opciones:
in      r18,pinc
andi   r18,0b00000100
cpi    r18,0b00000100
breq   presiono_tecla_opciones
rjmp  comparacion_tecla_mas
```

```
presiono_tecla_opciones:
in      r18,pinc
andi   r18,0b00000100
cpi    r18,0b00000100
breq   presiono_tecla_opciones
rcall  espera_500ms
ldi    isr,1
rcall  write_ins
rcall  tiempo_tratamiento_display
rcall  mensaje_regresar_display
```

```
comparacion2_tecla_mas:
in      r18,pinc
andi   r18,0b00000010
cpi    r18,0b00000010
brne   comparacion2_tecla_menos
```

```

presiono2_tecla_mas:
in          r18,pinc
andi       r18,0b00000010
cpi        r18,0b00000010
breq       presiono2_tecla_mas

solto2_tecla_mas:
lds        dtr,tiempo_tratamiento
cpi        dtr,'9'
breq       comparacion2_tecla_mas
inc        dtr
inc        dtr
inc        dtr
sts        tiempo_tratamiento,dtr
ldi        isr,0b10001010
rcall      write_ins
rcall      write_dtr
rjmp       comparacion2_tecla_mas

comparacion2_tecla_menos:
in          r18,pinc
andi       r18,0b00000001
cpi        r18,0b00000001
brne       comparacion2_tecla_O

presiono2_tecla_menos:
in          r18,pinc
andi       r18,0b00000001
cpi        r18,0b00000001
breq       presiono2_tecla_menos
lds        dtr,tiempo_tratamiento
cpi        dtr,'3'
breq       comparacion2_tecla_mas
dec        dtr
dec        dtr
dec        dtr
sts        tiempo_tratamiento,dtr
ldi        isr,0b10001010
rcall      write_ins
rcall      write_dtr
rjmp       comparacion2_tecla_mas

comparacion2_tecla_O:
in          r18,pinc
andi       r18,0b00000100
cpi        r18,0b00000100
brne       comparacion2_tecla_mas

presiono2_tecla_O:
in          r18,pinc
andi       r18,0b00000100
cpi        r18,0b00000100
breq       presiono2_tecla_O
rcall      mensaje_reconfigurar_display

comparacion3_tecla_mas:
in          r18,pinc
andi       r18,0b00000010
cpi        r18,0b00000010
brne       comparacion3_tecla_menos

presiono3_tecla_mas:
in          r18,pinc
andi       r18,0b00000010
cpi        r18,0b00000010
breq       presiono3_tecla_mas
ldi        dtr,'3'
sts        tiempo_tratamiento,dtr
ldi        isr,1
rcall      write_ins
rjmp       configuracion_set_point

comparacion3_tecla_menos:
in          r18,pinc
andi       r18,0b00000001
cpi        r18,0b00000001
brne       comparacion3_tecla_mas

presiono3_tecla_menos:
in          r18,pinc
andi       r18,0b00000001
cpi        r18,0b00000001
breq       presiono3_tecla_menos
rcall      espera_1seg
ldi        isr,1
rcall      write_ins
ldi        dtr,'0'
sts        tiempo_tratamiento+1,dtr
pop        r18
ret

; SUBROUTINA PARA MOSTRAR LA
; TEMPERATURA SEL SENSOR EN GRADOS
; CENTIGRADOS
mostrar_temp_sensor:
push       r18
ldi        r18,48
ldi        isr,0b10000100 ;Me dirijo a la posición 5 del
LCD
rcall      write_ins
lds        dtr,temp_centena
add        dtr,r18
rcall      write_dtr
lds        dtr,temp_decena
add        dtr,r18
rcall      write_dtr
lds        dtr,temp_unidad
add        dtr,r18
rcall      write_dtr
ldi        dtr,'.'
rcall      write_dtr
lds        dtr,temp_decimal0
    
```



```

add    dtr,r18
rcall  write_dtr
lds    dtr,temp_decimal1
add    dtr,r18
rcall  write_dtr
ldi    dtr,0b11011111 ;Con esto muestro el grado
rcall  write_dtr
pop    r18
ret

;SUBROUTINA QUE RESTA 10000 AL REGISTRO "Y"
resta_10000:
push  r18
clr   r18

compara_resta_10000:
inc   r18
sbiw  yh:yl,50
sbiw  yh:yl,50
cpi   r18,100
brne  compara_resta_10000
pop   r18
ret

