

# Proyecto: Dispositivo autorregulador de la percepción térmica corporal

Martinez, Gaston - 91383  
gaston.martinez.90@gmail.com

Vázquez, Matías - 91523  
mfvazquez@gmail.com

24 de junio de 2015

*Se diseñará e implementará una pulsera térmica que regulará la temperatura corporal. Se utilizará un módulo termoeléctrico para enviar variaciones de calor o frío a la muñeca del usuario para modificar la percepción térmica del cuerpo.*

## 1. Introducción

Su función es generar pulsos de frío o calor, de manera de generar una sensación de confort para una persona en condiciones donde la temperatura es muy alta o muy baja respectivamente. Está basado en el proyecto *Wristify* [1] ganador del concurso de intel *Make It Wearable* [2].

## 2. Especificaciones

El dispositivo utilizará una celda Peltier para enviar pulsos de calor o frío. De forma que se logre una diferencia de temperatura de 0,4 °C/seg. durante 5 segundos y durante los siguientes 10 segundos entrará en estado de espera, para luego volver a iniciar el ciclo.

Deberá contar con un sensor de temperatura para medir la temperatura ambiente y analizar si deberá enviar o recibir calor.

Finalmente deberá controlar que se cumpla el ciclo utilizando una termocupla para medir la temperatura corporal cercana a la placa de peltier.

### 2.1. Componentes

Deberá contar con los siguientes componentes.

- Placa de peltier: Generará los pulsos térmicos en la muñeca del usuario.
- Termómetro: Medirá la temperatura ambiente y en base a ella decidirá si se debe aumentar o reducir la temperatura en la termocupla.
- Termocupla: Contará con una doble finalidad. Por un lado permitirá medir el cambio de temperatura de la placa; y por el otro permitirá medir la temperatura actual del cuerpo al momento de colocarse la pulsera.
- Salida de puerto serie: Servirá para poder monitorear en una computadora la temperatura de la placa.
- Fuente: Se encargará de suministrar la corriente necesaria a la placa de peltier y proporcionará alimentación a todos los dispositivos utilizados.
- Pulsador: Para poder invertir el estado de trabajo, de frío a calor y viceversa.
- Disipador: Se encargará de disipar el calor del lado opuesto al de la muñeca de la placa de peltier.
- Controlador: Se utilizara un microcontrolador AVR. Recibirá la temperatura ambiente del termómetro para decidir que régimen de trabajo establecer, y con la temperatura suministrada por la termocupla decidirá cuanta corriente suministrarle a la celda Peltier mediante un circuito regulador de corriente. También estará conectado a un pulsador para invertir el régimen de trabajo.

