Anexo A

Análisis de costos

ITEM	INSUMO	CANT	SUB TOTAL (S/.)
1	Microcontrolador PIC 16F877	1	28.00
2	Pantalla LCD 20 x 4	1	50.00
3	Teclado Matricial 4x4	1	20.00
4	Driver IR2184	2	56.00
5	Mosfet IRFZ44	4	16.00
6	Optocoplador 4N25	1	1.00
7	Bobina 180uH	4	18.00
9	Amplificador de instrumentación INA125P	1	48.00
10	Red Termilinear 44201	1	140.00
11	Celda Peltier VT-199-1.4-0.8	2	250.00
12	Ventilador 12V	1	5.00
13	Disipador de calor	1	7.50
14	Componentes diversos	1	80.00
15	Pasta conductora	1	3.50
	COSTO TOTAL		723.00

Anexo B

- Hoja técnica Celda Peltier VT-199-1.4-0.8

Anexo C

Análisis de Corriente

Etapa de control y Etapa de Interfaz de Usuario	Cantidad	Voltaje de Alimentación	Corriente Consumida
PIC16F877,LCD Y TECLADO	1	5V	32.3 mA
Etapa de Sensado			
Tarjeta	1	12V Y -12V	17.7 mA
Etapa de Potencia			
Tarjeta	2	24V	15000 mA
Motor ventilador	1	12V	110 mA
		Total	15.16 A

Anexo D

- Hoja técnica fuente TDK-Lambda SWS600 - 24

Anexo E

- Hoja técnica fuente TDK-Lambda SCT40g

Anexo F

Se realizan pruebas para ver si nuestro sistema de adquisición está midiendo correctamente. Ver figura G.1

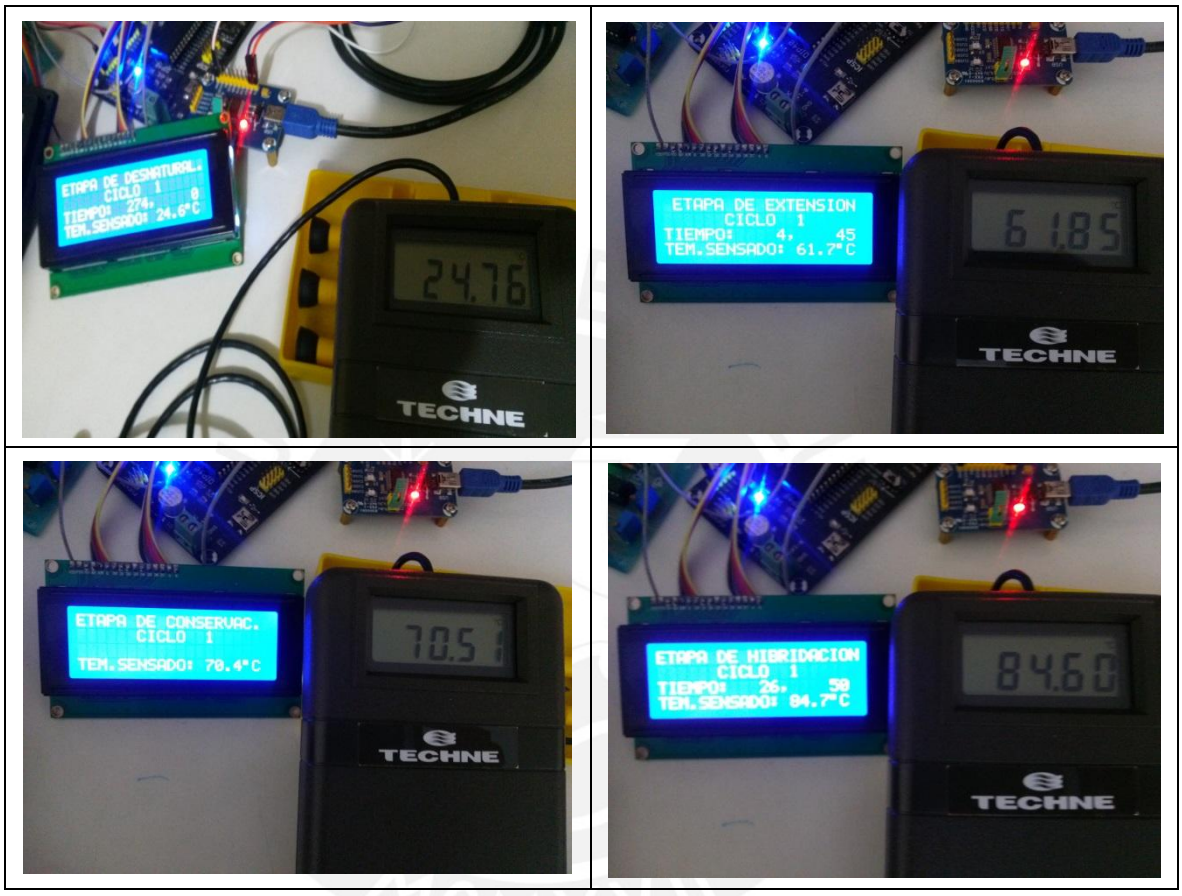
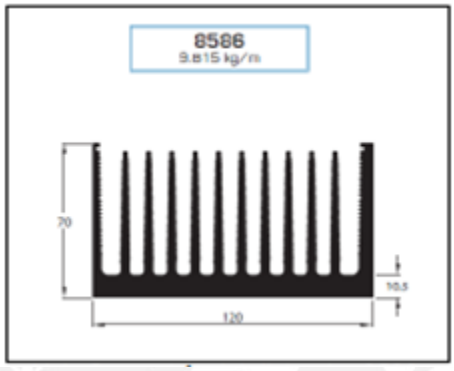