;SUBROUTINA QUE RESTA 1000 AL REGISTRO "Y"
resta_1000:
push  r18
clr   r18

compara_resta_1000:
inc   r18
sbiw  yh:yl,50
sbiw  yh:yl,50
cpi   r18,10
brne  compara_resta_1000
pop   r18
ret

;SUBROUTINA QUE DA FORMATO A LA
TEMPERATURA RECIBIDA DEL SENSOR
;LA GUARDA EN FORMATO DECIMAL (BCD) A LAS
VARIABLES temp_decena,
;temp_decena, temp_unidad,temp_decimal0 y
temp_decimal1
formatear_temperatura_lcd:
push  r18
push  yh
push  yl
clr   r18
sts   temp_centena,r18
sts   temp_decena,r18
sts   temp_unidad,r18
sts   temp_decimal0,r18
sts   temp_decimal1,r18
lds   yh,temp_formateada_high
lds   yl,temp_formateada_low

clr   r18

comparar_centena:
cpi   yh,$27
brsh  mayor_oigual_10000
rjmp  fin_comparar_centena

mayor_oigual_10000:
cpi   yh,$27
breq  igual_10000
rjmp  mayor_10000

igual_10000:
cpi   yl,$10
brlo  fin_comparar_centena

mayor_10000:
inc   r18
sts   temp_centena,r18
rcall resta_10000
rjmp  comparar_centena

fin_comparar_centena:
clr   r18

comparar_decena:
cpi   yh,$03
brsh  mayor_oigual_1000
rjmp  fin_comparar_decena

mayor_oigual_1000:
cpi   yh,$03
breq  igual_1000
rjmp  mayor_1000

igual_1000:
cpi   yl,$E8
brlo  fin_comparar_decena

mayor_1000:
inc   r18
sts   temp_decena,r18
rcall resta_1000
rjmp  comparar_decena

fin_comparar_decena:
clr   r18

comparar_unidad:
cpi   yh,$00
brsh  mayor_oigual_100
rjmp  fin_comparar_unidad

mayor_oigual_100:
cpi   yh,$00
breq  igual_100

```

```

rjmp    mayor_100

igual_100:
cpi    yl,$64
brlo   fin_comparar_unidad

mayor_100:
inc    r18
sts    temp_unidad,r18
sbiw   yh:yl,50
sbiw   yh:yl,50
rjmp   comparar_unidad

fin_comparar_unidad:
clr    r18

comparar_decimal0:
cpi    yl,10
brsh   mayor_oigual_10
rjmp   fin_comparar_decimal0

mayor_oigual_10:
inc    r18
sts    temp_decimal0,r18
sbiw   yh:yl,10
rjmp   comparar_decimal0

fin_comparar_decimal0:
comparar_decimal1:
sts    temp_decimal1,yl
clr    r18
pop    yl
pop    yh
pop    r18
ret

;SUBROUTINA QUE CONVIERTE LA INFORMACIÓN
ARROJADA
;POR EL SENSOR DE 16 BITS A VALORES DE
TEMPERATURA
;SEGÚN LA FÓRMULA DE CONVERSIÓN
formatear_centigrados:
;FÓRMULA DE TEMPERATURA OBJETO DEL
SENSOR:
;
;           To(K)= ToREG*0.02           K/LSB
;
;
;MÁXIMO VALOR PARA ToREG: 32767 ($7FFF)
;T0(K)= 65534 /100

;MÁXIMO VALOR DE REGISTRO DE 16BITS DEL
ATMEGA:65535
;TSREG*0.02-273 = (TSREG*2-27300)/100 ---->
temperatura en Celsius *100
push   r18

push   yh
push   yl
;Multiplicación x2
clr    yl
clr    yh
lds    yl,temperatura
lds    yh,temperatura+1
andi  yh,0b01111111
clc
rol    yl
rol    yh
;Tengo que restar ToREG-27300
;27300=13*7*5*5*3*2*2 =182 *150
clr    r18

registro_no_zero:
sbiw   yh:yl,50
sbiw   yh:yl,50
sbiw   yh:yl,50
inc    r18
cpi    r18,182
brne   registro_no_zero
sts    temp_formateada_low,yl
sts    temp_formateada_high,yh
clr    yh
clr    yl
pop    yl
pop    yh
pop    r18
ret