## 2.2. Diagrama de Flujo

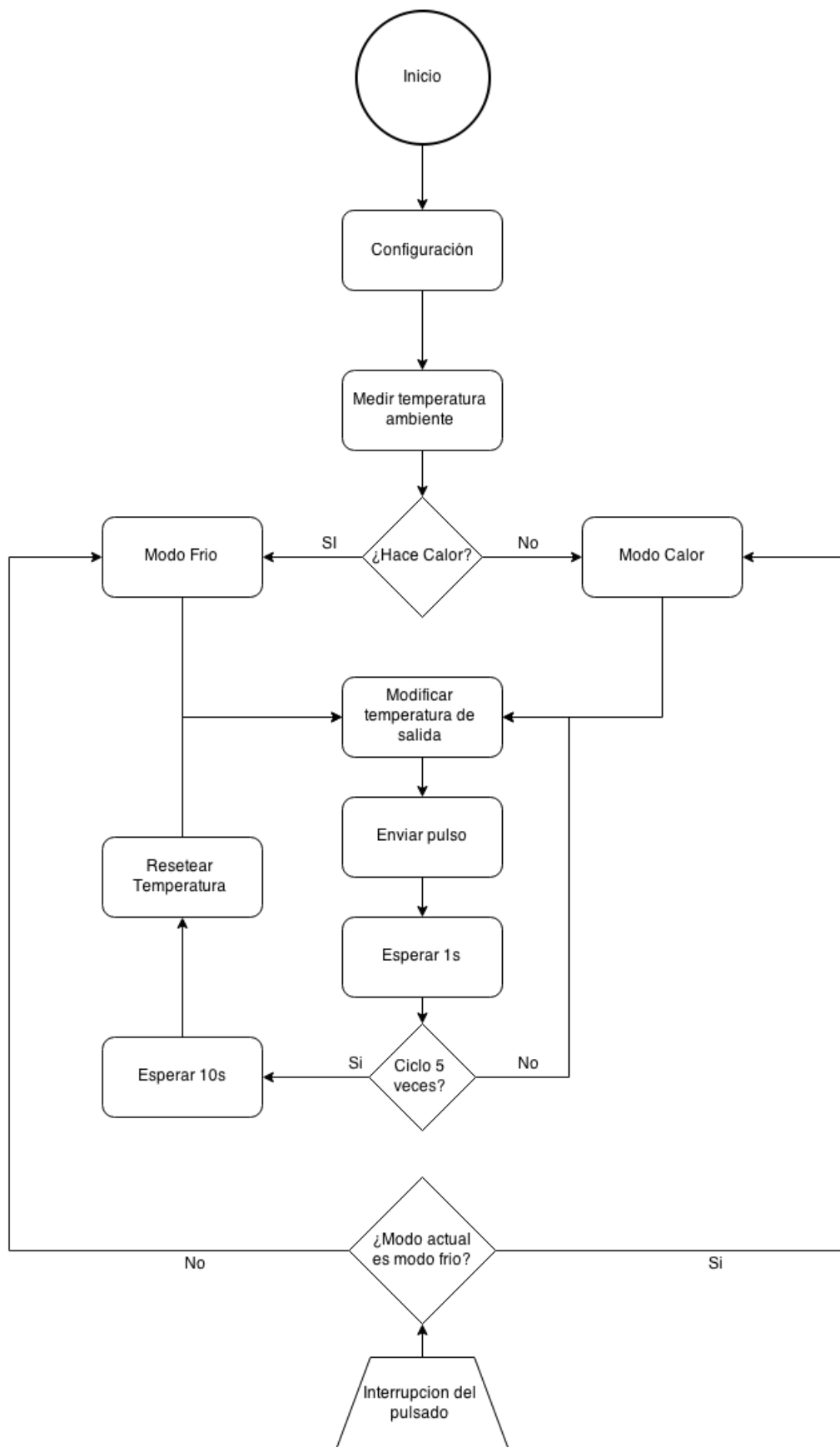


Figura 1: Diagrama de flujo del proceso

## 2.3. Diagrama de Bloques

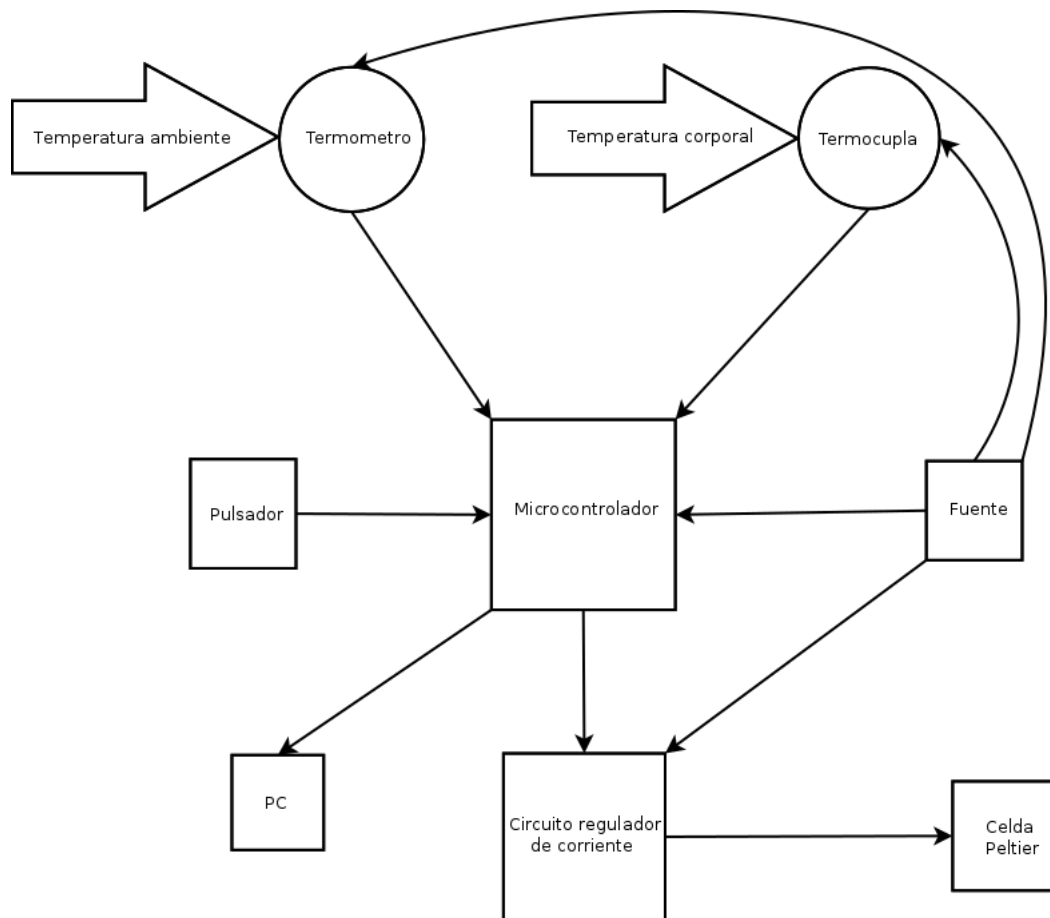


Figura 2: Diagrama de bloques

## 3. Diseño

### 3.1. Circuito regulador de corriente

Para la construcción del circuito regulador de corriente se utilizará un regulador de tensión **LM317**. Partiendo del circuito mostrado en la figura número 3. Se obtendrá el valor mínimo de  $R_1$  para obtener la corriente máxima de salida  $I_{out}$  mediante la ecuación número 1.

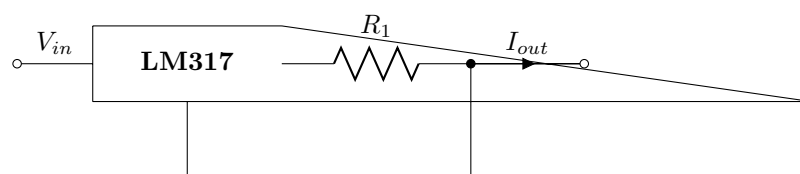


Figura 3: Circuito regulador de corriente

$$I_{out} = \frac{1,25 \text{ V}}{R_1} \quad (1)$$

La corriente máxima de salida será necesaria para alcanzar una diferencia de temperatura de  $2^\circ\text{C}$  entre una de las caras de la celda Peltier y la temperatura ambiente. Que se deberá obtener experimentalmente con la celda Peltier utilizada.

## 4. Especificaciones del microcontrolador

### 4.1. Microcontrolador

Para este proyecto se utilizó un microcontrolador Atmega8L. El datasheet del mismo se puede obtener en la página de Atmel[4]

### 4.2. Configuraciones

#### 4.2.1. Clock

El clock del microcontrolador fue establecido en 8MHz.

