

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

DESARROLLO DE UN DISPOSITIVO PARA TERAPIA TÉRMICA
SUPERFICIAL CON CONTROL DIGITAL, BASADO EN EL
PRINCIPIO PELTIER.

Anexos.

Alfredo Medina Pérez

ASESOR: Ing. Eduardo Toledo Ponce

Lima, Agosto del 2013

INDICE DE ANEXOS

Índice de Anexos.....	2
Índice de figuras.....	2
Índice de tablas.....	2
Anexo 1. Programa Principal del sistema.....	3
Anexo 2. Cálculo de disipadores utilizados.....	16
Anexo 3. Diseño de la carcasa y el equipo terminado.....	24
Anexo 4. Curvas de Schade.....	28

INDICE DE FIGURAS

Figura 2.1. Equivalente eléctrico del sistema térmico.....	17
Figura 2.2. Tipos de disipadores (disipadores.com).....	18
Figura 2.3. Especificaciones técnicas del bloque de disipación.....	23
Figura 2.4. Bloque de disipación.....	23
Figura 3.1. Dimensiones de la cara superior.....	25
Figura 3.2. Dimensiones de la cara lateral.....	25
Figura 3.3. Dimensiones de la cara frontal.....	25
Figura 3.4. Dimensiones de la cara posterior.....	26
Figura 3.5. Dimensiones de la cara inferior.....	26
Figura 3.6. Vista frontal superior.....	27
Figura 3.7. Vista lateral del equipo.....	27
Figura 3.8. Vista posterior.....	27
Figura 3.9. Vista interior.....	27
Figura 3.10. Equipo conectado al aplicador y en funcionamiento.....	27
Figura 4.1. Curvas de Schade para $R_s/R_L\%$ VS $\omega C R_L$ VS V_{RMS}/V_{MED} [32].....	28
Figura 4.2. Curvas de Schade para V_{MED}/V_P VS $\omega C R_L$ VS $R_s/R_L\%$ [32].....	29
Figura 4.3. Curvas de Schade para I_{RMS}/I_{MED} VS $n\omega C R_L$ VS R_s/nR_L [32].....	30

INDICE DE TABLAS

Tabla 2.1. Tabla de resistencia térmica contenedor – disipador.....	17
---	----

ANEXO 1. PROGRAMA PRINCIPAL DEL SISTEMA

A continuación se muestra el código del programa realizado en VMLAB

```
.include "C:\VMLAB\include\m8def.inc"

.list
.def dato = R22
.def instruccion = R23
.def temp_1 = R20
.def temp_2 = R21
.def TSensaH = R17
.def TSensal = R18
.def TprogrH = R16
.def TprogrL = R19
.equ temp_maxima = 50
.equ temp_minima = 2

.dseg
.org $60

Temp_programada: .BYTE 1 ;temperatura programada de 8 bits
Temp_sensada: .BYTE 1 ;temperatura sensada de 8 bits
cont_veces: .BYTE 1 ;cuenta las veces que se ha ingresado en la subrutina de interrupción externa

ID: .BYTE 1 ;valor que me indica si se está incrementando o decrementando la temperatura
cont: .BYTE 1 ;indica si ya transcurrió más de segundo y medio luego de q comenzaste a apretar el boton
.cseg
.org $0
RESET:
rjmp Programa_Principal
rjmp Int_ext_programar_start_stop
reti ; Addr $02
reti ; Addr $03
reti ; Addr $04
reti ; Addr $05
rjmp Interruccion_500ms ;subrutina de servicio de interrupcion por comparacion exitosa de ocr1A
reti ; Addr $07
reti ; Addr $08
reti ; Addr $09
reti ; Addr $0A
reti ; Addr $0B
reti ; Addr $0C
reti ; Addr $0D
reti ; Addr $0E
reti ; Addr $0F
reti ; Addr $10
reti
reti

Mensaje1:
.db "Terapia Termica"
Mensaje2:
.db "superficial"
Mensaje3:
.db "Temperatura:"
Mensaje4:
.db "programando:"

Mensaje5:
.db "comenzando"
Mensaje6:
.db "tratamiento.."
Mensaje7:
```

```
.db "crioterapia:"
Mensaje8:
.db "termoterapia:"
Mensaje9:
.db "sesion"
Mensaje10:
.db "concluida"
```

```
TablaValoresDisplay:
```

```
.db $00,$01
.db $02,$03
.db $04,$05
.db $06,$07
.db $08,$09
.db $10,$11
.db $12,$13
.db $14,$15
.db $16,$17
.db $18,$19
.db $20,$21
.db $22,$23
.db $24,$25
.db $26,$27
.db $28,$29
.db $30,$31
.db $32,$33
.db $34,$35
.db $36,$37
.db $38,$39
.db $40,$41
.db $42,$43
.db $44,$45
.db $46,$47
.db $48,$49
.db $50,$51
```

```
;-----
; El Programa empieza después de un RESTART
;-----
```

```
Programa_Principal
```

```
cli
ldi R16,HIGH(RAMEND)      ;configuro la pila
out SPH, R16
ldi R16, LOW(RAMEND)
out SPL, R16
clr R16
sts cont, R16             ;inicializo variables
sts cont_veces, R16
sts Temp_sensada, R16
sts Temp_programada, R16
sts ID, R16
Rcall Config_puertos
Rcall Config_ADC
Rcall Config_Interr_externa ; configuro interrupción externa PD2
Rcall configura_LCD
;-----
; muestra el mensaje inicial "Terapia térmica superficial"
; por cuatro segundos
;-----
ldi ZH,high(Mensaje1*2)   ;Z apunta al inicio del mensaje "Terapia térmica"
ldi ZL,low(Mensaje1*2)
clr R16                   ;Contador de caracteres a visualizar inicialmente en "0"
```

```

leer_otroc:
    rcall CheckBF
    lpm dato,Z+           ;Lee y muestra caracter en LCD
    rcall WriteDR
    inc R16
    cpi R16,15           ; Muestra 15 caracteres
    brne leer_otroc
Rcall situa_cursor_2da
    ldi ZH,high(Mensaje2*2) ;Z apunta al inicio del mensaje "superficial"
    ldi ZL,low(Mensaje2*2)
    clr R16
leer_otroc2:
    rcall CheckBF
    lpm dato,Z+           ;Lee y muestra caracter en LCD
    rcall WriteDR
    inc R16
    cpi R16,11           ; Muestra 11 caracteres
    brne leer_otroc2
    Rcall Retardo_1s      ; Muestra el mensaje por 4 segundos
    Rcall Retardo_1s
    Rcall Retardo_1s
    Rcall Retardo_1s
Rcall borra_display
    ldi ZH,high(Mensaje3*2) ;Z apunta al inicio del mensaje "Temperatura"
    ldi ZL,low(Mensaje3*2)
    Clr R16               ; Contador de caracteres a visualizar
leer_otroc3:
    rcall CheckBF
    lpm dato,Z+           ;Lee y muestra caracter en LCD
    rcall WriteDR
    inc R16
    cpi R16,12           ; Muestra 12 caracteres
    brne leer_otroc3
    Rcall situa_cursor_2da
    Sei                   ; Activo interrupciones

Lazo_principal:
    Rcall obtiene_promedio
    sts Temp_programada, R17 ; guardo en Temperatura programada el valor de la temperatura sensada
    sts Temp_sensada, R17   ; guardo en la posicion de memoria TEMP sensada
    Rcall Muestro_LCD       ; muestro en el display LCD
    Rcall Retardo200ms      ; espero para actualizar
    Rcall Retardo200ms
    Rcall Retardo200ms
    Rjmp Lazo_principal

;-----
; obtiene promedio:
; subrutina que lee 8 veces sensor de temperatura con una diferencia de 50 us
; y obtiene promedio de todas las lecturas
;-----
obtiene_promedio:
    push r16               ; salvo el valor de R16 en la pila
    clr R16                ; cargo el valor de "0" en los registros R16, R0, R1 y R2
    clr R0
    clr R1
    clr R2
suma:
    clr R3
  
