

Índice

1. Objetivos	2
1.1. Diagrama de Bloques	2
1.2. Diagrama de Flujo	3
1.3. Especificaciones	4
1.3.1. Componentes	4
2. Diseño	5
2.1. Esquemático	5
2.2. Componentes	6
2.3. Regulador de Corriente	6
2.4. Medición de la temperatura	7
2.5. Interruptor	8
3. Especificaciones del microcontrolador	8
3.1. Microcontrolador	8
3.2. Configuraciones	9
3.2.1. Low Fuse	9
3.2.2. UCSRC	9
3.2.3. UBRRL	9
3.2.4. TCCR2	9
3.2.5. ADCSRA	10
4. Software	10
4.1. Protocolo puerto serie	10
5. Conclusiones	10
A. Código Proyecto	12
A.1. Pulsera.S	12
A.2. Makefile	23
B. Presupuesto	24
C. Datasheets	24
D. Referencias	35

1. Objetivos

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

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].

1.1. Diagrama de Bloques

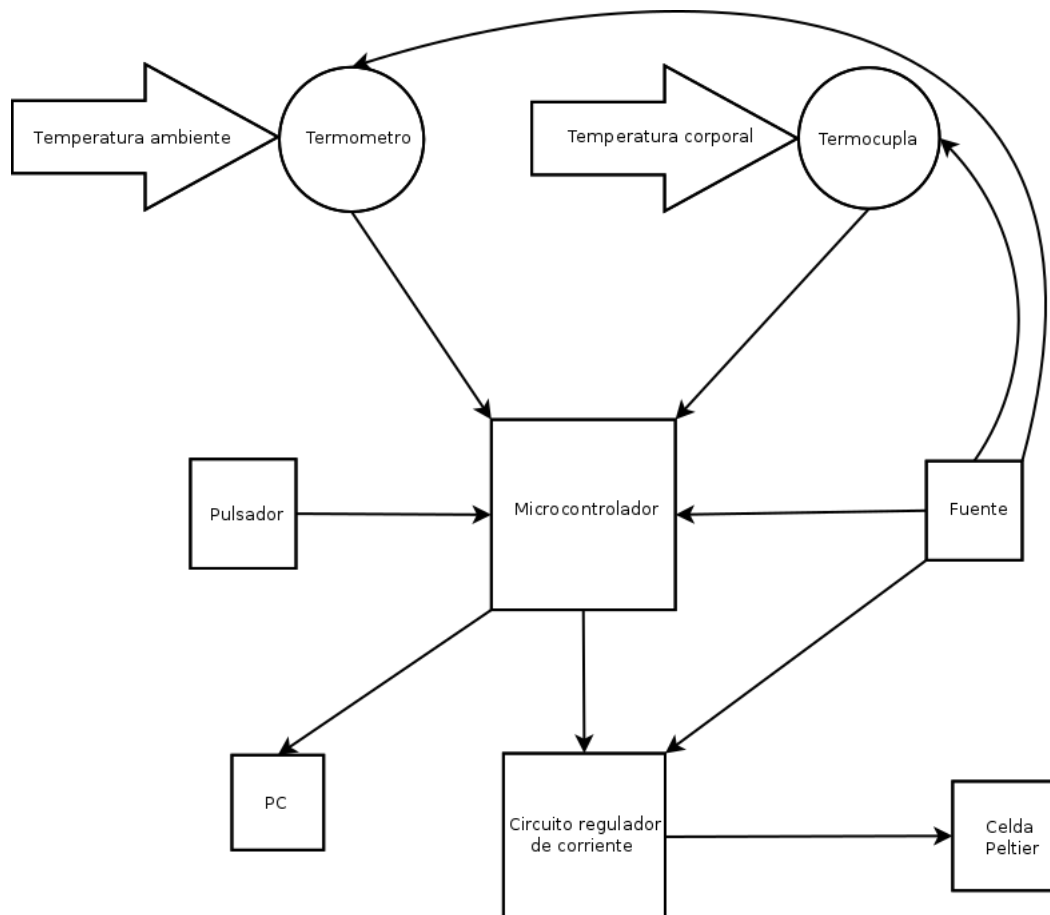


Figura 1: Diagrama de bloques

1.2. Diagrama de Flujo

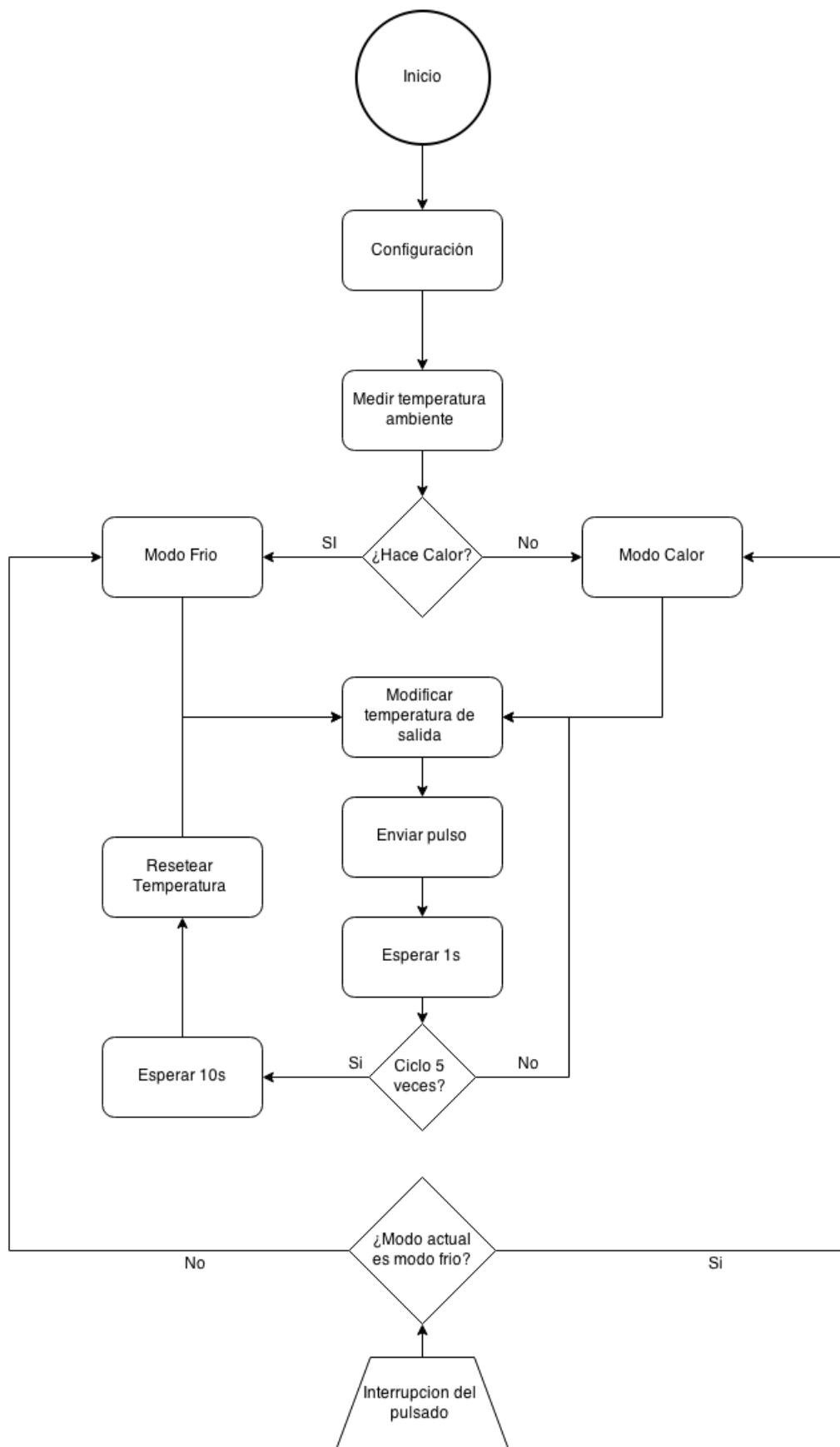


Figura 2: Diagrama de flujo del proceso

1.3. Especificaciones

El dispositivo utilizará una celda Peltier para enviar pulsos de calor o frío. De forma que se logre una diferencia de temperatura mayor a $0,4^{\circ}\text{C}/\text{seg.}$ 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 en base a la corriente que circulará por la celda Peltier.

1.3.1. Componentes

Deberá contar con los siguientes componentes:

- Celda Peltier: Generará los pulsos de calor en la muñeca del usuario.
- Circuito regulador de corriente: Regulará la corriente suministrada a la celda peltier.
- Disipador: La celda Peltier contará con un disipador para evitar fijar la temperatura de una de sus placas.
- Termistores: Contará con dos termistores. Uno para medir la temperatura ambiente y en base a esta decidir el modo de trabajo, frío o calor. El segundo termistor medirá la temperatura del disipador conectado a la celda Peltier para poder realizar una estimación de la temperatura de la celda.
- Salida de puerto serie: Servirá para poder monitorear en una computadora la temperatura de la placa.
- Fuente: Suministrará la corriente necesaria a la celda Peltier y proporcionará alimentación a todos los dispositivos utilizados.
- Interruptor: Para poder invertir el estado de trabajo, de frío a calor y viceversa.
- Controlador: Se utilizara un microcontrolador AVR. Es el encargado de obtener las temperaturas de los termistores para definir el modo de trabajo y autoregular la corriente de la celda Peltier mediante el circuito regulador de corriente.