7	6	5	4	3	2	1	0
a	a	a	a	a	a	a	a
0	0	0	0	0	0	0	0

#### 4.2.2. PWM

El PWM

## 5. Codigo Proyecto

```
1  #include <avr/io.h>

3  .section .data
   .org 0x000

5
   Temp_Ambiente: .byte 0
7   Temp_Disipador: .byte 0
   Temp_Peltier: .byte 0
9   Tension_Salida: .byte 0

11  Modo_Operacion: .byte 0

13  Modo_Standby: .byte 0

15  PWM: .byte 0
   Tension_min: .byte 0
17  Tension_max: .byte 0
   Iterador: .byte 0

19
   .section .text
21  .org 0x0
   .global main
23  rjmp main

25  #define low(x)    lo8(x)
   #define high(x)   hi8(x)

27
   ;Regs:
29   ;r16: Temporal, pasaje de parametro y de retorno
   #define Reg_Temporal r16

31
   ;r20: Contador
33   #define Contador r20

35
   ;Constantes:
   ;Tipos de dato para mandar por serial

37
   #define Dato_Tempe_Ambiente 'A'
39   #define Dato_Tempe_Disipador 'T'
   #define Dato_Num_Iteracion 'I'
41   #define Dato_Tension_Salida 'P'
   #define Dato_Tempe_Peltier 'X'
43   #define Dato_PWM 'W'
   #define Dato_max 'M'
45   #define Dato_min 'N'

47   #define Incremento_Pulso_Calor 3
   #define Incremento_Pulso_Frio 10
49   #define Incremento_Regulacion 1
   #define PWM_inicial 95

51
   #define Eeprom_Inicio_Calor 0x64
53   #define Eeprom_Inicio_Frio 0x84

55   ;Saltea el vector de óinterrupcin
   .org 0x0020
57   main:
```

```

59  STACK_Init:
    ldi    Reg_Temporal,    low(RAMEND)
61  out    _SFR_IO_ADDR(SPL),    Reg_Temporal
    ldi    Reg_Temporal,    high(RAMEND)
63  out    _SFR_IO_ADDR(SPH),    Reg_Temporal

65  rcall   PWM_Init
    rcall   PUENTE_H_Init
67  rcall   USART_Init

69  LOOP:
    clr     Contador                ;Iteraciones, se usa para saber la diferencia
71
    rcall   GET_MODE                ;obtengo el modo de operacion
73
    ldi     r29, PWM_inicial        ;PWM
75  sts     PWM, r29
    rcall   RESET_PWM
77
REDUCIR_LOOP:
79
    mov     Reg_Temporal,    Contador
81
    ldi     r27, 10
83  rcall   ESPERA
    rcall   AUMENTAR_PULSO
85

87  inc     Contador                ;Cumpli una vuelta
    ldi     Reg_Temporal,    6
89  cpse    Contador,    Reg_Temporal    ;No saltar si ya ejecuto 5 vueltas

91  rjmp    REDUCIR_LOOP

93  cbi     _SFR_IO_ADDR(DDRB),    3    ;(OC2) para salida
    sbi     _SFR_IO_ADDR(DDRB),    3    ;(OC2) para salida
95  cbi     _SFR_IO_ADDR(PORTB),    3

97  ;    rcall STANDBY
    rjmp    LOOP
99
    ;SEND SNIPPET
101 ;ldi Reg_Temporal, 'A' ;tipo de dato a mandar
    ;rcall USART_Transmit
103 ;mov Reg_Temporal, Modo_Operacion ;No se, imprimo el modo
    ;rcall USART_Transmit
105
    ;-----Funciones-----
107 RESET_PWM:
    clr     r17                    ; 8 bits mas significativos de la direccion
109
    lds     r31, Modo_Operacion
111  cpi     r31, 0
    breq    TABLA_FRIO
113 TABLA_CALOR:
    ldi     Reg_Temporal, Eeprom_Inicio_Calor
115  rjmp    CARGAR_TABLA
TABLA_FRIO:
    ldi     Reg_Temporal, Eeprom_Inicio_Frio
117 CARGAR_TABLA:

```

```

119     inc      Reg_Temporal
120     inc      Reg_Temporal
121     sts      Iterador , Reg_Temporal
122     rcall    LEER_EEPROM
123     sts      Tension_min , r25

125     inc      Reg_Temporal
126     inc      Reg_Temporal
127     rcall    LEER_EEPROM
128     sts      Tension_max , r25

129
130     lds      r29 , PWM
131     out      _SFR_IO_ADDR(OCR2) , r29

132
133     ret
;-----
134 ; espera r27 * 100mseg
135 ESPERA:
136     rcall    SET_PWM
137     rcall    TRANSMITIR_DATOS
138     rcall    DEMORA
139
140
141
142     dec      r27
143     clr      Reg_Temporal
144     cpse     r27 , Reg_Temporal
145     rjmp     ESPERA
146     ret
147
;-----
148 ; Lee de eeprom en la direccion indicada en los registros r17 para los 8 bits
149 ; mas significativos y r16 para los 8 bits menos significativos. Guarda el contenido
150 ; en el registro r25
151 LEER_EEPROM:
152     sbic     _SFR_IO_ADDR(EECR) , EEWB      ; espera hasta que la ultima escritura este te
153     rjmp     LEER_EEPROM
154     out      _SFR_IO_ADDR(EEARH) , r17      ; r17 elige la tabla
155     out      _SFR_IO_ADDR(EEARL) , r16      ; r16 el campo
156     sbi      _SFR_IO_ADDR(EECR) , EERE      ; habilita el modo lectura
157     in       r25 , _SFR_IO_ADDR(EEDR)      ; guarda el contenido de la ódireccin antes ca
158     ret
159
;-----
160 ; Inicializa el puente H seteando los pines 1 y 2 del puerto B como salida.
161 PUENTE_H_Init:
162     sbi      _SFR_IO_ADDR(DDRB) , 2
163     sbi      _SFR_IO_ADDR(DDRB) , 1
164     ret
165
166
167 MODO_FRIO:
168     sbi      _SFR_IO_ADDR(PORTB) , 1
169     cbi      _SFR_IO_ADDR(PORTB) , 2
170     ret
171
172 MODO_CALOR:
173     cbi      _SFR_IO_ADDR(PORTB) , 1
174     sbi      _SFR_IO_ADDR(PORTB) , 2
175     ret
176
;-----