```

```

Rcall Leo_sensor
Add R0,R18          ; Leo el sensor y sumo al valor anterior
Adc R1,R17
Adc R2,R3
Inc R16
rcall espera_50us   ; la lectura del siguiente valor se hara cada 50 us
cpi R16,8
brne suma          ; si ya leyó 8 veces ahora obtiene el promedio
swap R2
lsl R2
mov R18,R2
andi R18,0b11100000
mov R2,R18
mov R17,R1
mov R18,R1
lsl R17
lsl R17
lsl R17
or R17,R2
swap R18
lsl R18
andi R18, 0b11000000
pop R16
ret
;-----
; espera 50 us entre lectura
;-----
espera_50us:
    push R16          ;2us
    clr R16           ;1us

lazo000:
    inc R16           ;1us
    cpi R16,14        ;1us
    brne lazo000     ;1us
    pop R16           ;2us

    ret              ;4us

;-----
; borra display:
;subrutina que borra display y situa cursor en posición 1,1
;-----

borra_display:
    ldi instruccion, 0b00000001 ; borra el display
    rcall CheckBF
    rcall WritelR
    ldi instruccion, 0b10000001 ; se situa el cursor en 1,1
    rcall CheckBF
    rcall WritelR
    ret

;-----
; situa cursor en 2da fila
;-----

```

```

situa_cursor_2da:
    ldi instrucion, 0b11000001      ;se situa el cursor en 2,2
    rcall CheckBF
    rcall WriteR
    ret

;-----
; Interruccion externa para programar, iniciar tratamiento
; y parar el mismo
;-----

Int_ext_programar_start_stop:
    ldi R16,HIGH(RAMEND)          ;devuelvo el valor de la pila a su posición original
    out SPH, R16
    ldi R16, LOW(RAMEND)
    out SPL, R16
    lds R16, cont_veces           ; "Cont_veces" me indica cuantas veces se entró a esta interrupción
    inc R16                       ; de este valor dependerá el flujo del programa.
    cpi R16,1                     ; si es que pulse una sola vez se va a programar la temperatura
    breq Programo_temperatura
    cpi R16,2                     ; si es que se presiona por segunda vez entonces comienza el control de temperatura
    breq Com_control
    rjmp Detiene_control          ; si es que presioné más de 2 veces se detiene el control de temperatura
Com_control:
    rjmp Comienza_control_temperatura

;-----
; en la siguiente sección del programa se procede a programar la
; temperatura de operación.
;-----

Programo_temperatura:
    sts cont_veces, R16           ; actualizo el valor de la variable contador de veces
    rcall Configuracion_Timer1    ; configura TIMER
    rcall borra_display
    ldi ZH,high(Mensaje4*2)      ; Z apunta al inicio del mensaje "programando:"
    ldi ZL,low(Mensaje4*2)
    clr R16                       ; Contador de caracteres a visualizar
leer_otroc4:
    rcall CheckBF
    lpm dato,Z+                  ; lee y muestra caracter en LCD
    rcall WriteDR
    inc R16
    cpi R16,12                   ; muestra 12 caracteres
    brne leer_otroc4
    rcall situa_cursor_2da
    rcall muestro_LCD
    sei                           ; activo interrupciones
analizando_entradas:
    in R16, PIND                  ; PD1 es incremento
    andi R16,$02                  ; enmascaro el PIN PD1 (INCREMENTO)
    cpi R16,0                     ; está en 0? (esta pulsado?)
    breq Incrementa_temperatura   ; si está pulsado incremento la temperatura
    in R16, PIND
    andi R16,$01                  ; enmascaro el PIN PD0 (DECREMENTO)
    cpi R16, 0                   ; está en 0? (esta pulsado?)
    breq Decrementa_temperatura   ; si está pulsado decremento la temperatura
    rjmp analizando_entradas      ; si no entonces sigo analizando las entradas

Incrementa_temperatura:
    lds R16, Temp_programada      ; cargo en R16 el valor almacenado en TEMP PROGRAMADA
  
```

```

Cpi R16, Temp_maxima           ; comparo con el valor máximo
Breq analizando_entradas       ; si es igual a la temperatura máxima entonces vuelvo al lazo de análisis de entradas
ldi R17,1                       ; cambio el valor de ID (informa si está aumentando o decrementando)
sts ID, R17
Inc R16                          ; incremento la temperatura
Sts Temp_programada, R16        ; guardo el valor de la temperatura modificada en TEMP PROGRAMADA
Rcall Muestro_LCD              ; muestro la temperatura incrementada en los displays
ldi R16,(1<<OCIE1A)            ; habilito interrupción por comparación exitosa.
out TIMSK, R16

Dejo_de_pulsar_inc:
In R16, PIND                    ; enmascaro PD1
Andi R16, $02                  ; es 1? (dejo de pulsar?)
Cpi R16, $02
Brne Dejo_de_pulsar_inc        ; si no entonces vuelvo a analizar
ldi R16,(0<<OCIE1A)            ; deshabilito interrupción por comparación exitosa.
out TIMSK, R16
clr R16
sts cont, R16
Rjmp analizando_entradas

Decrementa_temperatura:
Lds R16, Temp_programada       ; cargo en R16 el valor almacenado en TEMP PROGRAMADA
Cpi R16, Temp_minima           ; comparo con el valor mínimo
Breq analizando_entradas       ; si es igual a la temperatura mínima entonces vuelvo al lazo de analisis de entradas
clr R17                          ; cambio el valor de ID a cero
sts ID, R17
Dec R16                          ; decremento temperatura
sts Temp_programada, R16        ; actualizo valor
Rcall Muestro_LCD              ; muestro la temperatura decrementada en el DISPLAY
ldi R16,(1<<OCIE1A)            ; habilito interrupción por comparación exitosa.
out TIMSK, R16

Dejo_de_pulsar_dec:
In R16, PIND
Andi R16, $01                  ; enmascaro PDO
Cpi R16, $01                    ; dejo de pulsar?
Brne Dejo_de_pulsar_dec        ; si no entonces vuelvo a analizar
ldi R16,(0<<OCIE1A)            ; deshabilito interrupción por comparación exitosa.
out TIMSK, R16
clr R16
sts cont, R16
Rjmp analizando_entradas

;-----
; Interrupción 500ms
; Interrupción a la que se ingresa cada 500 ms una vez activada
;-----

Interrupcion_500ms:
in R17, SREG                    ; salvo el registro de estados en la pila
push R17
lds R17, Cont
inc R17
sts cont, R17
cpi R17, 3
brsh prosigue
pop R17
out SREG, R17
reti
prosigue:
pop R16
pop R16                          ; puntero de pila apunta a una dirección diferente
pop R16
ldi R16,(0<<OCIE1A)            ; deshabilito interrupción por comparación exitosa.
out TIMSK, R16

```

```

sei                                     ; activo nuevamente interrupciones
lds R16, ID
cpi R16, 1
brne Decrementa_temperatura
rjmp Incrementa_temperatura

;-----
; Subrutina que comienza el control de temperatura
;-----

Comienza_control_temperatura:
sts Cont_veces, R16                   ; actualizo Cont Veces
Rcall borra_display
ldi ZH,high(Mensaje5*2)              ; Z apunta al inicio del mensaje "comenzando"
ldi ZL,low(Mensaje5*2)
clr R16                               ; Contador de caracteres a visualizar
leer_otroc5:
rcall CheckBF
lpm dato,Z+                          ; Lee y muestra caracter en LCD
rcall WriteDR
inc R16
cpi R16,10                            ; Muestra 10 caracteres
brne leer_otroc5
Rcall situa_cursor_2da
ldi ZH,high(Mensaje6*2)              ; Z apunta al inicio del mensaje "tratamiento..."
ldi ZL,low(Mensaje6*2)
clr R16                               ; Contador de caracteres a visualizar
leer_otroc6:
rcall CheckBF
lpm dato,Z+                          ; Lee y muestra caracter en LCD
rcall WriteDR
inc R16
cpi R16,13                            ; Muestra 13 caracteres
brne leer_otroc6
Rcall Retardo_1s                     ; muestra el mensaje por 3 segundos
Rcall Retardo_1s
Rcall Retardo_1s
sei                                   ; vuelvo a habilitar las interrupciones
;-----
; analisis si es crioterapia o termoterapia
;-----
lds R16, Temp_programada              ; cargo en el registro R16 la temperatura programada anteriormente
Rcall obtiene_promedio
cp R16, R17                           ; temperatura programada es mayor que la temperatura sensada??
brlo Realmente_inv

Realmente_dir:
;-----
; imprimo "termoterapia:"
; y en la segunda linea el valor de la temperatura previamente programada
;-----
Rcall borra_display
ldi ZH,high(Mensaje8*2)               ; Z apunta al inicio del mensaje "termoterapia"
ldi ZL,low(Mensaje8*2)
clr R16                               ; Contador de caracteres a visualizar
leer_otroc8:
rcall CheckBF
lpm dato,Z+                          ; Lee y muestra caracter en LCD
rcall WriteDR
inc R16
cpi R16,13                            ; Muestra 13 caracteres
brne leer_otroc8