Figura G.1 Contraste con el Termómetro digital y el módulo desarrollado en este trabajo.

Anexo H

Según el trabajo realizado por Piminchumo, O. [14], el disipador y ventilador del prototipo de Termociclador debería contar con las siguientes características.

El disipador seleccionado será de la empresa *Pada Enginering*, tipo extruded, código 8586, cuyas características se muestra en la figura H.1 y tabla H.1.



H.1 Geometría del disipador 8586.

Tabla H.1 Datos del disipador seleccionado.

Código	Peso (kg/m)	Rth (°C/W)	Material	Conductividad (W/m.k)
8586	9.815	1.02	Aleación de aluminio AW-6060	200 - 240

Elaboración: Piminchumo. O [14]

El ventilador elegido será el modelo F-12038X24-2 de la empresa *Supercool* que se muestra en la figura H.2 y que presenta las siguientes características en la tabla H.2.

Tabla H.2. Datos del ventilador seleccionado.

Modelo	Dimensiones (mm)	Flujo (m3/h)	Corriente (A)	Voltaje DC(V)
F-12038X24-2	120x120x38	323	0.6	24

Elaboración: Piminchumo. O [14]

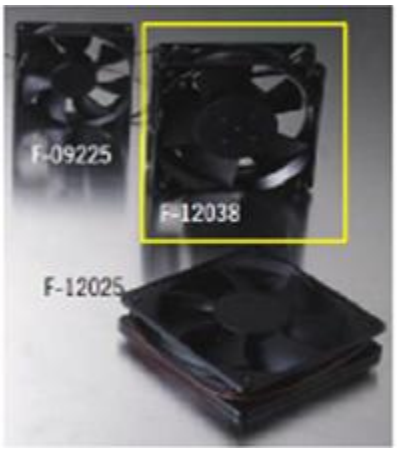


Figura H.2 Ventilador seleccionado

Cabe resaltar que nuestra celda Peltier hace uso exclusivo de una sola fuente de alimentación que cumpla con las siguientes características en la Tabla H.3.

Tabla H.3 Requerimientos mínimos de una fuente.

Imax (amps)	Vmax (volts)
11.3	24.6

Elaboración: Piminchumo, O [14]

La fuente de alimentación a usar para la celda Peltier y el sistema de adquisición de datos se encuentran en la figura H.4.