2. Diseño

2.1. Esquemático

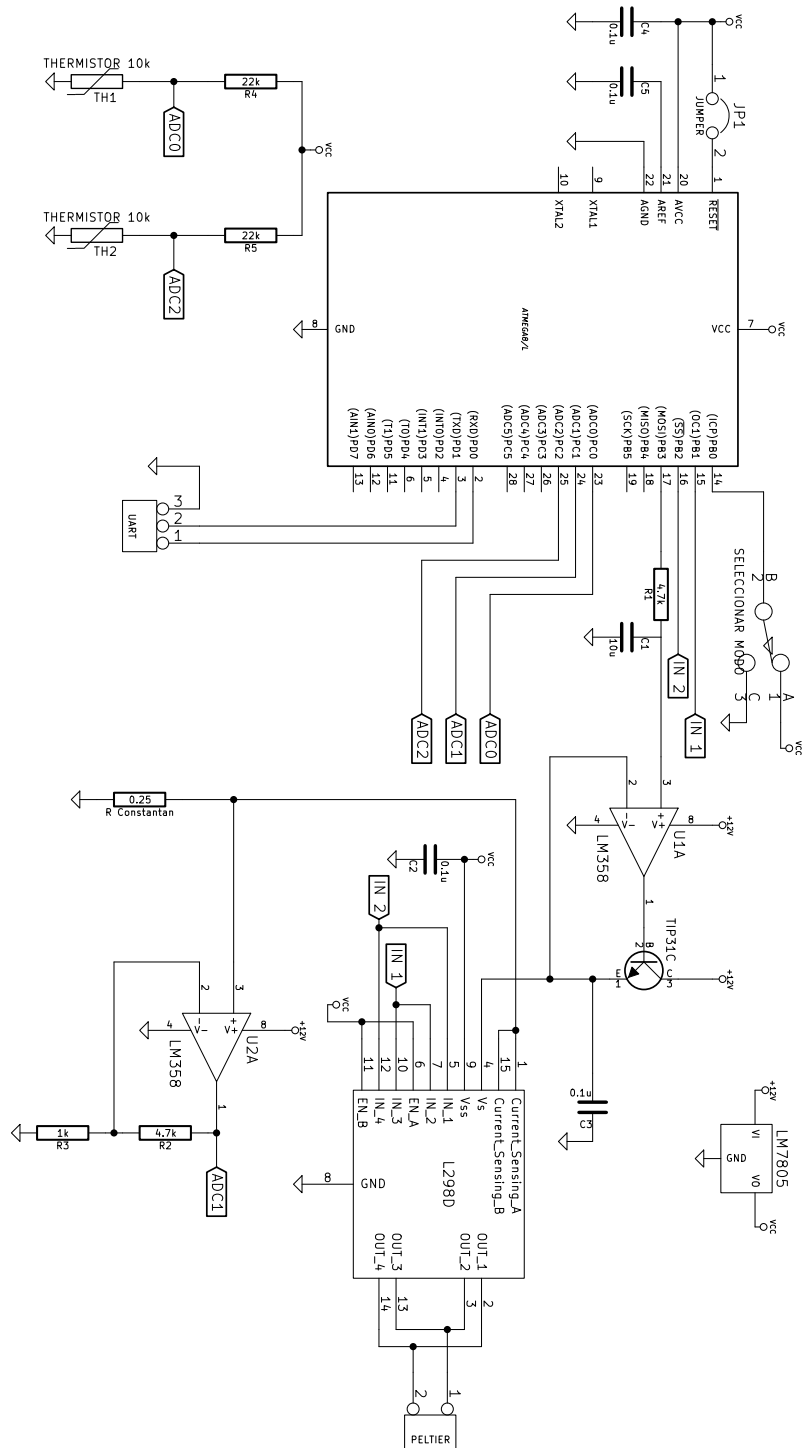


Figura 3: Diagrama esquemático

2.2. Componentes

- Celda Peltier: Celda peltier de 10 W y 15 mmx15 mm
- LM7805: Regulador de tensión para habilitar el puente H, alimentar el microcontrolador y suministrarle tensión constante a las resistencias conectadas en serie a los termistores.
- Interruptor: Interruptor para activar la inversión de la polaridad.
- Resistencias:
 - Dos resiststencias de 4,7 k Ω
 - Dos resiststencias de 22,0 k Ω
 - Una resistencia de 1,0 k Ω
- Capacitores:
 - 1 capacitor de 10 μ F para generar tensión constante del PWM recibido.
 - 2 capacitores de 0.1 μ F Conectados en paralelo a las alimentaciones del puente H, recomendados por el fabricante.
- LM358: Dos amplificadores operacionales. Uno para suministrar corriente a la base del NPN y el segundo para amplificar la tensión leída del constantán.
- TIP31C: Transistor de potencia NPN, utilizado para regular la corriente.
- L298D: Puente H utilizado para invertir la polaridad de la celda Peltier
- Constantán: alambre utilizado para sensar la corriente generada.
- Bateria: de 12 V y 2,9 Ah
- Pines:
 - 3 pines para el puerto serie.
 - 2 pines para el reseteo del microcontrolador.
 - 2 pines para conectar la celda peltier al circuito.
- Termistores: Dos termistores NTC de 10 k Ω