```

```

rcall situa_cursor_2da          ; sitúo el cursor en la segunda fila y muestro la temperatura
rcall muestro_LCD

Realmente_directa:
  Idi R19, $A0                  ; polarizo en directa la celda peltier y habilito
  out PORTD, R19
Analizo_real_directa:
  Rcall retardo20ms
  lds R16, Temp_programada
  Rcall obtiene_promedio
  clr R19
  sub R19,R18                  ;comparo la temperatura programada (2 bytes) con la real sensada (2 bytes)
  sbc R16,R17
  brsh Analizo_real_directa    ;La temperatura programada es mayor que la temperatura sensada?
  Idi R19, $80                  ; deshabilito driver
  out PORTD, R19
Analizo_real_dir:
  rcall retardo20ms
  lds R16, Temp_programada
  Rcall obtiene_promedio
  clr R19
  sub R19, R18
  sbc R16, R17
  Brsh Realmente_directa      ; entonces vuelvo a polarizar en directa
  rjmp Analizo_real_dir

Realmente_inv:
;-----
;imprimo "Crioterapia:"
;y en la segunda linea el valor de la temperatura previamente programada
;-----
  Rcall borra_display
  Idi ZH,high(Mensaje7*2)      ;Z apunta al inicio del mensaje "crioterapia"
  Idi ZL,low(Mensaje7*2)
  clr R16                      ;Contador de caracteres a visualizar
leer_otroc7:
  rcall CheckBF
  lpm dato,Z+                  ;Lee y muestra caracter en LCD
  rcall WriteDR
  inc R16
  cpi R16,12                   ; Muestra 13 caracteres
  brne leer_otroc7
  rcall situa_cursor_2da
  rcall muestro_LCD

Realmente_inversa:
  Idi R19, $60                  ; polarizo en inversa la celda peltier
  out PORTD, R19
Analizo_real_inversa:
  Rcall retardo20ms
  lds R16, Temp_programada
  Rcall obtiene_promedio
  clr R19
  sub R19, R18
  sbc R16, R17
  brlo Analizo_real_inversa   ; si la temperatura programada es mayor entonces deshabilito
  Idi R19, $40
  out PORTD, R19
Analizo_real_inv:

```

```

Rcall retardo20ms
lds R16, Temp_programada
Rcall obtiene_promedio
clr R19
sub R19, R18
sbc R16, R17                ; si la temperatura programada es mayor a la temperatura sensada
brlo Realmente_inversa     ; entonces polarizo en inversa
rjmp Analizo_real_inv

;-----
; subrutina para configurar puertos de entrada/salida
;-----

Config_puertos:
ldi R16, $FF                ;LCD
out DDRB, R16
ldi R16, $07                ;PC2,PC1,PC0 control LCD y el resto como entradas
out DDRC, R16
ldi R16, $F8                ;PD7,PD6,PD5,PD4,PD3 como salidas, además PD2,PD1,PD0 como entradas
out DDRD, R16
ret

;-----
; subrutina que muestra la temperatura en el DISPLAY
;-----

Muestro_LCD:
push R17
push R16
Rcall situa_cursor_2da
lds R17, Temp_programada   ;carga valor de la temperatura programada
ldi ZH,HIGH(TablaValoresDisplay*2) ;ubico la primera posición de la tabla
ldi ZL,LOW(TablaValoresDisplay*2)
clr R16
add ZL,R17                 ;sumo el valor de la temperatura
adc ZH,R16
lpm R16,Z                  ;almaceno la temperatura a mostrar en el registro R16
ldi R17,$30                ;para darle la forma ascii
mov dato,R16               ;copio la temperatura en el registro dato
swap dato
andi dato,$0F
add dato,R17                ;le doy la forma ascii
rcall CheckBF
rcall WriteDR
mov dato, R16
andi dato, $0F
add dato, R17
rcall CheckBF
rcall WriteDR
ldi dato, ""
rcall CheckBF
rcall WriteDR
ldi dato,'C'
rcall CheckBF
rcall WriteDR
pop R16
pop R17
ret

```

```

;-----
; subrutina que detiene control de la celda peltier
;-----

Detiene_control:
  ldi R16, 0
  out PORTD, R16
  rcall borra_display
  ldi ZH,high(Mensaje9*2)           ;Z apunta al inicio del mensaje "sesión"
  ldi ZL,low(Mensaje9*2)
  clr R16                          ;Contador de caracteres a visualizar
leer_otroc9:
  rcall CheckBF
  lpm dato,Z+                       ;Lee y muestra caracter en LCD
  rcall WriteDR
  inc R16
  cpi R16,6                         ; Muestra 6 caracteres
  brne leer_otroc9
  rcall situa_cursor_2da
  ldi ZH,high(Mensaje10*2)         ;Z apunta al inicio del mensaje "concluida"
  ldi ZL,low(Mensaje10*2)
  clr R16                          ;Contador de caracteres a visualizar
leer_otroc10:
  rcall CheckBF
  lpm dato,Z+                       ;Lee y muestra caracter en LCD
  rcall WriteDR
  inc R16
  cpi R16,9                         ; Muestra 11 caracteres
  brne leer_otroc10
  rcall Retardo_1s
  rcall Retardo_1s
  rjmp Programa_principal

;-----
;configuro el CONVERTOR ANALOGO DIGITAL:
;Registro ADCSRA:
;habilito adc(ADEN=1),selecciono prescalador = 8(ADPS2=0,ADPS1=1,ADPS0=1)
;Registro ADMUX:
;selecciono forma de conexion Voltaje Referencia interno 2.56V y Pin AREF conectado a capacitor (Refs1=1,Refs0=1)
;los 8 bits mas significativos se grabaran en el registro ADCH(ADLAR=1)
;selecciono entrada PC5- ADC5(MUX3=0,MUX2=1,MUX1=0,MUX0=0)
;-----

Config_ADC:
  ldi R16,0b10000011
  out ADCSR,R16
  ldi R16,0b11100101
  out ADMUX,R16
  ret

;-----
;Leo sensor TEMP:
;subrutina que espera la conversion de un dato,de analogico a digital y luego copia los 8 bits mas significativos
;en el registro R17
;-----

Leo_Sensor:
  sbi ADCSR,6
  Espera_conversion:
  sbic ADCSR,6

```

```

rjmp Espera_conversion
in R18,ADCL                ; y en el registro R18 el valor de los 2 bits menos significativos
andi R18,$C0              ; enmascaro solo los dos bits q me interesan
in R17,ADCH                ; guardo en el registro R17 el valor de los 8 bits mas significativo
ret

```

```

;-----
;CONFIGURACION_TIMER1:
;subrutina que configura el timer para generar un evento cada 500 ms por comparación exitosa en OCR1A
;modo CTC con OCR1A como tope
;parametros a configurar:
;circuito desconectado del pin(COM1A1=0,COM1A0=0,COM1B1=0,COM1B0=0)
;pre-escalador = 1024
;cuenta hasta 488
;-----

```

```

Configuracion_Timer1:
  Idi R16, HIGH(488)
  out OCR1AH, R16
  Idi R16, LOW(488)
  out OCR1AL, R16
  Idi R16, 0
  out TCCR1A, R16
  Idi R16,(0<<WGM13) | (1<<WGM12) | (1<<CS12) | (0<<CS11) | (1<<CS10)
  out TCCR1B, R16
ret

```

```

;-----
;CONFIGURACION INTERRUPCION EXTERNA
;configuraré PIN PD2 o INTO para interrupción por flanco de bajada
; ISCO1 = 1 y ISCO0 = 0 (falling edge)
; y la habilito INTO = 1
;-----

```

```

Config_Interr_externa:
  Idi R16, (1<<ISCO1) | (0<<ISCO0)
  out MCUCR, R16
  Idi R16, (1<<INT0)
  out GICR, R16
ret