;RUTINA QUE ESPERA 200us
espera_200us:
push   r16
clr    r16

compara1:
inc    r16
cpi    r16,50
brne   compara1
pop    r16
ret

;RUTINA QUE ESPERA 43ms
espera_43ms:
push   r16
clr    r16

compara2:
rcall  espera_200us
inc    r16
cpi    r16,200
brne   compara2
pop    r16
ret
    
```

```

;RUTINA QUE ESPERA 1seg
espera_1seg:
push    r18
push    r19
; Preescalamiento 1:1024
; Modo CTC
LDI     R18, (0<<COM1A1 | 0<<COM1A0 |
0<<WGM11 | 0<<WGM10)
OUT     TCCR1A, R18
LDI     R19, high(976) ; Valor de Registro OCR1A:
976
LDI     R18,low(976)
OUT     OCR1AH, R19 ; (1us)x(1024)x(OCR1A + 1) =
1seg
OUT     OCR1AL, R18
LDI     R18, (0<<WGM13 | 1<<WGM12 | 1<<CS12 |
0<<CS11 | 1<<CS10)
OUT     TCCR1B, R18

espera1seg_compare:
in      r18,TIFR
sbrs   r18,4
rjmp   espera1seg_compare
LDI     R18, (0<<WGM13 | 1<<WGM12 | 0<<CS12 |
0<<CS11 | 0<<CS10)
OUT     TCCR1B, R18
ldi    r18,(1<<OCF1A)
out    TIFR,r18
pop    r19
pop    r18
ret

;RUTINA QUE ESPERA 500ms
espera_500ms:
push    r18
push    r19
; Preescalamiento 1:1024
; Modo CTC
LDI     R18, (0<<COM1A1 | 0<<COM1A0 |
0<<WGM11 | 0<<WGM10)
OUT     TCCR1A, R18
LDI     R19, high(488) ; Valor de Registro OCR1A:
488
LDI     R18,low(488)
OUT     OCR1AH, R19 ; (1us)x(1024)x(OCR1A + 1) =
0.5
OUT     OCR1AL, R18
LDI     R18, (0<<WGM13 | 1<<WGM12 | 1<<CS12 |
0<<CS11 | 1<<CS10)
OUT     TCCR1B, R18

espera500ms_compare:
in      r18,TIFR
sbrs   r18,4
rjmp   espera500ms_compare

LDI     R18, (0<<WGM13 | 1<<WGM12 | 0<<CS12 |
0<<CS11 | 0<<CS10)
OUT     TCCR1B, R18
ldi    r18,(1<<OCF1A)
out    TIFR,r18
pop    r19
pop    r18
ret

;RUTINA QUE VERIFICA EL Busy Flag
check_bf:
push    r16
push    r19
cbi     DDRB,4 ;configuro pb4-7 como entrada
cbi     DDRB,5
cbi     DDRB,6
cbi     DDRB,7
clr     r19

compara_bf:
ldi    r16,0b00000010
out    portb,r16
nop
sbi    portb,2
cbi    portb,2
in     r19,pinb
ldi    r16,0b00000010
out    portb,r16
nop
sbi    portb,2
cbi    portb,2
in     r16,pinb
andi  r16,0b11110000
swap  r16
andi  r19,0b11110000
or    r19,r16
andi  r19,0b10000000
cpi   r19,0b10000000
breq  compara_bf
sbi   DDRB,4
sbi   DDRB,5
sbi   DDRB,6
sbi   DDRB,7
pop   r19
pop   r16
ret

;SUBROUTINA QUE ESCRIBE UNA INSTRUCCIÓN
write_ins:
push    r17
push    r18
rcall   check_bf
mov     r17,isr
mov     r18,isr
andi   r17,0b11110000
andi   r18,0b00001111
    
```

```

swap    r18
out     portb,r17
nop
sbi     portb,2
nop
cbi     portb,2
nop
nop
out     portb,r18
nop
sbi     portb,2
nop
cbi     portb,2
nop
rcall   check_bf
pop     r18
pop     r17
ret