2.3. Regulador de Corriente

Se utilizó un regulador de corriente controlado por un PWM como se muestra en la figura 4

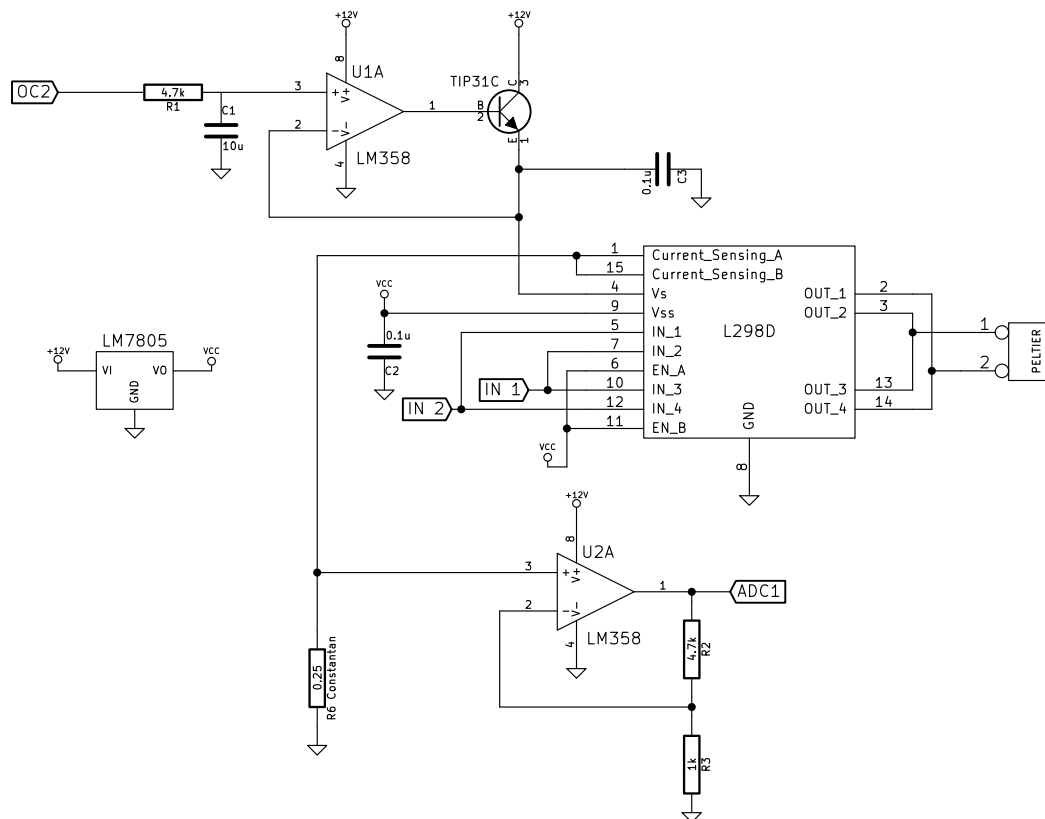


Figura 4: Regulador de Corriente

El circuito RC generará una tensión constante del PWM generado por el microcontrolador. Dicha tensión regulará la corriente suministrada a la base del NPN por el amplificador operacional, generando una corriente constante entre el colector y el emisor del transistor.

Se optó por utilizar un L298D para el puente H ya que cuenta en un mismo integrado dos puentes H que soportan 2 A de corriente. Conectados en paralelo como se muestra en la figura 4 se puede duplicar dicha corriente máxima para que soporte hasta 4 A de corriente.

Al final del circuito se sensorá la corriente generada mediante la tensión en el alambre constatan que es amplificada por el amplificador operacional. Para que la tensión de salida varíe entre 0 V y 2,56 V y sea leído por el microcontrolador.

Las resistencias del amplificador se obtuvieron considerando que para la corriente máxima registrada, la salida no supere los 2,56 V. Se registró una corriente máxima de 1,75 A y se midió una resistencia de 0,25 Ω .