```

```

;-----
;Subrutina que introduce un retardo de 1 segundos
;-----

```

```

Retardo_1s:
  push R16                ; 2 ciclos de reloj
  push R17                ; 2 ciclos de reloj
  push R18                ; 2 ciclos de reloj
  clr R18                 ; 1 ciclo de reloj
  clr R17                 ; 1 ciclo de reloj
  clr R16                 ; 1 ciclo de reloj
lazo_exter:
  inc R17                 ; 1 ciclo de reloj
  brne lazo_exter        ; 1 ciclo de reloj
  inc R16                 ; 1 ciclo de reloj
  brne lazo_exter        ; 1 ciclo de reloj
  inc R18                 ; 1 ciclo de reloj
  cpi R18,5               ; 1 ciclo de reloj
  brne lazo_exter        ; 1 ciclo de reloj
  pop R18                 ; 2 ciclos de reloj
  pop R17                 ; 2 ciclos de reloj
  pop R16                 ; 2 ciclos de reloj

```

```

ret

;-----
; subrutina que introduce un retardo de 20 ms
;-----
Retardo20ms:
    push R16
    push R17
    clr R16
lazo_ext:
    clr R17
lazo_int:
    inc R17
    brne lazo_int
    inc R16
    cpi R16,13
    brne lazo_ext
    pop R17
    pop R16
ret

;-----
; retardo de 200 ms
;-----
Retardo200ms:
    push R16
    push R17
    clr R16
lazo_externo:
    clr R17
lazo_interno:
    inc R17
    brne lazo_interno
    inc R16
    cpi R16,130
    brne lazo_externo
    pop R17
    pop R16
ret

;-----
; Subrutina que envía una instrucción al LCD
;-----
WriteR:
    push R16
    ldi R16,$00                ;RW=0, RS=0 y E=0
    out PORTC,R16
    ldi R16,$01                ;RW=0, RS=0 y E=1
    out PORTC,R16
    out PORTB,instruccion     ;se envía la instrucción
    ldi R16,$00                ;RW=0, RS=0 y E=0
    out PORTC,R16
    ldi R16,$04                ;RW=1, RS=0 y E=0
    out PORTC,R16
    pop R16
ret

;-----
;Subrutina que Envía un dato al LCD
;-----
WriteDR:
    push R16
    ldi R16,$02                ;RW=0, RS=1 y E=0

```

```

out PORTC,R16
ldi R16,$03                ;RW=0, RS=1 y E=1
out PORTC,R16
out PORTB,datos            ;se envía el dato
ldi R16,$02                ;RW=0, RS=1 y E=0
out PORTC,R16
ldi R16,$04                ;RW=1, RS=0 y E=0
out PORTC,R16
pop R16

ret
;-----
; Subrutina que espera BF = 0
;-----
CheckBF:
  push R16
  push R17
  ldi R16,$00                ;Bus de datos: entrada
  out DDRB,R16

LecturaBF:
  ldi R16,$04                ;RW=1, RS=0 y E=0
  out PORTC,R16
  ldi R16,$05                ;RW=1, RS=0 y E=1
  out PORTC,R16
  nop
  in R17,PINB                ;Analiza bit BF (DB7)
  ldi R16,$04                ;RW=1, RS=0 y E=0
  out PORTC,R16
  andi R17,0b10000000
  cpi R17,0
  brne LecturaBF            ;Si LCD esta ocupado => espera
  ldi R16,$FF                ;Bus de datos: salida
  out DDRB,R16
  pop R17
  pop R16

ret
;-----
; Subrutina que Configura LCD
;-----
Configura_LCD:
  push R16
  push R17
  rcall Retardo20ms
  ldi instruccion,$38        ;Configuracion: 8 bits
  rcall WriteR
  rcall Retardo20ms
  ldi instruccion,$38
  rcall WriteR
  rcall Retardo20ms
  ldi instruccion,$38
  rcall WriteR
  rcall Retardo20ms

;-----
; Funcion SET. Activa funcion: 0 0 1 DL N F x x
;
;                               DL=1 --> 8 bits; DL=0 --> 4 bits
;                               N=0 --> una linea; N=1 --> dos lineas
;                               F=0 --> 5x7 puntos; F=1 --> 5x10 puntos
;-----
  ldi instruccion,$38        ;Trabajar con datos de 8 bits, 1 linea
  rcall WriteR
  rcall CheckBF
  ldi instruccion,8          ;Display OFF

```

```

rcall WriteIR
rcall CheckBF
ldi instruccion,1          ;Clear display
rcall WriteIR
rcall CheckBF

;-----
; Función Seleccionar modo:  0 0 0 0 0 1 ID S
;
;                               ID=0 --> incrementa la dirección DDRAM
;                               ID=0 --> decrementa
;                               S=1 --> desplazamiento de toda la pantalla
;                               (con ID=1 --> desplaz. a la izquierda)
;                               (con ID=0 --> desplaz. a la derecha)
;                               S=0 --> no desplaza
;-----
ldi instruccion,6          ;Cursor, sin desplazamiento
rcall WriteIR

;-----
; Funcion ON/OFF del LCD:  0 0 0 0 1 D C B
;
;                               D=0 --> apagar la pantalla; D=1 -->encender
;                               C=0 --> desactivar cursor; D=1 -->activar
;                               B=0 --> no parpadea el caracter señalado por el cursor
;-----

rcall CheckBF
ldi instruccion,$0C        ;Enciende pantalla y oculta cursor
rcall WriteIR
rcall CheckBF
pop R17
pop R16
ret
    
```

ANEXO 2. CÁLCULO DE DISIPADORES UTILIZADOS

2.1 Cálculo del disipador utilizado en el integrado L298n.

Como se vio en la sección de “pruebas del driver”, el integrado L298n debe disipar una potencia igual a 8.24 watts. Haciendo uso de la *ecuación 2.1* obtenemos que, sin disipador el L298n llegará a una temperatura de operación igual a:

$$T_J = T_A + (Q_{JC} + Q_{CA}) \times P \dots\dots\dots \text{Ecuación 2.1.}$$

Donde:

T_J = temperatura de operación

T_A = la temperatura ambiente, en este caso la fijaremos a un valor de 25°C.

$Q_{JC} = 3^\circ\text{C/W}$ resistencia térmica entre la unión y la caja [hoja de datos L298n].

$Q_{CA} = 35^\circ\text{C/W}$ resistencia térmica entre la caja y el ambiente [hoja de datos L298n]

$$P = 8.24W$$

$$T_J = 25^{\circ}C + (3^{\circ}C/W + 35^{\circ}C/W) \times 8.24W$$

$$T_J = 338.12^{\circ}C$$

El rango de temperatura permitido de este componente es: [-25; 130 °C], según hoja de datos del integrado L298n, por lo que si no se toma en cuenta un elemento disipador que lo mantenga dentro de su rango de temperatura permitido, el L298n se degradará rápidamente y terminará malográndose.

Para calcular el disipador adecuado primero se realiza el equivalente eléctrico del sistema térmico como se observa en la *figura 2.1*.

RESISTENCIA TERMICA CONTENEDOR -DISIPADOR				
Tipo de conenedor	Contacto directo sin mica	Contacto directo y silicona	Contacto con mica	Contacto con mica silicona
TO.5	1	0,7	--	--
TO.39	1	0,7	2	1,5
TO.126	1,4	1	1,4	1,3
TO.220	0,8	0,5	1,4	1,2
TO.202	0,8	0,5	1,4	1,2
TO.152	0,8	0,5	1,2	0,9
TO.90	0,5	0,3	1	0,7
TO.3P.	0,4	0,2	2,1	1,5
TO.59	1,2	0,7	--	--
TO.117	2	1,7	--	--
SOT.48	1,8	1,5	--	--
DIA.4L	1,1	0,7	--	--
TO.66	1,1	0,65	--	--

Tabla 2.1. Tabla de resistencia térmica conenedor – Disipador.