;RUTINA DE ENVÍO DE DATOS AL LCD
write_dtr:
push   r16
push   r18
rcall  check_bf
mov    r16,dtr
mov    r18,dtr
andi   r16,0b11110000
andi   r18,0b00001111
swap   r18
ori    r16,1
ori    r18,1
out    portb,r16
nop
sbi    portb,2
nop
cbi    portb,2
nop
out    portb,r18
nop
sbi    portb,2
nop
cbi    portb,2
nop

rcall  check_bf
pop    r18
pop    r16
ret

;RUTINA DE INICIALIZACIÓN DEL LCD
lcd_ini:
push   r17
ldi    r17,0b11110111
out    DDRB,r17
rcall  espera_43ms
rcall  espera_43ms

rcall  espera_43ms
rcall  espera_43ms
rcall  espera_43ms
ldi    r17,0b00110000
out    portb,r17
nop
sbi    portb,2
nop
cbi    portb,2
rcall  espera_43ms
ldi    r17,0b00110000
out    portb,r17
nop
sbi    portb,2
nop
cbi    portb,2
rcall  espera_200us
ldi    r17,0b00110000
out    portb,r17
nop
sbi    portb,2
nop
cbi    portb,2
rcall  check_bf
ldi    r17,0b00100000
out    portb,r17
nop
sbi    portb,2
nop
cbi    portb,2

;FUNCTION SET:
;configurado a 4bits
ldi    isr,0b00101000
rcall  write_ins
;DISPLAY ON/OFF CONTROL:
;encendido,cursor se mueve a la derecha, blink activo
ldi    isr,0b00001111
rcall  write_ins
;ENTRY MODE SET:
;desplaza cursor automáticamente al escribir datos
ldi    isr,0b00000110
rcall  write_ins
pop    r17
ret

mensaje_sensor:
.db "Tse=",0

sensor_display:
push   r18
ldi    isr,0b10000000
rcall  write_ins
ldi    zh,high(mensaje_sensor*2)
ldi    zl,low(mensaje_sensor*2)
clr    r18
    
```

```

comp_men1:
lpm    dtr,z+
cpi    dtr,0
breq   men1fin
rcall  write_dtr
rjmp   comp_men1

men1fin:
clr    zl
clr    zh
pop    r18
ret

mensaje_set_point:
.db "Ts.p=",0

setpoint_display:
push  r18
ldi   zh,high(mensaje_set_point*2)
ldi   zl,low(mensaje_set_point*2)
clr   r18

comp_men2:
lpm    dtr,z+
cpi    dtr,0
breq   men2fin
rcall  write_dtr
rjmp   comp_men2

men2fin:
clr    zl
clr    zh
pop    r18
ret

start_twi:
push  r18
ldi   r18,(1<<TWINT)|(1<<TWSTA)|(1<<TWEN)
|(1<<TWEA)|(0<<TWSTA)
out   TWCR,r18
rcall TWINT_wait
in    r18,TWSR
andi  r18,0b11111000
cpi   r18,$08
brne  start_fail
rjmp  fin_start_twi

start_fail:
ldi   r18,(1<<TWINT)|(0<<TWSTA)|(1<<TWSTO)|(
1<<TWEN)
out   TWCR,r18

fin_start_twi:
pop   r18

ret

TWINT_wait:
push  r18

check_TWINT:
in    r18,TWCR
sbrs  r18,TWINT
rjmp  check_TWINT
pop   r18
ret

address_recog:
push  r18
ldi   r18,0b10110100
out   TWDR,r18
ldi   r18,(1<<TWINT)|(0<<TWSTA)|(0<<TWSTO)|(
1<<TWEN)|(1<<TWEA)
out   TWCR,r18
rcall TWINT_wait
in    r18,TWSR
andi  r18,$F8
cpi   r18,$18
brne  fail_address_recog

fail_address_recog:
pop   r18
ret

repeat_start:
push  r18
ldi   r18,(1<<TWINT)|(1<<TWSTA)|(0<<TWSTO)|(
1<<TWEN)|(1<<TWEA)
out   TWCR,r18
rcall TWINT_wait
in    r18,TWSR
andi  r18,$F8
cpi   r18,$10
brne  fail_repeat_start

fail_repeat_start:
pop   r18
ret

byte_sent:
push  r18
lds   r18,smbus_byte_sent ;Comando para leer
temperatura objeto del sensor
out   TWDR,r18
ldi   r18,(1<<TWINT)|(0<<TWSTA)|(0<<TWSTO)|(
1<<TWEN)|(1<<TWEA)
out   TWCR,r18
rcall TWINT_wait

```