La tensión de salida se obtiene mediante:

$$V_{ADC1} = R_{constantan} I_{MAX} \frac{R_3 + R_2}{R_3} \quad (1)$$

Luego fijando $R_3 = 1 \text{ k}\Omega$ y $R_2 = 4,7 \text{ k}\Omega$ se verificó que la tensión no supere los 2,56 V:

$$V_{ADC1} = 0,25 \Omega \cdot 1,75 \text{ A} \frac{1 \text{ k}\Omega + 4,7 \text{ k}\Omega}{1 \text{ k}\Omega} = 2,49 \text{ V}$$

2.4. Medición de la temperatura

Para medir la temperatura se utilizó un divisor resistivo utilizando termistores para medir su tensión y poder estimar la temperatura. Se obtuvieron las resistencias a conectar en serie con los termistores de

forma que la tensión máxima no supere los 2,56 V

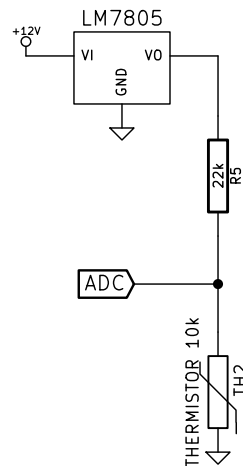


Figura 5: Divisor de tensión de los termistores

$$V_{termistor} = 5V \frac{R_{termistor}}{R_{termistor} + R_{serie}} \quad (2)$$

Finalmente se eligió una resistencia de $R_{serie} = 22k\Omega$ para R_4 y R_5 verificando que la tensión en los termistores no supere la tensión de referencia del ADC del microcontrolador para la resistencia máxima registrada en los termistores a $R_{0^{\circ}C} = 15k\Omega$:

$$V_{termistor} = 5V \frac{15k\Omega}{15k\Omega + 22k\Omega} = 2V$$

2.5. Interruptor

La salida del interruptor es recibida por el microcontrolador, para en el siguiente ciclo invertir la polaridad mediante de la celda Peltier y evitar un cambio brusco de corriente en la celda Peltier.

El interruptor invertirá el modo de trabajo actual. En caso de encontrarse el dispositivo a temperaturas ambiente menores a $25^{\circ}C$ automáticamente entrará en modo calor. En caso de estar el interruptor habilitado entrará en modo frío. Caso contrario ocurre para temperaturas ambiente mayores a $25^{\circ}C$ por defecto entrará en modo frío, con el interruptor habilitado entrará en modo calor.

3. Especificaciones del microcontrolador

3.1. Microcontrolador

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

3.2. Configuraciones

3.2.1. Low Fuse

Se configuro este registro para que el clock del microcontrolador sea de 8MHz.

7	6	5	4	3	2	1	0
1	1	1	0	0	1	0	0
BODLVL	BODEN	SUT1	SUT0	CKSEL3	CKSEL2	CKSEL1	CKSEL0

Solo se modificaron los valores de CKSEL, el resto de los bits fue dejado en la configuración que venia de fabrica.

3.2.2. UCSRC

Se configuro este registro para setear que el puerto serie envíe datos de 8bits, con un bit de stop y sin bit de paridad

7	6	5	4	3	2	1	0
1	0	0	0	0	1	1	0
URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL

Configuracion segun el bit

Bit 7: Selecciona entre acceder el UCSRC seteando el bit en 1 o acceder al UBRRL seteando el bit en 0.

Bit 6: Modo Asincrónico

Bit 5 y 4: Sin bit de paridad

Bit 3: Un bit de STOP

Bit 2 y 1: Datos de 8 bits

Bit 0: 0 Por modo Asincrónico

3.2.3. UBRRL

Se configuro este registro para setear el Baud Rate del puerto serie a 38,4Mhz

7	6	5	4	3	2	1	0
0	0	0	0	1	1	0	0
UBRR[7:0]							

3.2.4. TCCR2

Se configuro este registro para setear el modo de funcionamiento del contador 2.

7	6	5	4	3	2	1	0
0	1	1	1	0	0	0	1
FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20

Configuracion segun el bit

Bit 7: 0, por ser modo PWM

Bit 6 y 3: Modo PWM, Phase Correct

Bit 5 y 4: Modo set on match en subida y clear on match en bajada

Bit 2 - 0: Sin prescaler

3.2.5. ADCSRA

Se configuro este registro para el conversor analógico digital

7	6	5	4	3	2	1	0
1	1	0	0	1	1	1	1
ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0

Configuracion segun el bit

Bit 7: Habilita ADC

Bit 6: Inicializa la conversión

Bit 5: Desabilita el free running

Bit 4: Resetea el flag de interrupción pasandole 1

Bit 3: Habilita la interrupción

Bit 2 a 0: Factor de división: 128 para 111

4. Software

4.1. Protocolo puerto serie

Para la comunicación desde el puerto serie se utilizo un protocolo con el siguiente formato:

- **Primer bloque:** Un byte con un carácter *ASCII* alfanumerico identificador el dato a mandar.
- **Siguientes bloque:** Uno o mas bytes con el dato a enviar. El receptor se debe encargar de determinar el largo de este en base a lo recibido en el primer bloque.