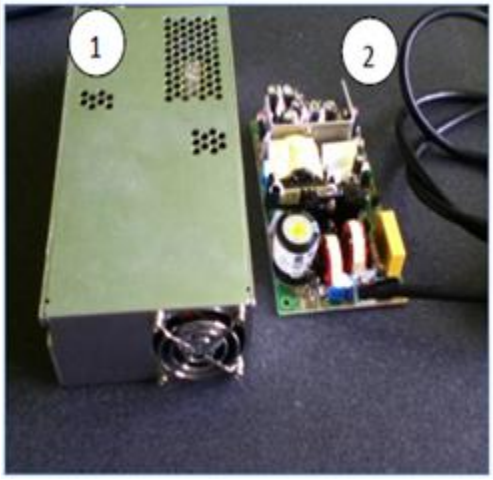
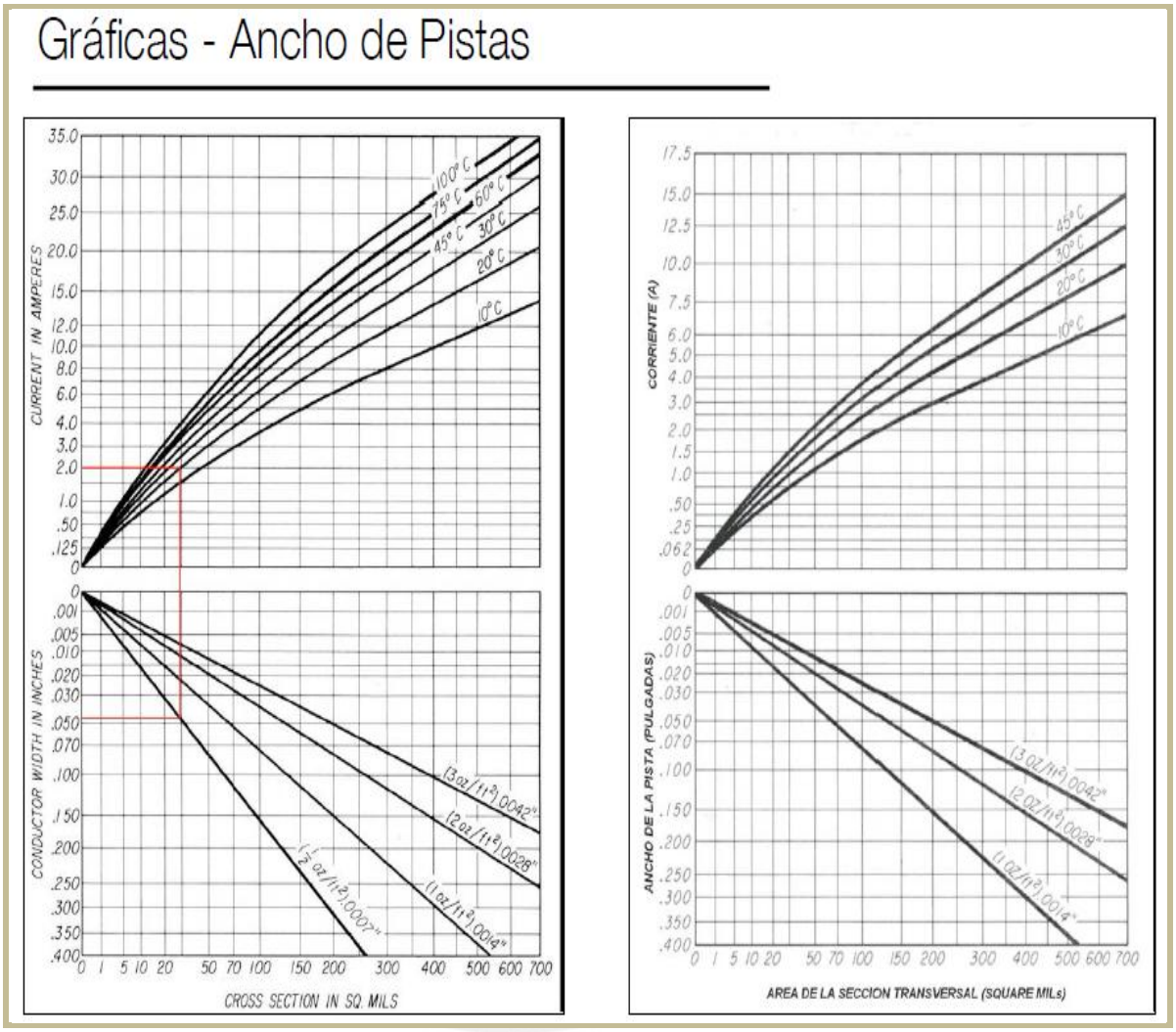