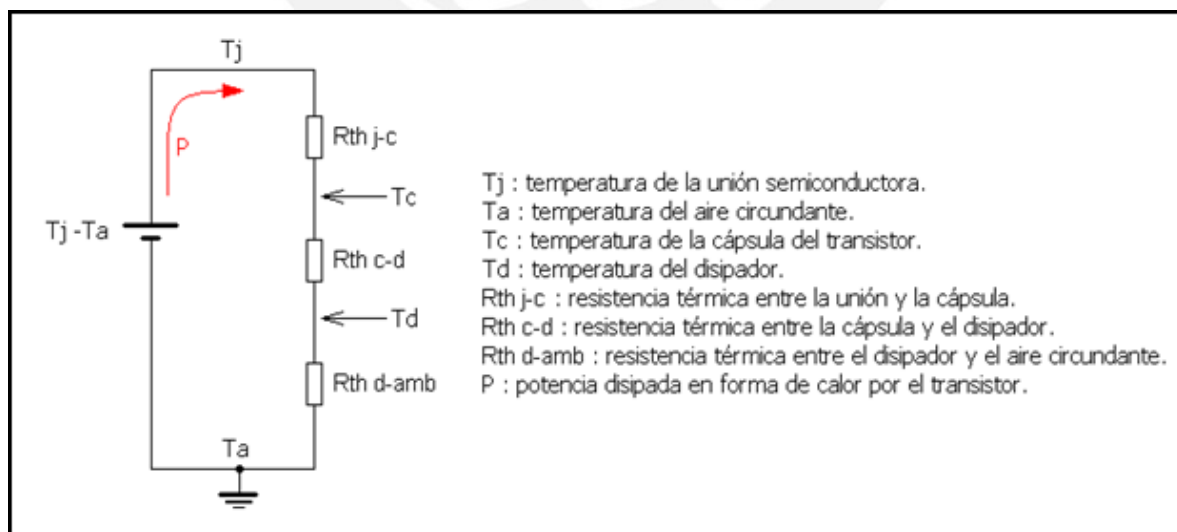


Figura 2.1. Equivalente eléctrico del sistema térmico.

Donde T_J debe estar dentro del rango de operación permitido del L298n, fijaremos esta temperatura en 100°C .

$$R_{th\ j-c} = Q_{JC} = 3^\circ\text{C/W}$$

$R_{th\ c-d}$ = Resistencia térmica entre la cápsula y el disipador, al estar utilizando un elemento con un encapsulado TO202 y silicona, vemos en la *tabla 2.1* que la resistencia térmica es 0.5°C/W .

$R_{th\ d-amb}$ = Resistencia térmica entre el disipador y el ambiente, que es la incógnita que queremos hallar.

$$T_A = 25^\circ\text{C}$$

Por lo tanto la ecuación quedará como sigue:

$$100^\circ\text{C} - 25^\circ\text{C} = (3^\circ\text{C/W} + 0.5^\circ\text{C/W} + R_{th\ d-amb}) \times 8.24\text{W}$$

$$R_{th\ d-amb} = 5.60^\circ\text{C/W}.$$

Por lo tanto se requiere un disipador con una resistencia térmica máxima de 5.60°C/W . Se puede observar en la *figura 2.2* tipos de disipadores.


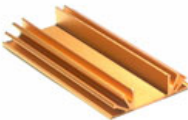

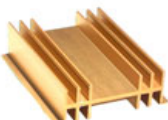

DISIPADOR	CARACTERISTICAS
	Artículo 6025 ZD-1 Dimensiones: Base 58 mm - Altura 29mm - Espesor núcleo central 3.5mm. Distancia interior entre aletas(del centro): 33 mm. Resistencia Térmica: 3.5°C/W para 75mm Superficie: $525,10\text{ mm}^2/\text{mm}$
	Artículo 6125 ZD-3 Dimensiones: Base 50mm - Altura 14mm - Espesor núcleo central 2mm. Distancia interior entre aletas(del centro): 20 mm. Resistencia térmica: 5°C/W para 75mm Superficie: $252\text{ mm}^2/\text{mm}$
	Artículo 6325M ZD-9 Dimensiones: Base 57mm - Altura 23mm - Espesor núcleo central 27mm Resistencia térmica: 3.80°C/W para 75mm Superficie: $343\text{mm}^2/\text{mm}$
	Artículo 4325M ZD-6 Dimensiones: Base 80mm - Altura 30mm - Espesor núcleo central 32mm. Distancia interior entre aletas(del centro): 32 mm. Resistencia térmica: 2.90°C/W para 75mm Superficie: $626,5\text{ mm}^2/\text{mm}$
	Artículo 7229 ZD-29 Dimensiones: Base 37mm - Altura 15mm - Espesor núcleo central 2mm Resistencia térmica: 10.8°C/W para 75mm Superficie: $156\text{ mm}^2/\text{mm}$

Figura 2.2. Tipos de disipadores (disipadores.com).

Se optó por el 6025ZD-1 ya que cumple con la resistencia térmica buscada (3.5°C/W es menor a la hallada) Y porque sus dimensiones son aproximadas a la del L298n, con este bloque el valor de la temperatura final del L298n será:

$$T_J = 25^{\circ}\text{C} + (3^{\circ}\text{C/W} + 0.5^{\circ}\text{C/W} + 3.5^{\circ}\text{C/W}) \times 8.24\text{W}$$

$$T_J = 82.68^{\circ}\text{C}.$$

Por lo que la temperatura del L298n estará dentro de su rango permitido de operación.

2.2 Cálculo de disipadores para los reguladores, si fuera necesario.

Para el 7805:

Como se halló en el capítulo 4, el consumo de corriente máximo es: $I_{\text{MAX}} = 60.45 \text{ mA}$ (micro, LEDs, pulsadores). Se observa una caída de tensión de 15V a 5V entonces $\Delta T = 10\text{V}$

Por lo tanto la potencia disipada debe ser:

$$P = 10\text{V} \times 60.45 \text{ mA} = 0.6045 \text{ Watts}.$$

Según hoja la hoja de datos:

Rango de operación: $[0; 125^{\circ}\text{C}]$

$$Q_{\text{JC}} = 5^{\circ}\text{C/W}$$

$$Q_{\text{JA}} = 60^{\circ}\text{C/W}$$

$$T_A = 25^{\circ}\text{C}$$

Por lo tanto:

$$T_J = 25^{\circ}\text{C} + (5^{\circ}\text{C/W} + 60^{\circ}\text{C/W}) \times 0.6045\text{W}$$

$$T_J = 64.29^{\circ}\text{C}$$

Temperatura que está dentro del rango de operación permitido, por lo que no requerirá utilizar ningún disipador.

7812:

Según la tabla de consumo de corriente: $I_{MAX} = 0.25 \text{ A}$, se observa una caída de tensión de 15V a 12V por lo que el $\Delta T = 3\text{V}$, por lo tanto la potencia disipada debe ser:

$$P = 3\text{V} \times 0.25 \text{ A} = 0.75 \text{ watts}$$

Según hojas de dato:

Rango de operación: [0; 125°C]

$$Q_{JC} = 5^\circ\text{C/W}$$

$$Q_{JA} = 60^\circ\text{C/W}$$

$$T_A = 25^\circ\text{C}$$

Por lo tanto:

$$T_J = 25^\circ\text{C} + (5^\circ\text{C/W} + 60^\circ\text{C/W}) \times 0.75 \text{ watts}$$

$$T_J = 73.75^\circ\text{C}.$$

Temperatura que se encuentra dentro del rango permitido de operación por lo que no requerirá un disipador.

78T12

Según la tabla de consumo de corriente: $I_{MAX} = 2.0 \text{ A}$ (consumo de la celda)

Se observa una caída de tensión de 15V a 12V entonces $\Delta T = 3\text{V}$

Por lo tanto la potencia disipada debe ser:

$$P = 3\text{V} \times 2.0 \text{ A} = 6 \text{ watts}$$

Según hojas de dato:

Rango de operación: [0; 150°C]

$$Q_{JC} = 2.5^\circ\text{C/W}$$

$$Q_{JA} = 65^\circ\text{C/W}$$

$$T_A = 25^\circ\text{C}$$

Por lo tanto:

$$T_J = 25^\circ\text{C} + (2.5^\circ\text{C/W} + 65^\circ\text{C/W}) \times 6 \text{ watts}$$

$$T_J = 430^\circ\text{C}$$

Lo cual está muy por encima de la temperatura máxima. Sabemos que el equivalente térmico será:

$$T_J = T_A + (Q_{JC} + R_{th\ c-d} + R_{th\ d-a}) \times P$$

Donde

T_J debe estar dentro del rango de operación permitido del 78T15, lo fijaremos en 140°C

$$R_{th\ j-c} = Q_{JC} = 2.5^\circ\text{C/W}$$

$R_{th\ c-d}$ = Resistencia térmica entre la cápsula y el disipador, al estar utilizando un elemento con un encapsulado TO202 y silicona, vemos en la *tabla 2.1* que la resistencia térmica es 0.5°C/W .