En particular para este proyecto, se utilizaron datos de un byte¹ y se enviaron con la configuración detallada en la sección 3.2.2 (Pagina: 9), con lo cual los paquetes enviados por puerto respetan el siguiente formato:

1-bit	8-bits	1-bit	1-bit	8-bits	1-bit
START	Tipo de dato	STOP	START	Dato enviado	STOP

5. Conclusiones

En este punto se hará una autocrítica acerca de los errores y aciertos logrados . También se podrán hacer sugerencias sobre futuras mejoras en el proyecto y si consideran que otro grupo lo puede continuar. Incluir en este punto las diferencias, si las hubiere, entre las especificaciones originales previstas en el anteproyecto y las finalmente alcanzadas.

Las celdas Peltier son dispositivos muy delicados, por lo que requerirá estar bien protegidas a posibles impactos, ya que el mínimo impacto los daña permanentemente. No se logró encontrar una relación simple entre la corriente de la celda y su diferencia de temperatura, ya que esta no solo depende de la corriente sino también de la temperatura inicial. Por los datos medidos para realizar las tablas se llegó a la conclusión de que la corriente fija una velocidad a la que la temperatura variará durante intervalos cortos.

Fue mas fácil regular el modo frío ya que al iniciar el standby la temperatura regresaba rápidamente a su valor inicial. En cambio para el modo calor, al iniciar el standby la celda peltier no llega a enfriarse hasta obtener su valor inicial luego del standby, por lo que en cada ciclo irá acumulando temperatura hasta llegar a un valor máximo que depende de la corriente máxima que reciba.

Por las mismas razones mencionadas sobre el comportamiento de la celda Peltier, deducir su temperatura en base a la temperatura del disipador y la corriente que circula por la celda devuelve datos poco precisos, lo que se podría calcular es la diferencia de temperatura entre cada pulso con el inicio del ciclo. Para medir la temperatura del Peltier se deberá agregar una placa que transmita el calor del Peltier para aumentar su superficie y poder medir su temperatura directamente con una termocupla, ya que detectan mas rápido los cambios bruscos de temperatura, a diferencia de los termistores.

¹Esto fue en parte una consecuencia de la elección del modelo del microcontrolador. Por otro lado, tampoco eran necesarios datos mas grandes.

Finalmente, para lograr las condiciones especificadas por el proyecto no es necesario generar corrientes altas. Con una corriente máxima de 500 mA bien reguladas se hubiese logrado el objetivo del producto y se habría reducido su tamaño debido a que no requiere una batería de gran tamaño.

A. Codigo Proyecto

A.1. Pulsera.S

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

3  .section .data
   .org 0x000

5
6  Temp_Ambiente: .byte 0
7  Temp_Disipador: .byte 0
8  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
20 .section .text
21 .org 0x0
   .global main
23 rjmp main

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

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

31
32 ;r20: Contador
33 #define Contador r20

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

37
38 #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

52 #define Eeprom_Inicio_Calor 0x64
   #define Eeprom_Inicio_Frio 0x94

55
```

```

57 ;Saltea el vector de interrupcion
.org 0x0020
59 main:

61 STACK_Init:
    ldi    Reg_Temporal,    low(RAMEND)
63    out    _SER_IO_ADDR(SPL), Reg_Temporal
    ldi    Reg_Temporal,    high(RAMEND)
65    out    _SER_IO_ADDR(SPH), Reg_Temporal

67    rcall  PWM_Init
    rcall  PUENTE_H_Init
69    rcall  USART_Init

71 LOOP:
    clr    Contador

73
    rcall  GET_MODE                ;obtengo el modo de operacion

75
    ldi    r29,    PWM_inicial    ;PWM
77    sts    PWM,    r29
    rcall  RESET_PWM                ; Inicializa el PWM con el valor inicial

79 REDUCIR_LOOP:

81    mov    Reg_Temporal,    Contador

83
    ldi    r27,    10
85    rcall  ESPERA                ; Espera de 1 seg, mientras espera regul
    rcall  AUMENTAR_PULSO        ; Modifica el PWM para el siguiente segu

87

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

93    rjmp   REDUCIR_LOOP

95    cbi    _SER_IO_ADDR(DDRB), 3    ;(OC2) para salida
    sbi    _SER_IO_ADDR(DDRB), 3    ;(OC2) para salida
97    cbi    _SER_IO_ADDR(PORTB),3

99    rcall  STANDBY
    rjmp   LOOP

101
;-----Funciones-----
103 ; Reinicia el PWM al valor inicial de cada ciclo y carga las tablas a ser
; recorridas dependiendo del modo de operacion
105 RESET_PWM:
    clr    r17                    ;8 bits mas significativos de la direcci
107    lds    r31,    Modo_Operacion
    cpi    r31,    0
109    breq   TABLA_FRIO

TABLA_CALOR:
111    ldi    Reg_Temporal,    Eeprom_Inicio_Calor
    rjmp   CARGAR_TABLA