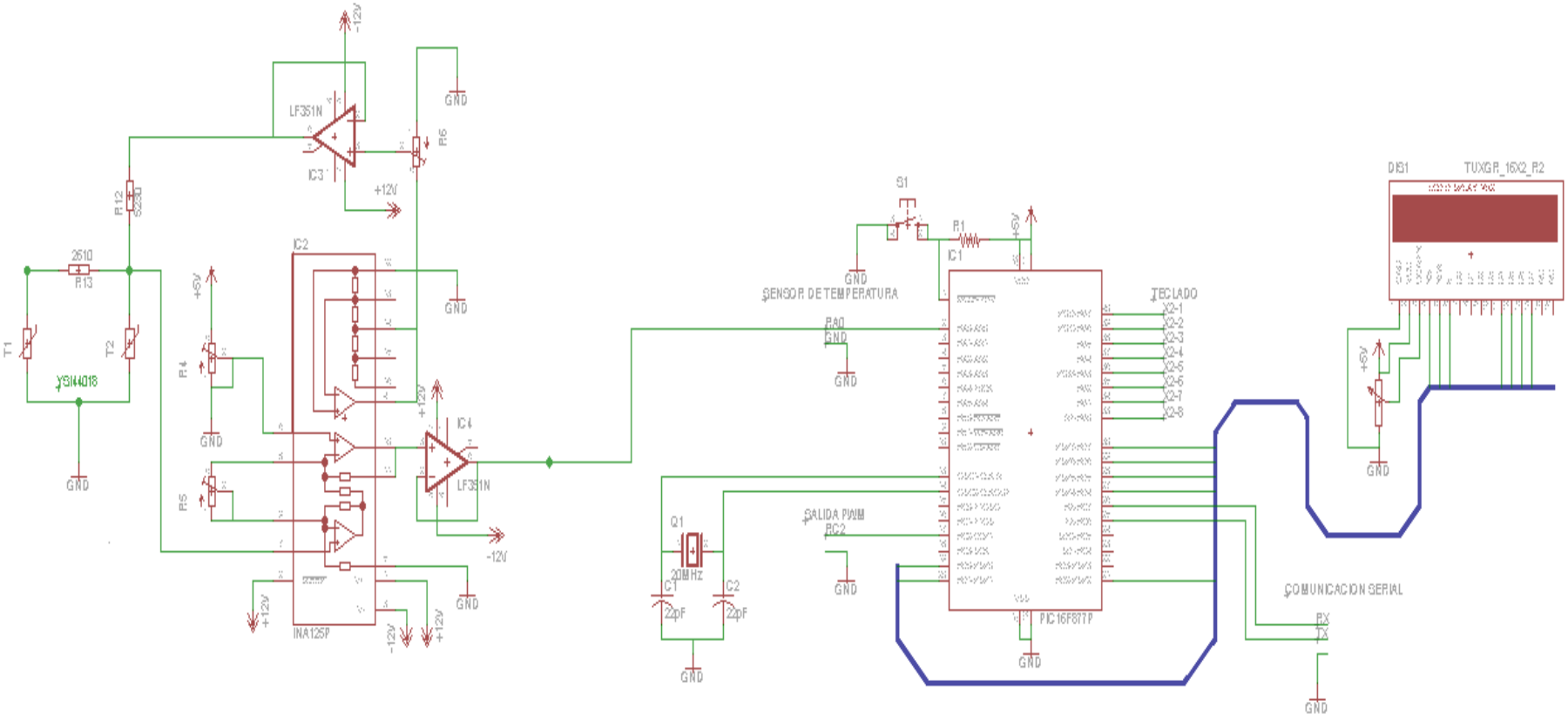
Figura H.4: 1) Fuente TDK-Lambda SWS600-24 de 24v a 22A
2) Fuente TDK-Lambda SCT40g +5v, -12v y +12v a 6A