$R_{th\ d-amb}$ = Resistencia térmica entre el disipador y el ambiente, que es la incognita.

$$T_A = 25^\circ\text{C}$$

Por lo tanto la ecuación quedará:

$$140^\circ\text{C} - 25^\circ\text{C} = (2.5^\circ\text{C/W} + 0.5^\circ\text{C/W} + R_{th\ d-amb}) \times 6\text{W}$$

$$R_{th\ d-amb} = 16.16^\circ\text{C/W}.$$

Por lo tanto se requiere un disipador con una resistencia térmica máxima de 16.16°C/W

Se eligió el modelo 2125D ya que cuenta una resistencia térmica de 10°C/W , lo cual es inferior a la hallada anteriormente, por lo que cumple con llevar la temperatura del regulador dentro de su rango de operación permitido.

2.3 Calculo del Disipador para la celda Peltier

Según las hojas de dato del Peltier QMAX = 21.4 watts, $\Delta T = 67^{\circ}\text{C}$

El Peltier posee una resistencia térmica muy reducida por lo que podríamos obviarla ($Q_{JC} = 0$). Se utiliza para la unión del Peltier con el disipador una silicona conductora de calor por lo que su resistencia será $R_{th\ c-d} = 0.5\ ^{\circ}\text{C/W}$.

$$T_J = T_A + (Q_{JC} + R_{th\ c-d} + R_{th\ d-a}) \times P$$

T_J = temperatura en la cara caliente del Peltier la cual fijaremos en 45°C para que no se degrade y para que la temperatura en la cara fría pueda ser la mínima.

Por lo que la resistencia térmica requerida será:

$$45^{\circ}\text{C} = 25^{\circ}\text{C} + (0 + 0.5^{\circ}\text{C/W} + R_{th\ d-a}) \times 21.4\text{W}$$

$$R_{th\ d-a} = 0.435\ ^{\circ}\text{C/W}$$

Por lo que se requiere un disipador que tenga una resistencia térmica menor.

Debido a que los disipadores que tienen esta resistencia térmica son grandes y pesados lo cual no sería práctico para los fines del proyecto, se hará uso de un bloque de disipación constituido por un disipador más un ventilador de 12V, como se puede observar en la figura 2.4, el cual es utilizado normalmente para enfriar procesadores de computadoras, en la *figura 2.3* se muestran sus especificaciones técnicas.

Compañía	Speeze
Modelo	QuadroFlow V / Quadroflow VI
Disipador:	Aluminio con base de cobre / Aluminio
Dimensiones del disipador	92×92×32 mm / 95×95×90 mm (l × w × h)
Ventilador	Sleeve bearing 30 mil horas / Ball bearing 50 mil horas
Dimensiones del ventilador	90×90×25 mm (l × w × h)
Velocidad	2.000 RPM +/-10%
Nivel de ruido	22 dBA / 19 dBA
Corriente de aire	36.80 CFM
Plataforma	Intel Celeron D~2.93GHz (340J), Pentium 4~3.8GHz (775 Prescott), Pentium D~3.4GHz (775 Dual-Core) / Intel Celeron D~2.93GHz (340J), Pentium 4~3.8GHz (775 Prescott)
Consumo	2.4 W / 1.56 W
Resistencia térmica	0.312 °C/W / 0.368 °C/W

Figura 2.3. Especificaciones técnicas del bloque de disipación.



Figura 2.4. Bloque de disipación.

Como se puede observar en las especificaciones técnicas el bloque de disipación posee una resistencia térmica igual a 0.312 °C/W / 0.368°C/W, lo cual es menor a la resistencia hallada y por lo tanto sirve para los fines del proyecto.

ANEXO 3. DISEÑO DE LA CARCASA Y EL EQUIPO TERMINADO.

3.1. DISEÑO DE LA CARCASA

Para el diseño de la carcasa se tomo en cuenta las siguientes pautas:

3.1.1 Ubicación de la fuente de alimentación.

La fuente de alimentación se considerará externa al equipo, la cual se conectará al mismo mediante el conector RCA.

3.1.2 Tamaño de las tarjetas y disipadores.

Para poder tomar las medidas adecuadas para la carcasa se debe conocer la medida de las tarjetas y además de los disipadores que se van a utilizar.

Las medidas son las siguientes:

De la tarjeta de control: 8.5cm x 5 cm

De la tarjeta de la pantalla LCD: 8cm x 3,5cm

De la pantalla LCD: 7cm x 2.4cm

De la tarjeta Driver: 5cm x 5cm

Del disipador de la tarjeta driver: 10cm x 2,5cm x 5cm

La tarjeta de la pantalla LCD va montada directamente sobre la tarjeta de control, es por esto que las dimensiones aproximadas de este bloque son: 9,5 cm x 5cm.

3.1.3 Tipo y ubicación de los conectores para el aplicador, interface, SWITCH

De encendido y conector de alimentación.

Los conectores utilizados para el aplicador son dos del tipo micrófono metálico uno de tres puntos para la alimentación y señal del sensor de temperatura y otro de cuatro puntos para la alimentación de la celda Peltier y el ventilador, ambos irán posicionados en la cara frontal del equipo. La interface ira ubicada en la cara superior mientras que el SWITCH de encendido y conector RCA considerado para la alimentación irán posicionados en la cara posterior.

Entonces, tomando en cuenta lo anterior se procedió a realizar el diseño de la carcasa:

Cara superior

Las dimensiones se pueden observar en la *figura 3.1*.

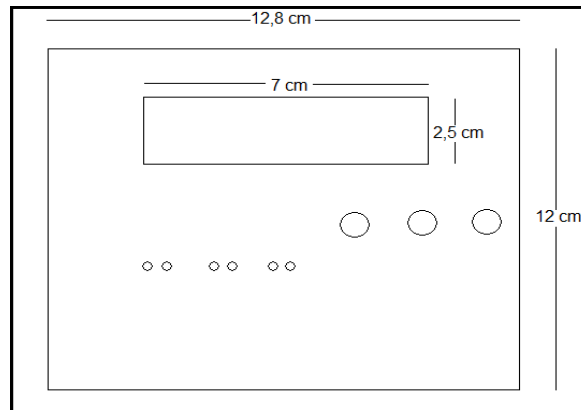


Figura 3.1. Dimensiones de la cara superior.

Se puede observar los agujeros considerados para la pantalla LCD, Leds indicadores y para los pulsadores.

Cara lateral

El diseño y las dimensiones se observan en la *figura 3.2*.

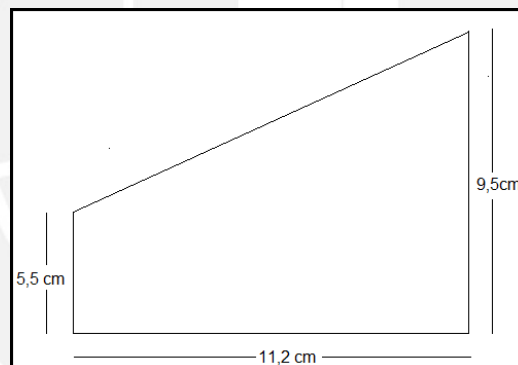


Figura 3.2. Dimensiones de la cara lateral.

Cara frontal

El diseño y las dimensiones se pueden observar en la *figura 3.3*.

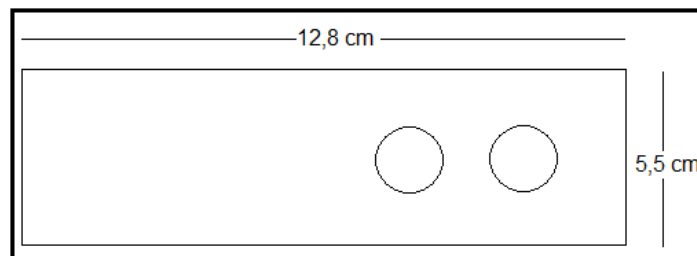


Figura 3.3. Dimensiones de la cara frontal.

Donde se puede observar los agujeros para los conectores del aplicador.

Cara posterior

Las dimensiones se pueden observar en la *figura 3.4*.