```

```

179 TRANSMITIR_DATOS:
    rcall    LEER_AMBIENTE
181    rcall    LEER_DISPADOR
    rcall    LEER_PELTIER

183
185    ldi      Reg_Temporal,    Dato_Num_Iteracion    ;Envio la iteracion
    rcall    USART_Transmit
    mov      Reg_Temporal,    Contador
187    rcall    USART_Transmit

189    ldi      Reg_Temporal,    Dato_Tempe_Ambiente    ; tipo de dato a mandar
    rcall    USART_Transmit
191    lds      Reg_Temporal,    Temp_Ambiente
    rcall    USART_Transmit

193
195    ldi      Reg_Temporal,    Dato_Tension_Salida    ; tipo de dato a mandar
    rcall    USART_Transmit
197    lds      Reg_Temporal,    Tension_Salida
    rcall    USART_Transmit

199    ldi      Reg_Temporal,    Dato_Tempe_Disipador    ; tipo de dato a mandar
    rcall    USART_Transmit
201    lds      Reg_Temporal,    Temp_Disipador
    rcall    USART_Transmit

203
205    ldi      Reg_Temporal,    Dato_Tempe_Peltier    ; tipo de dato a mandar
    rcall    USART_Transmit
207    lds      Reg_Temporal,    Temp_Peltier
    rcall    USART_Transmit

209    ldi      Reg_Temporal,    Dato_PWM    ; tipo de dato a mandar
    rcall    USART_Transmit
211    lds      Reg_Temporal,    PWM
    rcall    USART_Transmit

213
215    ldi      Reg_Temporal,    Dato_min    ; tipo de dato a mandar
    rcall    USART_Transmit
217    lds      Reg_Temporal,    Tension_min
    rcall    USART_Transmit

219    ldi      Reg_Temporal,    Dato_max    ; tipo de dato a mandar
    rcall    USART_Transmit
221    lds      Reg_Temporal,    Tension_max
    rcall    USART_Transmit

223
    ret

225 ;-----
LEER_AMBIENTE:
227    ldi      Reg_Temporal,    0b11000000    ; canal 0 temperatura ambiente
    rcall    READ_ADC    ; leer tension del peltier
229    rcall    TRADUCIR_TERMISTOR
    sts      Temp_Ambiente,    Reg_Temporal
231    ret

233 LEER_DISPADOR:
    ldi      Reg_Temporal,    0b11000010    ; canal 2 temperatura disipador
235    rcall    READ_ADC    ; leer tension del termistor
    rcall    TRADUCIR_TERMISTOR
237    sts      Temp_Disipador,    Reg_Temporal
    ret

```



```

239 LEER_PELTIER:
241     ldi      Reg_Temporal,      0b11000001          ; canal 1 tension peltier
242     rcall    READ_ADC              ; leer tension del peltier
243     sts      Tension_Salida, Reg_Temporal
244     rcall    TRADUCIR_PELTIER
245     ret

247 ;-----
248 ;Set PWM
249 ;Setea el pum del pin OC2 con el tiempo en bajo pasado como parametro
250 ;Reg_Temporal: tiempo en bajo a asignar

251 SET_PWM:
252     lds      r30, PWM
253     lds      Reg_Temporal, Modo_Standby
254     cpi      Reg_Temporal, 1
255     breq     APLICAR_CAMBIO          ;Si esta en modo standby no realiza cambios
256
257     lds      r29, Tension_Salida
258     lds      r17, Tension_min
259     lds      r18, Tension_max
260     ldi      r26, Incremento_Regulacion          ;valor a ser restado o sumado
261
262     cp       r29, r17
263     brlo     AUMENTAR
264
265     cp       r29, r18
266     brlo     APLICAR_CAMBIO
267
268 DISMINUIR:
269     cpi      r30, 100
270     breq     APLICAR_CAMBIO
271     add      r30, r26
272     rjmp     APLICAR_CAMBIO
273 AUMENTAR:
274     cpi      r30, 0
275     breq     APLICAR_CAMBIO
276     sub      r30, r26
277 APLICAR_CAMBIO:
278     sts      PWM, r30
279     out      _SER_IO_ADDR(OCR2), r30
280     ret
281 ;-----
282
283 AUMENTAR_PULSO:
284     clr      r17          ; 8 bits mas significativos de la direccion
285
286     lds      Reg_Temporal, Iterador
287     inc      Reg_Temporal
288     inc      Reg_Temporal
289     sts      Iterador, Reg_Temporal
290
291     rcall    LEER_EEPROM
292     sts      Tension_min, r25
293     inc      Reg_Temporal
294     inc      Reg_Temporal
295     rcall    LEER_EEPROM
296     sts      Tension_max, r25