Anexo I



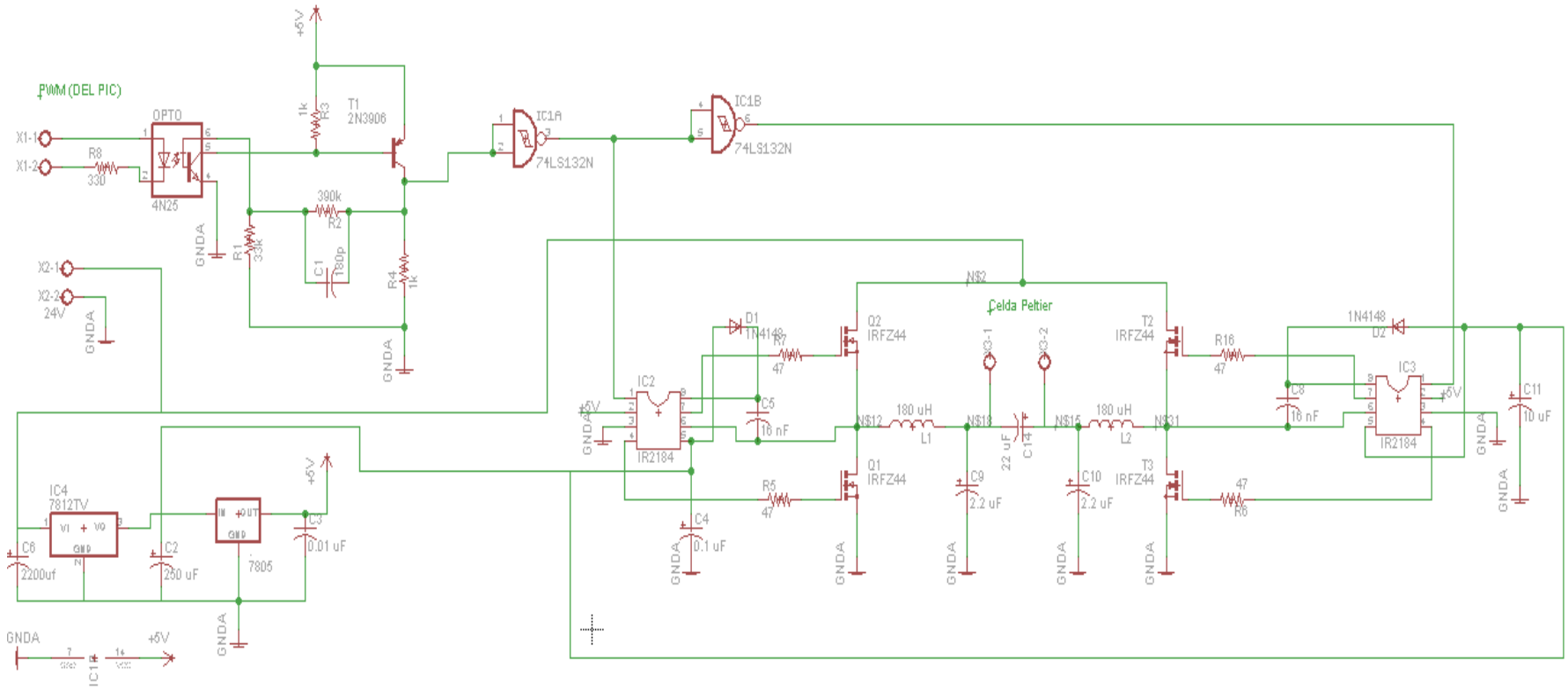
Anexo j

Diagrama esquemático de la Etapa de Sensado de Temperatura y la etapa de Control.