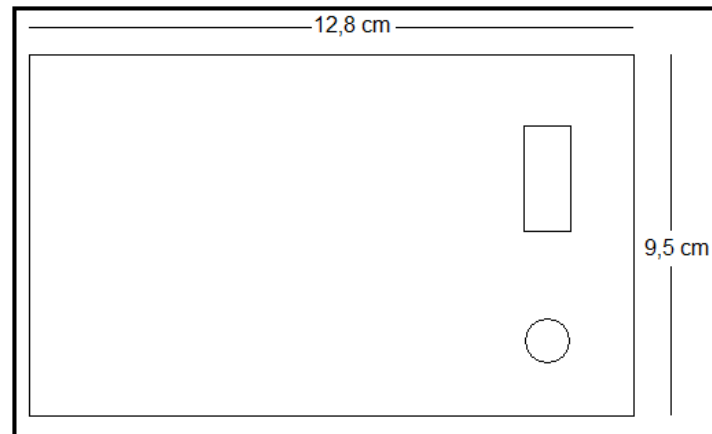


Figura 3.4. Dimensiones de la cara posterior.

Donde se puede observar los agujeros para el SWITCH de encendido y el conector de entrada para la tensión proveniente de la fuente de alimentación.

Cara inferior o Base

Las dimensiones se pueden observar en la *figura 3.5*.

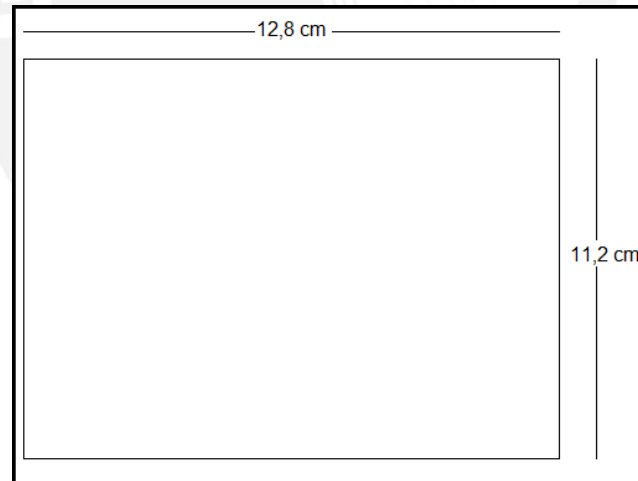


Figura 3.5. Dimensiones de la cara inferior.

3.2 CARCASA TERMINADA Y EQUIPO INSTALADO.

En las *figuras 3.6, 3.7, 3.8, 3.9 y 3.10* se puede observar las diferentes vistas del equipo ya instalado en la carcasa, el material utilizado para la elaboración de la carcasa fue acrílico de color blanco de 3mm de espesor.



Figura 3.6. Vista frontal superior.



Figura 3.7. Vista lateral del equipo.



Figura 3.8. Vista posterior.

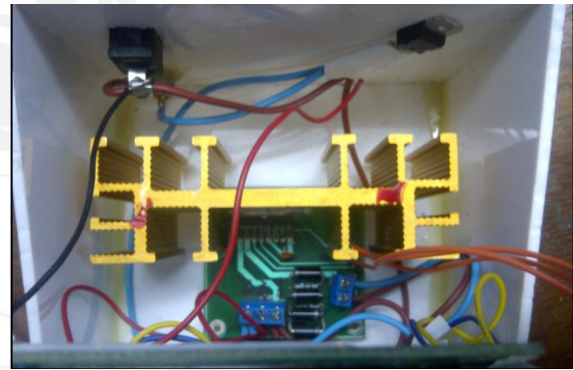


Figura 3.9. Vista interior.



Figura 3.10. Equipo conectado al aplicador y en funcionamiento.

ANEXO 4. CURVAS DE SCHADE.

Anexo 4.1.

Las curvas de la *figura 4.1* son las Curvas de Schade para rectificadores de media onda (OC, rojo), dobladores de tensión (DT, verde) y rectificadores de onda completa (MO, azul). Se indica el valor de $\omega C R_L$ necesario en función del factor de ripple V_{RMS}/V_{MED} requerido. Se da una curva para cada valor porcentual del parámetro R_s/RL .

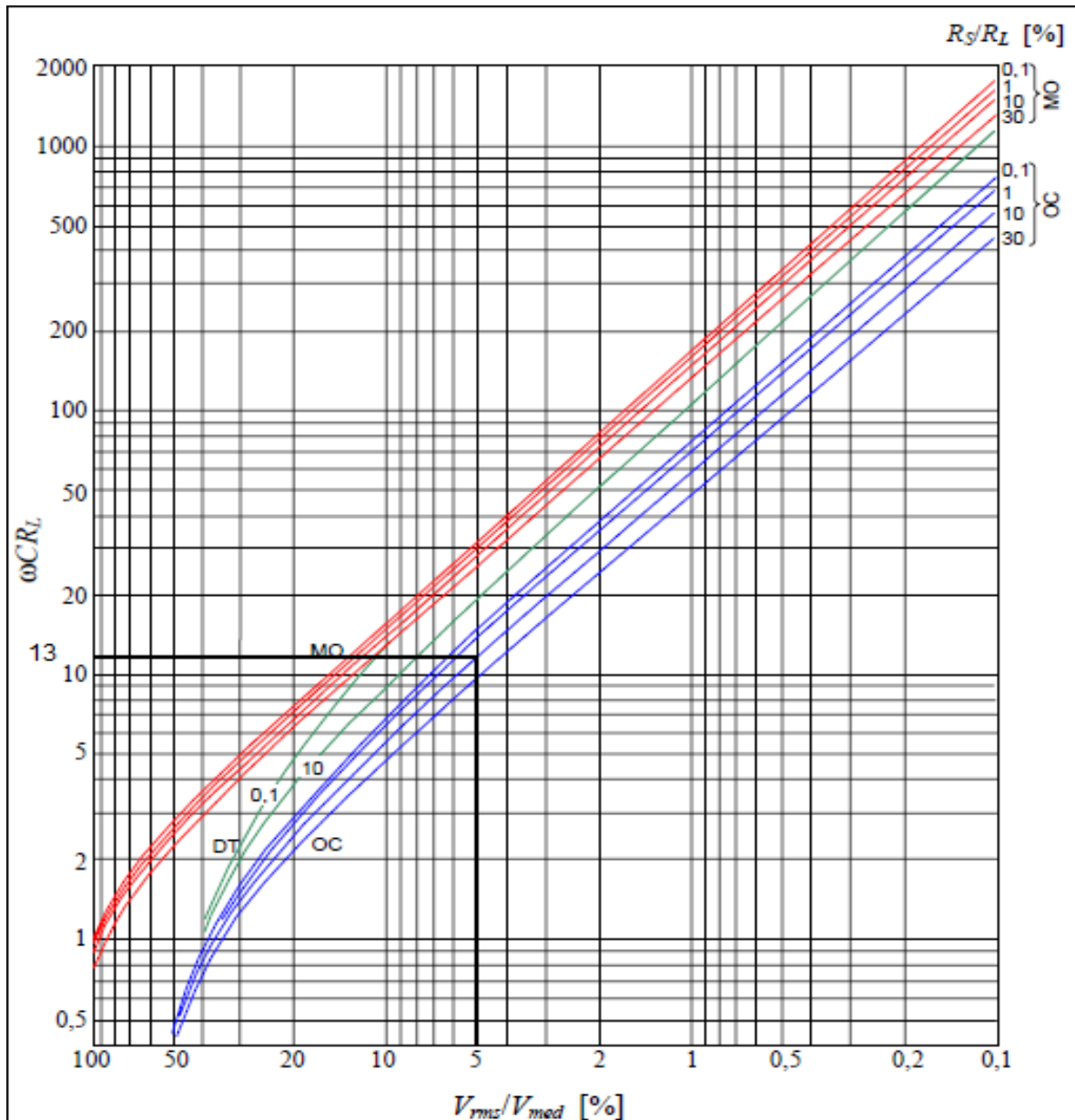


Figura 4.1. Curvas de Schade para $R_s/RL\%$ vs $\omega C R_L$ vs V_{RMS}/V_{MED} [32].