```

```

299     lds      r31, Modo_Operacion
300     cpi      r31, 0
301     breq     AUMENTO_FRIO
AUMENTO_CALOR:
303     ldi      r26, Incremento_Pulso_Calor
304     rcall    AUMENTAR
305     ret
AUMENTO_FRIO:
307     ldi      r26, Incremento_Pulso_Frio
308     rcall    AUMENTAR
309     ret

311 ;-----
312 ;Espera durante 100mseg
313 DEMORA:
314     ldi      Reg_Temporal, 0xCF                ; Valores de los que empieza a contar
315     out      _SFR_IO_ADDR(TCNT1H), Reg_Temporal
316     ldi      Reg_Temporal, 0x2B
317     out      _SFR_IO_ADDR(TCNT1L), Reg_Temporal
318     ldi      Reg_Temporal, 4                    ; 0000 0100 habilita poner en 1
319     out      _SFR_IO_ADDR(TIFR), Reg_Temporal
320     out      _SFR_IO_ADDR(TIMSK), Reg_Temporal ; el bit 3 de TIRF cuando haya overflow
321     ldi      Reg_Temporal, 0b00000011          ; velocidad: clk/64
322     out      _SFR_IO_ADDR(TCCR1B), Reg_Temporal
323 DEMORA_LOOP:
324     in       Reg_Temporal, _SFR_IO_ADDR(TIFR)
325     sbrc     Reg_Temporal, 2
326     rjmp     DEMORA_LOOP
327
328     ldi      Reg_Temporal, 1
329     out      _SFR_IO_ADDR(TIFR), Reg_Temporal
330     clr      Reg_Temporal                    ; finalizo contador
331     out      _SFR_IO_ADDR(TIFR), Reg_Temporal
332     out      _SFR_IO_ADDR(TCCR1B), Reg_Temporal
333
334     ret
335
336 ;-----
337 ;Standby
338 ;Espera durante 10 segundos
339 STANDBY:
340     ldi      Reg_Temporal, 1
341     sts      Modo_Standby, Reg_Temporal
342     ldi      r29, 255                        ;PWM
343     sts      PWM, r29
344     rcall    SET_PWM
345     ldi      r27, 100
346     rcall    ESPERA
347     clr      Reg_Temporal
348     sts      Modo_Standby, Reg_Temporal
349     ret
350
351 ;-----
352 ;Change Mode
353 ;Invierte el bit 0 en el registro Modo_Operacion que es el modo de funcionamiento
354 CHANGE_MODE:                                ;Atiende la interrupcion de cambio de modo
355     ;inc     Modo_Operacion                ;Invierte el ultimo bit
356     rjmp     LOOP

```

```

359 ;-----
    ;Transmit
361 ;Transmite por el puerto paralelo el dato pasado como parametro
    ;Reg_Temporal: valor a transmitir
363
USART_Transmit:
365     sbis     _SFR_IO_ADDR(UCSRA), UDRE           ;Espero a que se libere el UDRE
    rjmp     USART_Transmit
367
    out      _SFR_IO_ADDR(UDR), Reg_Temporal
369     ret

371 ;-----
    ;Uart init
373 ;Inicializa el USART para poder enviar datos

375 USART_Init:
    ldi      Reg_Temporal, (1<<TXEN)           ;enable
377     out     _SFR_IO_ADDR(UCSRB), Reg_Temporal

379     ldi      Reg_Temporal, (1<<URSEL)|(3<<UCSZ0) ;8bits, 1bit de stop, sin bit de paridad
    out     _SFR_IO_ADDR(UCSRC), Reg_Temporal
381
    ldi      Reg_Temporal, 0xC                 ;Baud 4800
383     out     _SFR_IO_ADDR(UBRR1L), Reg_Temporal

385     ret

387 ;-----
    ;Read adc
389 ;Lee un dato del conversor adc y lo devuelve
    ;Reg_Temporal: canal del cual leer
391 ;Reg_Temporal: valor leído devuelto

393 READ_ADC:

395     out     _SFR_IO_ADDR(ADMUX), Reg_Temporal ;
    ldi      Reg_Temporal, 0b11001111          ;
397     out     _SFR_IO_ADDR(ADCSRA), Reg_Temporal ;

399 WAIT_ADC:

401     in      Reg_Temporal, _SFR_IO_ADDR(ADCSRA) ;READ THE STATUS

403     sbrs    Reg_Temporal, 4
    rjmp     WAIT_ADC
405
    sbi      _SFR_IO_ADDR(ADCSRA), 4

407
    in      Reg_Temporal, _SFR_IO_ADDR(ADCL) ;
409     in      r17, _SFR_IO_ADDR(ADCH) ;
    lsr     r17
411     ror     Reg_Temporal
    lsr     r17
413     ror     Reg_Temporal

415     ret

417 ;-----
    ;PWM init