Anexo k

Diagrama esquemático de la Etapa de Potencia



Anexo L

Programa principal.

```
#include "C:\Users\Jesus Macassi\Desktop\Programa PIC Jesus\main.h"


#use fast_io(A)
#use fast_io(B)
#use fast_io(C)
#use fast_io(D)
#use fast_io(E)


#include <stdio.h>
#include <string.h>
#include "lcd420_portd.c"
#include "kbd2.c"

#define ms100 3036
#define Tmax 0
```

```
#define Tmin 1
#define Test 2
#define timemax 3
#define timemin 4
#define timeest 5
#define ciclos 6


#define duty_cycle_elevacion 98
#define duty_cycle_disminucion 25
#define duty_cycle_mantenimiento 50
#define duty_cycle_bajo_aumento 85

int1 relacion;
float t_prueba;
float t_intermedia;

//Variables Globales

int correcto = 0;
char char_in;
```

```
int16 variables[7];
float temperatura;
int16 i_temper;
int16 d_temper;
int cont_timer1=0;
int16 cont_seg_por10=0;
int ciclo_actual=0;

int16 tiempo_pasado=0;

int16 lectura2;



int16 tiempo_serial=0;

#separate
void setup(void);
#separate
void bienvenida(void);
#separate
void setupVariables(void);
```

```
#separate
void cicloTmax(void);
#separate
void cicloTmin(void);
#separate
void cicloTest(void);
#separate
void cicloTconservacion(void);
#separate
void ingreso_tmax(void);
#separate
void ingreso_timemax(void);
#separate
void ingreso_tmin(void);
#separate
void ingreso_timemin(void);
#separate
void ingreso_test(void);
#separate
void ingreso_timeest(void);
#separate
```

```
void ingreso_tmax(void);
#separate
void ingreso_ciclos(void);
#separate
void ingreso_variable(int,int16,int16);
#separate
void etapa_comun_ciclo(int16,int16,int1,int16,int);

void iniciar_temporizacion(void);

#separate
int16 tiempo_seg(void);
#separate
void config_pwm(int16 unduty);
#separate
void mostrar_ciclo(void);
#separate
void mostrar_segundos(void);
#separate
void mostrar_temperatura(void);
#separate
```

```
void leer_temperatura(void);



#int_timer1
void timer_1(void)
{
  set_timer1(ms100);
  if(cont_timer1==5)
  {
      cont_timer1=0;
      leer_temperatura();
      printf("T:%3Lu.%1Lu\xF8C,%Lu\r\n",i_temper,d_temper,lectura2);



  }
  else
  {
     cont_timer1++;
```

```
  }
  cont_seg_por10++;
}

void main()
{
   setup();
   //LOOP
   while(1)
   {
      bienvenida();
      setupVariables();
      do
      {
         ciclo_actual = ciclo_actual+1;
         cicloTmax();
         cicloTmin();
         cicloTest();
      }
      while(variables[ciclos]!=ciclo_actual);
      lcd_gotoxy(1,3);
```

```
      printf(lcd_putc,"                    ");
      cicloTconservacion();
      disable_interrupts(int_timer1);
      disable_interrupts(global);
   }
}



#separate
void setup(void)
{


   setup_psp(PSP_DISABLED);
   setup_spi(SPI_SS_DISABLED);
   setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
   setup_comparator(NC_NC_NC_NC);
   setup_vref(FALSE);
```

```
   set_tris_D(0x00);
   output_D(0x00);
   set_tris_C(0x80);

   setup_adc_ports(AN0);
   setup_adc(ADC_CLOCK_DIV_8);

   setup_ccp1(CCP_PWM);
   setup_timer_2(T2_DIV_BY_16,125,1);
   set_pwm1_duty(63);

   setup_timer_1(T1_INTERNAL|T1_DIV_BY_8);

   lcd_init();
}

#separate
void bienvenida(void)
{
   lcd_gotoxy(1,1);
   printf(lcd_putc,"  PCR:SISTEMA PARA  ");
```

```
   lcd_gotoxy(1,2);
   printf(lcd_putc," REPLICACION DE ADN ");
   lcd_gotoxy(1,3);
   printf(lcd_putc,"  Presione <ENTER>  ");
   lcd_gotoxy(1,4);
   printf(lcd_putc,"    para iniciar    ");
   while(kbd_getc()!='E')
   {

   }
}

#separate
void setupVariables(void)
{
   int confirmacion = 0;
   do
   {

      lcd_gotoxy(1,1);
      printf(lcd_putc,"                    ");
```

```
lcd_gotoxy(1,2);
printf(lcd_putc,"       INGRESE      ");
lcd_gotoxy(1,3);
printf(lcd_putc,"       VALORES      ");
lcd_gotoxy(1,4);
printf(lcd_putc,"                    ");
delay_ms(1500);
lcd_gotoxy(1,2);
printf(lcd_putc,"                    ");