Anexo 4.2

Las curvas de la *figura 4.2* son las Curvas de Schade para rectificadores de onda completa. Se indica el valor porcentual de V_{MED} / V_P obtenido en función de ωCR_L . Se da una curva para cada valor porcentual del parámetro $R_S / R_L \%$

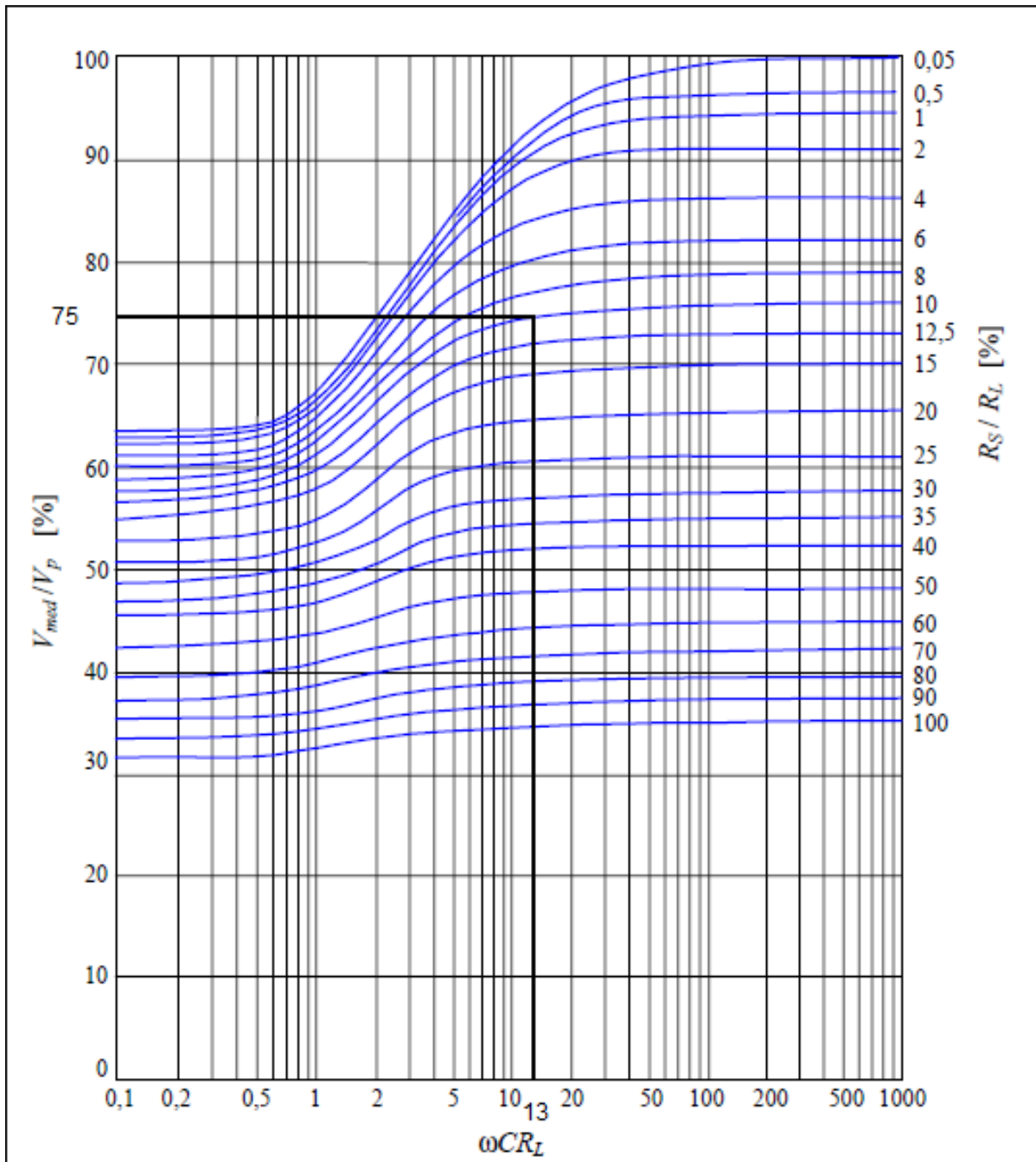


Figura 4.2. Curvas de Schade para V_{MED} / V_P VS ωCR_L VS $R_S / R_L \%$ [32].

Anexo 4.3.

Las curvas de la *figura 4.3* son las curvas de Schade para el cociente entre la corriente eficaz y la corriente media por cada diodo en función de $n\omega CR_L$. El valor de n es 2 para rectificadores de onda completa, 1 para rectificadores de media onda y 0,5 para dobladores de tensión. Se da una curva para cada valor porcentual del parámetro $R_s/nR_L\%$.

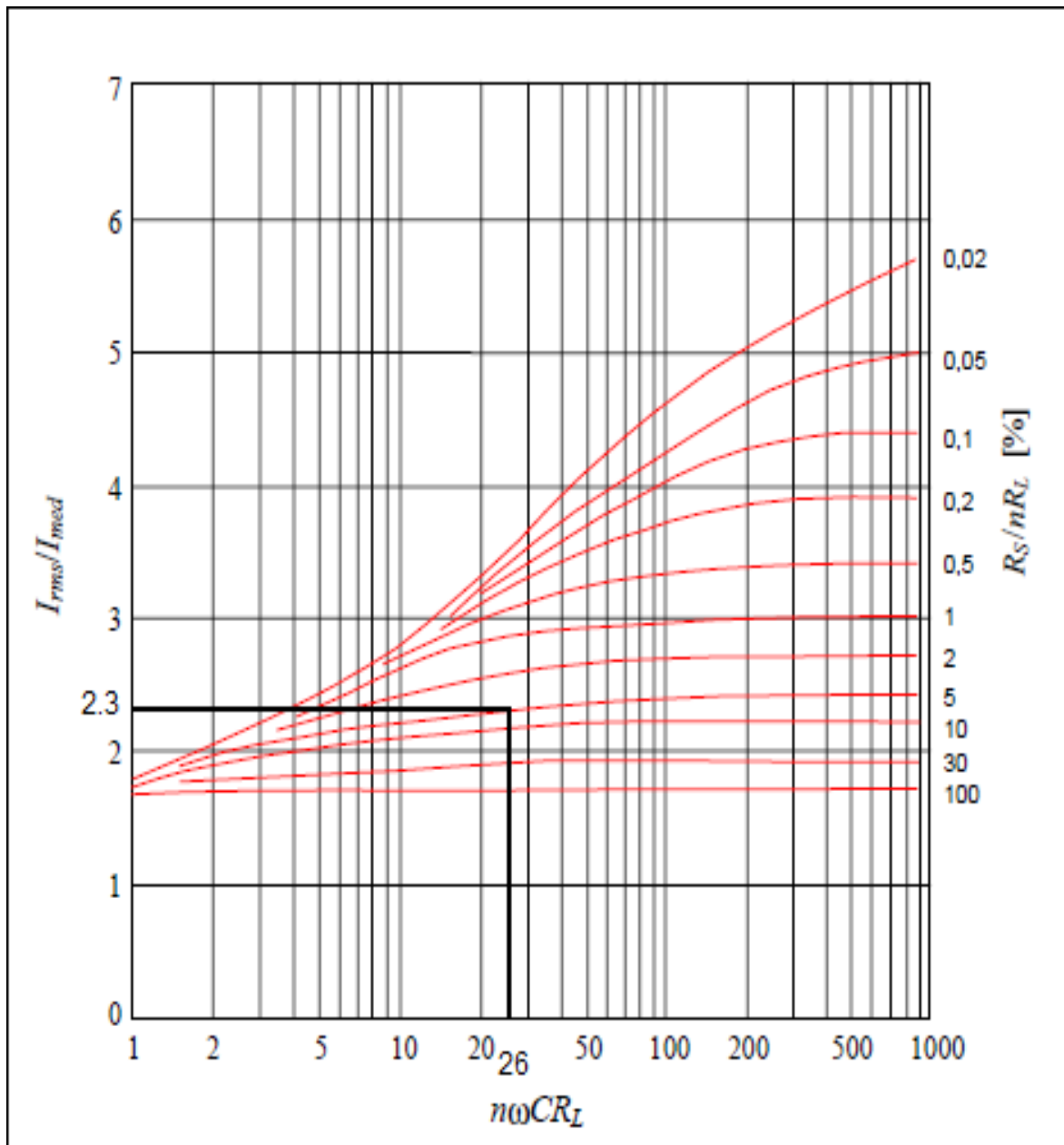


Figura 4.3. Curvas de Schade I_{RMS}/I_{MED} VS $n\omega CR_L$ VS R_s/nR_L [32].