```

```

419 ;Inicializa los puertos de salida del pum
PWM_Init:
421     sbi     _SFR_IO_ADDR(DDRB) ,      3           ;(OC2) para salida
         ldi     Reg_Temporal ,      0x71           ;(01110001) Phase correct, no pre escala
423     out     _SFR_IO_ADDR(TCCR2) ,      Reg_Temporal
         ret

425
;-----
427 ;Traducir termistor
;Convierte el valor recibido por parametro en su temperatura equivalente
429 ;entrada: Reg_Temporal: valor leído por el ADC

431 TRADUCIR_TERMISTOR:
         mov     r18 ,      Reg_Temporal
433         ldi     r17 ,      0           ;tabla termistor
         ldi     Reg_Temporal ,      0           ;indice
435 LOOP_BUSQUEDA_TERM:
         rcall    LEER_EEPROM
437         cpi     r25 ,      0
         breq     FIN_TABLA
439         cp      r18 , r25           ;leído vs valor tabla
         brsh     END_TERMISTOR
441         inc     Reg_Temporal
         inc     Reg_Temporal
443         rjmp    LOOP_BUSQUEDA_TERM

445 END_TERMISTOR:
         inc     Reg_Temporal
447         rcall    LEER_EEPROM
         mov     Reg_Temporal ,      r25
449         ret

451 FIN_TABLA:
         dec     Reg_Temporal
453         dec     Reg_Temporal
         rjmp     END_TERMISTOR
455
;-----
457 ;Traducir peltier
;Convierte el valor recibido por parametro en su temperatura equivalente
459 ;entrada: Temp_Disipador, Tension_Salida, Modo_Operacion
;salida: Temp_Peltier

461 TRADUCIR_PELTIER:
463         lds     r18 , Tension_Salida
         ldi     r17 ,      0           ;tabla termistor
465         lds     r31 , Modo_Operacion
         cpi     r31 ,      0
467         breq     TABLA_FRIO_PELTIER
TABLA_CALOR_PELTIER:
469         ldi     Reg_Temporal , Eeprom_Inicio_Calor
         rjmp     LOOP_BUSQUEDA_TERM_PELTIER
471 TABLA_FRIO_PELTIER:
         ldi     Reg_Temporal , Eeprom_Inicio_Frio
473         rjmp     LOOP_BUSQUEDA_TERM_PELTIER
LOOP_BUSQUEDA_TERM_PELTIER:
475         rcall    LEER_EEPROM
         cpi     r25 ,      0xFF
477         breq     FIN_TABLA_PELTIER
         cp      r25 , r18           ; valor tabla vs leído

```

```

479     brsh    END_PELTIER
        inc    Reg_Temporal
481     inc    Reg_Temporal
        rjmp   LOOP_BUSQUEDA_TERM_PELTIER
483 END_PELTIER:
        inc    Reg_Temporal
485     rcall   LEER_EEPROM
        lds    r17, Temp_Disipador
487     lds    r18, Modo_Operacion

489     mov     Reg_Temporal, r25

491     sbrc    r18,0
        rjmp   CALCULO_CALOR
493     rjmp   CALCULO_FRIO

495 CALCULO_CALOR:
        add     Reg_Temporal, r17
497     sts     Temp_Peltier, Reg_Temporal
        ret

499 CALCULO_FRIO:
501     sub     r17, Reg_Temporal
        mov     Reg_Temporal, r17
503     sts     Temp_Peltier, Reg_Temporal
        ret

505 FIN_TABLA_PELTIER:
507     dec     Reg_Temporal
        dec     Reg_Temporal
509     rjmp    END_PELTIER

511 ;-----
512 ;Get mode
513 ;Devuelve el valor del modo en el cual se ejecuta para la temperatura Reg_Temporal
514 ;Reg_Temporal: temperatura leida
515 ;Reg_Temporal: valor leido devuelto
GET_MODE:
517     ldi     Reg_Temporal, 0b11000000        ; canal 0 temperatura ambiente
        rcall   READ_ADC
519     rcall   TRADUCIR_THERMISTOR             ; obtengo la temperatura
        cpi     Reg_Temporal, 50              ; temperatura arbitraria para el modo
521     brsh    COLD_MODE

523 HOT_MODE:
        sbic    _SFR_IO_ADDR(PINB), 0        ; Si el pin esta en 1 cambia de modo
525     rjmp    SET_COLD

527 SET_HOT:
        cbi     _SFR_IO_ADDR(PORTB), 1
529     sbi     _SFR_IO_ADDR(PORTB), 2

531     ldi     Reg_Temporal, 1                ; 1 es modo calor
        sts     Modo_Operacion, Reg_Temporal ; Modo_Operacion modo de operacion iniciado
533     ret

535 COLD_MODE:
537     sbic    _SFR_IO_ADDR(PINB), 0        ; Si el pin esta en 1 cambia de modo
        rjmp    SET_HOT