lcd_gotoxy(1,3);
printf(lcd_putc,"    CLEAR:BORRAR    ");
lcd_gotoxy(1,4);
printf(lcd_putc,"   ENTER:INGRESAR   ");
ingreso_tmax();
ingreso_timemax();
ingreso_tmin();
ingreso_timemin();
ingreso_test();
ingreso_timeest();
ingreso_ciclos();
```

```
lcd_gotoxy(1,1);
printf(lcd_putc,"CONF.FINAL:Ciclos:%2Lu",variables[ciclos]);
lcd_gotoxy(1,2);
printf(lcd_putc,"Temper.: %3Lu,%3Lu,%3Lu",variables[Tmax],variables[Tmin],variables[Test]);
lcd_gotoxy(1,3);
printf(lcd_putc,"Tiempos: %3Lu,%3Lu,%3Lu",variables[timemax],variables[timemin],variables[timeest]);
lcd_gotoxy(1,4);
printf(lcd_putc,"C:BORRAR E:CONFIRMAR");
do
{
  char_in=kbd_getc();
}
while((char_in!='E')&&(char_in!='C'));
if(char_in=='E')
{
  confirmacion=1;
}
else
{
  confirmacion=0;
}
```

```
   }
   while(confirmacion==0);


   set_timer1(ms100);
   enable_interrupts(int_timer1);
   enable_interrupts(global);
}

#separate
void cicloTmax(void)
{
   leer_temperatura();
   lcd_gotoxy(1,1);
   printf(lcd_putc,"ETAPA DE DESNATURAL.");
   etapa_comun_ciclo(variables[Tmax],variables[timemax],0,duty_cycle_elevacion,1);
}

#separate
void cicloTmin(void)
{
```

```
   lcd_gotoxy(1,1);
   printf(lcd_putc,"ETAPA DE HIBRIDACION");
   etapa_comun_ciclo(variables[Tmin],variables[timemin],1,duty_cycle_disminucion,2);
}

#separate
void cicloTest(void)
{
   lcd_gotoxy(1,1);
   printf(lcd_putc," ETAPA DE EXTENSION ");
   etapa_comun_ciclo(variables[Test],variables[timeest],0,duty_cycle_elevacion,3);
}

#separate
void cicloTconservacion(void)
{
   lcd_gotoxy(1,1);
   printf(lcd_putc,"ETAPA DE CONSERVAC. ");
   config_pwm((duty_cycle_mantenimiento));
   delay_ms(500);
   config_pwm(duty_cycle_disminucion);
```

```
   do
   {
      mostrar_temperatura();
   }
   while(temperatura>4);
   config_pwm(duty_cycle_mantenimiento);
}


////////////////////////////////////////////////////////////////////////////////

#separate
void ingreso_tmax()
{
      lcd_gotoxy(1,1);
      printf(lcd_putc,"Temp.Desnatural.(\xDFC)");
      ingreso_variable(Tmax,0,100);
}

#separate
void ingreso_timemax()
```

```
{
    lcd_gotoxy(1,1);
    printf(lcd_putc,"Tiemp.Desnatural.(s)");
    ingreso_variable(timemax,0,1080);
}

#separate
void  ingreso_tmin()
{
    lcd_gotoxy(1,1);
    printf(lcd_putc,"Temp.Hibridac.  (\xDFC)");
    ingreso_variable(Tmin,0,100);
}

#separate
void ingreso_timemin()
{
    lcd_gotoxy(1,1);
    printf(lcd_putc,"Tiemp.Hibridac.  (s)");
    ingreso_variable(timemin,0,1080);
}
```

```
#separate
void ingreso_test()
{
      lcd_gotoxy(1,1);
      printf(lcd_putc,"Temp.Extension  (\xDFC)");
      ingreso_variable(Test,0,100);
}

#separate
void ingreso_timeest()
{
   //timemin:hibridacion
      lcd_gotoxy(1,1);
      printf(lcd_putc,"Tiemp.Extension  (s)");
      ingreso_variable(timeest,0,1080);
}

#separate
void ingreso_ciclos()
{
```

```
   //ciclos
      lcd_gotoxy(1,1);
      printf(lcd_putc," Numero de Ciclos:  ");
      ingreso_variable(ciclos,0,20);
}

////////////////////////////////////////////////////////////////////////////////
#separate
void ingreso_variable(int unavariable,int16 valormin,int16 valormax)
{
      correcto = 0;
      variables[unavariable] = 0;
      while(correcto==0)
      {
            do
            {
               char_in = kbd_getc();
            }
            while(char_in==0);
            delay_ms(100);
            if((char_in>('0'-1))&&(char_in<('9'+1)))
```

```
{
   variables[unavariable] = variables[unavariable]*10+(char_in-'0');
   lcd_gotoxy(1,2);
   printf(lcd_putc,"        %4Lu        ",variables[unavariable]);
}
else
{
   if(char_in=='C')
   {
      correcto = 0;
      variables[unavariable] = 0;
      lcd_gotoxy(1,2);
      printf(lcd_putc,"                    ");
   }
   if(char_in=='E')
   {
      if((variables[unavariable]>valormin)&&(variables[unavariable]<valormax))