```

in            r18,TWSR
andi         r18,$F8
cpi         r18,$28
brne        byte_sent_fail

byte_sent_fail:
pop         r18
ret

slave_read:
push        r18
push        r19
clr         r19
ldi         r18,0b10110101
out         TWDR,r18
ldi
            r18,(1<<TWINT)|(0<<TWSTA)|(0<<TWSTO)|(
1<<TWEN)|(1<<TWEA)
out         TWCR,r18
rcall       TWINT_wait
in          r18,TWSR
andi         r18,$F8
cpi         r18,$40
brne        fail_slave_read

data_delivery:
ldi
            r18,(1<<TWINT)|(0<<TWSTA)|(0<<TWSTO)|(
1<<TWEN)|(1<<TWEA)
out         TWCR,r18
rcall       TWINT_wait
in          r18,TWSR
andi         r18,$F8
cpi         r18,$50
brne        fin_slave_read
in          r18,TWDR
st          x+,r18
inc         r19
cpi         r19,3
breq        fin_slave_read
rjmp        data_delivery

fail_slave_read:
fin_slave_read:
ldi         xh,high(temperatura)
ldi         xl,low(temperatura)
pop         r19
pop         r18
ret

stop_twi:
push        r18
ldi         r18,(1<<TWINT)|(1<<TWSTO)|(1<<TWEN)
out         TWCR,r18

check_TWSTO:
in          r18,TWCR
sbrc        r18,TWSTO
rjmp        check_TWSTO
pop         r18
ret

leer_temperatura:
push        r18
ldi         r18,0b00000111 ;Comando para leer
temperatura objeto del sensor
sts         smbus_byte_sent,r18
rcall       start_twi
rcall       adress_recog
rcall       byte_sent
rcall       repeat_start
rcall       slave_read
rcall       stop_twi
pop         r18
ret

;INTERRUPCIÓN dcz 60 hz
;ESTA INTERRUPCIÓN SE HARÁ CADA 8.33 ms
ZCD_routine:
PUSH        R18
IN          R18,sreg
PUSH        R18
PUSH        R19
PUSH        R17
PUSH        R16

;          dcz
;          ZONA DE REFRESCAMIENTO DE DATA
DEL SENSOR
cpi         flag_timer, 1
;          ;INICIA LA DISMINUCIÓN
DEL TIEMPO DE TRATAMIENTO CADA 1 SEG.
brne        SEGUIR
inc         time_1seg
cpi         time_1seg, 180
brne        SEGUIR
;          REDUCIR EL TIEMPO DE TRATAMIENTO:
clr         time_1seg
;rcall      cuenta_tiempo_tratamiento
lds         r18,tiempo_tratamiento
lds         r19,tiempo_tratamiento+1

lazo_cuenta_cero:
cpi         r18,'0'
brne        SEGUIR_reduciendo
cpi         r19,'0'
brne        SEGUIR_reduciendo
ldi         fin_del_tratamiento, 1
rjmp        Comparación_temperatura

SEGUIR_reduciendo:
rcall       cuenta_tiempo_tratamiento

```

```

lds    r18,tiempo_tratamiento
lds    r19,tiempo_tratamiento+1
SEGUIR:
inc    cruces
cpi    cruces,159
brlo   sigue_refrescamiento
rjmp   lectura_data_sensor

sigue_refrescamiento:
cbi    portd,0
rjmp   end

lectura_data_sensor:
cpi    cruces,160
brlo   Comparación_temperatura
rjmp   ESTADO_ON_OFF

Comparación_temperatura:
ldi    bandera_zcd,1
rjmp   end_2

ESTADO_ON_OFF:
cpi    cruces,255
breq   comparo_luego_borro
rjmp   end

comparo_luego_borro:
lds    r16,bandera_ON
cpi    r16, 0
breq   duplico_ON
cpi    r16, 1
breq   termino_ON
rjmp   borro

duplico_ON:
ldi    cruces,160
ldi    r16,1
sts    bandera_ON,r16
rjmp   end

termino_ON:
clr    r16
sts    bandera_ON,r16

borro:
cbi    portd,0
clr    cruces

end:
clr    bandera_zcd

end_2:
clr    r18
out    TCNT1H,r18
out    TCNT1L,r18
POP    R16
    POP    R17
    POP    R19
    POP    R18
    OUT    sreg,R18
    POP    R18
    RETI
    ,*****
    ,
    
```