```

539

SET\_COLD:

541

```
sbi _SFR_IO_ADDR(PORTB), 1
cbi _SFR_IO_ADDR(PORTB), 2
```

543

```
ldi Reg_Temporal, 0 ;0 es el modo frio
sts Modo_Operacion, Reg_Temporal ;Modo_Operacion modo de operacion inicia
```

547

```
ret
```

549

```
.section .eeprom
.org 0x0000
```

551

```
;Tablas de conversion
```

553

```
;Formato: [(Tension medida, Temperatura*2),...]
```

555

```
.byte 204 , 14
```

555

```
.byte 201 , 16
```

557

```
.byte 199 , 18
```

557

```
.byte 197 , 20
```

559

```
.byte 195 , 22
```

559

```
.byte 194 , 24
```

561

```
.byte 193 , 26
```

561

```
.byte 192 , 28
```

563

```
.byte 190 , 30
```

563

```
.byte 188 , 32
```

565

```
.byte 186 , 34
```

565

```
.byte 184 , 36
```

567

```
.byte 182 , 38
```

567

```
.byte 181 , 40
```

569

```
.byte 172 , 42
```

569

```
.byte 169 , 44
```

571

```
.byte 166 , 46
```

571

```
.byte 163 , 48
```

573

```
.byte 160 , 49
```

573

```
.byte 159 , 50
```

575

```
.byte 158 , 53
```

575

```
.byte 157 , 56
```

577

```
.byte 156 , 60
```

577

```
.byte 155 , 62
```

579

```
.byte 154 , 64
```

579

```
.byte 153 , 66
```

581

```
.byte 152 , 68
```

581

```
.byte 151 , 70
```

583

```
.byte 147 , 72
```

583

```
.byte 145 , 74
```

585

```
.byte 143 , 76
```

585

```
.byte 141 , 78
```

587

```
.byte 140 , 80
```

587

```
.byte 0, 0 ;FIN DE TABLA
```

589

```
; TABLA PARA MODO CALOR
```

591

```
;Formato: [Tension minima, Dif_Temperatura*2]
```

591

```
.org 0x0064
```

593

```
.byte 0 , 0
```

593

```
.byte 5 , 0
```

595

```
.byte 10 , 1
```

595

```
.byte 15 , 2
```

597

```
.byte 20 , 3
```

597

```
.byte 25 , 4
```

597

```
.byte 30 , 5
```

```
599 .byte    35    ,    6
      .byte    40    ,    7
601 .byte    45    ,    8
      .byte    50    ,    9
603 .byte    60    ,   10
      .byte    80    ,   12
605 .byte    90    ,   16
      .byte   100    ,   24
607 .byte   110    ,   36
      .byte  0xFF, 0xFF ;FIN DE TABLA
609
      ; TABLA PARA MODO FRIO
611 ;Formato: [Tension minima, Dif_Temperatura*2]
      .org      0x0100
613 .byte    0      ,    0
      .byte    1      ,    0
615 .byte   40      ,    1
      .byte   60      ,    2
617 .byte   80      ,    4
      .byte   90      ,   16
619 .byte  100      ,   18
      .byte  110      ,   20
621 .byte  115      ,   22
      .byte  0xFF, 0xFF ;FIN DE TABLA
623
      .end
```

## Referencias

- [1] <http://www.embrlabs.com/>
- [2] <https://youtu.be/sDZHITVfYrI>
- [3] <https://youtu.be/kvUMCip-r4A>
- [4] [http://www.atmel.com/images/atmel-2486-8-bit-avr-microcontroller-atmega8\\_1\\_datasheet.pdf](http://www.atmel.com/images/atmel-2486-8-bit-avr-microcontroller-atmega8_1_datasheet.pdf)