      {
         correcto = 1;


      }
```

```
            else
            {
              correcto = 0;
              variables[unavariable] = 0;
              lcd_gotoxy(1,2);
              printf(lcd_putc,"     INCORRECTO     ");
              delay_ms(2000);
              lcd_gotoxy(1,2);
              printf(lcd_putc,"  INGRESE DE NUEVO  ");
              //clearLine(2);
            }
          }
        }
    }
    lcd_gotoxy(1,2);
    printf(lcd_putc,"                    ");
}
```

```
////////////////////////////////////////////////////////////////////////////

#separate
void etapa_comun_ciclo(int16 unT,int16 untime,int1 signo,int16 un_duty,int una_etapa)
{

//!   tiempo_serial = untime;
   int16 t_transcurrido;

   lcd_gotoxy(1,2);
   printf(lcd_putc,"                    ");
   lcd_gotoxy(1,3);
   printf(lcd_putc,"TIEMPO:             ");
   lcd_gotoxy(1,4);
   printf(lcd_putc,"TEM.SENSADO:   . \xDFC ",);
   mostrar_ciclo();
   if(signo==1)
   {
      config_pwm(duty_cycle_mantenimiento);
   }
   else
```

```
{
   config_pwm((duty_cycle_mantenimiento+un_duty)/2);
}
delay_ms(500);
config_pwm(un_duty);
relacion=1;
lcd_gotoxy(13,3);
printf(lcd_putc,'', %5Lu",tiempo_pasado);
iniciar_temporizacion();
do
{
   mostrar_temperatura();
   mostrar_segundos();
   if(signo)
   {
      if(temperatura>unT)
      {
         relacion=1;
      }
      else
      {
```

```
      if(temperatura>unT)
      {
         relacion=1;
      }
      else
      {
      relacion=0;
      }
   }
}
else
{
   if(temperatura<unT)
   {
      relacion=1;
   }
   else
   {
      if(temperatura<unT)
      {
         relacion=1;
```

```
        }
        else
        {
        relacion=0;
        }
      }
    }
}
while(relacion);
tiempo_pasado=tiempo_seg();
lcd_gotoxy(1,1);
if(una_etapa==1)
{
   printf(lcd_putc,"DESNAT. ESTABLECIDA ");
}
else
{
   if(una_etapa==2)
   {
      printf(lcd_putc,"HIBRID. ESTABLECIDA ");
   }
```

```
  else
  {
    printf(lcd_putc,"EXTENS. ESTABLECIDA ");
  }
}
lcd_gotoxy(13,3);
printf(lcd_putc,", %5Lu",tiempo_pasado);


config_pwm((duty_cycle_mantenimiento+un_duty)/2);
delay_ms(500);
config_pwm(duty_cycle_mantenimiento);
iniciar_temporizacion();
do
{
  if(temperatura<((float)unT)-0.5)
  {
    config_pwm(duty_cycle_bajo_aumento);
  }
  else
  {
```

```
        if(temperatura>((float)unT)+0.5)
        {
           config_pwm(duty_cycle_mantenimiento);
        }
      }

      mostrar_temperatura();
      mostrar_segundos();
      t_transcurrido = cont_seg_por10;
   }
   while(t_transcurrido<(untime*10));
   tiempo_pasado = t_transcurrido/10;
}


////////////////////////////////////////////////////////////////////////////////


void iniciar_temporizacion(void)
{
   cont_seg_por10=0;
```

```
}

#separate
int16 tiempo_seg(void)
{
   int16 untiemposeg;
   untiemposeg = (cont_seg_por10/10);
   return untiemposeg;
}

#separate
void config_pwm(int16 unduty)
{
   int16 control;
   control = unduty*125;
   control = (int16)(control/100);
   set_pwm1_duty((int)control);
}

#separate
void mostrar_ciclo(void)
```

```
{
   lcd_gotoxy(1,2);
   printf(lcd_putc,"      CICLO %2u      ",ciclo_actual);
}

#separate
void mostrar_segundos(void)
{
   lcd_gotoxy(8,3);
   printf(lcd_putc,"%5Lu",tiempo_seg());
}

#separate
void mostrar_temperatura(void)
{
   lcd_gotoxy(13,4);
   printf(lcd_putc,"%3Lu",i_temper);
   lcd_gotoxy(17,4);
   printf(lcd_putc,"%1Lu",d_temper);
}
```

```
#separate
void leer_temperatura(void)
{
   int16 lectura;
   set_adc_channel(0);
   delay_us(10);
   lectura = read_adc();
   lectura2 = lectura;
   delay_us(10);
   t_intermedia = ((float)lectura)/1023;
   t_intermedia = t_intermedia*100;
   temperatura=t_intermedia;
   i_temper=(int16)temperatura;
   d_temper=((int16)(temperatura*10))%10;
}
```