113 TABLA_FRIO:
    ldi    Reg_Temporal,    Eeprom_Inicio_Frio
115 CARGAR_TABLA:
    inc    Reg_Temporal

```

```

117     inc      Reg_Temporal
118     sts      Iterador ,          Reg_Temporal
119     rcall    LEER_EEPROM
120     sts      Tension_min ,      r25
121
122     inc      Reg_Temporal
123     inc      Reg_Temporal
124     rcall    LEER_EEPROM
125     sts      Tension_max ,      r25
126
127     lds      r29 ,          PWM
128     out      _SFR_IO_ADDR(OCR2) , r29
129
130     ret
131 ;-----
132 ; espera r27 * 100mseg. Cada 100 mseg hace un muestre de datos y regula
133 ; el PWM
134 ESPERA:
135     rcall    SET_PWM
136     rcall    TRANSMITIR_DATOS
137     rcall    DEMORA
138
139     dec      r27
140     clr      Reg_Temporal
141     cpse     r27 ,          Reg_Temporal
142     rjmp     ESPERA
143     ret
144
145 ;-----
146 ; Lee de eeprom en la direccion indicada en los registros r17 para los 8 bits
147 ; mas significativos y r16 para los 8 bits menos significativos. Guarda el contenido
148 ; en el registro r25
149 LEER_EEPROM:
150     ;Espera hasta que la ultima escritura este terminada
151     sbic     _SFR_IO_ADDR(EECR) , EWE
152     rjmp     LEER_EEPROM
153
154     ;r17 elige la tabla
155     out      _SFR_IO_ADDR(EEARH) , r17
156     ;r16 el campo
157     out      _SFR_IO_ADDR(EEARL) , r16
158
159     ;habilita el modo lectura
160     sbi      _SFR_IO_ADDR(EECR) , EERE
161     ;guarda el contenido de la ódireccin antes cargada en r25
162     in       r25 ,          _SFR_IO_ADDR(EEDR)
163     ret
164
165 ;-----
166 ; Inicializa el puente H seteando los pines 1 y 2 del puerto B como salida.
167 PUENTE_H_Init:
168     sbi      _SFR_IO_ADDR(DDRB) , 2
169     sbi      _SFR_IO_ADDR(DDRB) , 1
170     ret
171
172 MODO_FRIO:
173     sbi      _SFR_IO_ADDR(PORTB) , 1
174     cbi      _SFR_IO_ADDR(PORTB) , 2
175     ret

```

```

177 MODO_CALOR:
    cbi    _SFR_IO_ADDR(PORTB), 1
179    sbi    _SFR_IO_ADDR(PORTB), 2
    ret

181 ;-----

183 TRANSMITIR_DATOS:
    rcall  LEER_AMBIENTE
185    rcall  LEER_DISIPADOR
    rcall  LEER_PELTIER

187
    ;Envio la iteracion
189    ldi    Reg_Temporal, Dato_Num_Iteracion    ; tipo de dato a mandar
    rcall  USART_Transmit
191    mov    Reg_Temporal, Contador
    rcall  USART_Transmit

193
    ldi    Reg_Temporal, Dato_Tempe_Ambiente    ; tipo de dato a mandar
195    rcall  USART_Transmit
    lds    Reg_Temporal, Temp_Ambiente
197    rcall  USART_Transmit

199
    ldi    Reg_Temporal, Dato_Tension_Salida    ; tipo de dato a mandar
201    rcall  USART_Transmit
    lds    Reg_Temporal, Tension_Salida
203    rcall  USART_Transmit

205
    ldi    Reg_Temporal, Dato_Tempe_Disipador    ; tipo de dato a mandar
207    rcall  USART_Transmit
    lds    Reg_Temporal, Temp_Disipador
    rcall  USART_Transmit

209
    ldi    Reg_Temporal, Dato_Tempe_Peltier    ; tipo de dato a mandar
211    rcall  USART_Transmit
    lds    Reg_Temporal, Temp_Peltier
213    rcall  USART_Transmit

215
    ldi    Reg_Temporal, Dato_PWM    ; tipo de dato a mandar
217    rcall  USART_Transmit
    lds    Reg_Temporal, PWM
    rcall  USART_Transmit

219
    ldi    Reg_Temporal, Dato_min    ; tipo de dato a mandar
221    rcall  USART_Transmit
    lds    Reg_Temporal, Tension_min
223    rcall  USART_Transmit

225
    ldi    Reg_Temporal, Dato_max    ; tipo de dato a mandar
227    rcall  USART_Transmit
    lds    Reg_Temporal, Tension_max
    rcall  USART_Transmit

229    ret

231 ;-----
LEER_AMBIENTE:
233    ldi    Reg_Temporal, 0b11000000    ; canal 0 temperatura ambiente
    rcall  READ_ADC    ; leer tension del peltier
235    rcall  TRADUCIR_TERMISTOR
    sts    Temp_Ambiente, Reg_Temporal

```

```

237      ret

239  LEER_DISIPADOR:
      ldi      Reg_Temporal,      0b11000010      ; canal 2 temperatura disipador
241      rcall   READ_ADC          ; leer tension del termistor
      rcall   TRADUCIR_TERMISTOR
243      sts     Temp_Disipador,    Reg_Temporal
      ret

245  LEER_PELTIER:
247      ldi      Reg_Temporal,      0b11000001      ; canal 1 tension peltier
      rcall   READ_ADC          ; leer tension del peltier
249      sts     Tension_Salida,    Reg_Temporal
      rcall   TRADUCIR_PELTIER
251      ret

253  ;-----
253  ; Set PWM
255  ; Setea el pum del pin OC2 con el tiempo en bajo pasado como parametro
255  ; Reg_Temporal: tiempo en bajo a asignar
257
257  SET_PWM:
259      lds      r30, PWM
      lds      Reg_Temporal,      Modo_Standby
261      cpi      Reg_Temporal,      1
      ; Si esta en modo standby no realiza cambios
263      breq     APLICAR_CAMBIO

265      lds      r29, Tension_Salida
      lds      r17, Tension_min
267      lds      r18, Tension_max
      ldi      r26, Incremento_Regulacion ; valor a ser restado o sumado
269
      cp       r29, r17
271      brlo     AUMENTAR

273      cp       r29, r18
      brlo     APLICAR_CAMBIO

275
275  DISMINUIR:
277      cpi      r30, 100
      breq     APLICAR_CAMBIO
279      add      r30, r26
      rjmp     APLICAR_CAMBIO
281  AUMENTAR:
      cpi      r30, 0
283      breq     APLICAR_CAMBIO
      sub      r30, r26
285  APLICAR_CAMBIO:
      sts      PWM, r30
287      out     _SFR_IO_ADDR(OCR2), r30
      ret

289  ;-----
289  ; Aumenta el pulso de calor/frio del segundo que recién empieza,
291  ; Carga la tension minima y maxima leidas de eeprom de la iteracion
291  ; correspondiente.
293  AUMENTAR_PULSO:
      clr      r17

295
      lds      Reg_Temporal,      Iterador

```



```

297     inc      Reg_Temporal
298     inc      Reg_Temporal
299     sts      Iterador ,          Reg_Temporal

301     rcall    LEER_EEPROM
302     sts      Tension_min ,      r25
303     inc      Reg_Temporal
304     inc      Reg_Temporal
305     rcall    LEER_EEPROM
306     sts      Tension_max ,      r25

307
308     lds      r31 ,              Modo_Operacion
309     cpi      r31 ,              0
310     breq     AUMENTO_FRIO
311 AUMENTO_CALOR:
312     ldi      r26 ,              Incremento_Pulso_Calor
313     rcall    AUMENTAR
314     ret
315 AUMENTO_FRIO:
316     ldi      r26 ,              Incremento_Pulso_Frio
317     rcall    AUMENTAR
318     ret
319
320 ;-----
321 ;Espera durante 100mseg
DEMORA:
322     ldi      Reg_Temporal ,      0xCF          ; Valores de los que empieza a contar
323     out      _SFR_IO_ADDR(TCNT1H) , Reg_Temporal
324     ldi      Reg_Temporal ,      0x2B
325     out      _SFR_IO_ADDR(TCNT1L) , Reg_Temporal
326     ldi      Reg_Temporal ,      4              ; 0000 0100 habilita poner en 1
327     out      _SFR_IO_ADDR(TIFR) , Reg_Temporal ; Reinicia el overflow flag
328     out      _SFR_IO_ADDR(TIMSK) , Reg_Temporal ; bit 2 en 1 habilita la interrupcion por timer
329     ldi      Reg_Temporal ,      0b00000011    ; velocidad: clk/64
330     out      _SFR_IO_ADDR(TCCR1B) , Reg_Temporal
331
332 DEMORA_LOOP:
333     in        Reg_Temporal ,      _SFR_IO_ADDR(TIFR)
334     sbrs     Reg_Temporal ,      2
335     rjmp     DEMORA_LOOP
336
337     ldi      Reg_Temporal ,      1
338     out      _SFR_IO_ADDR(TIFR) , Reg_Temporal ; Se limpia TIRF
339     clr      Reg_Temporal          ; finalizo contador
340     out      _SFR_IO_ADDR(TIFR) , Reg_Temporal
341     out      _SFR_IO_ADDR(TCCR1B) , Reg_Temporal
342
343     ret
344
345
346 ;-----
347 ;Standby
348 ;Espera durante 10 segundos, setea el PWM en 255 para no generar corriente.
STANDBY:
349     ldi      Reg_Temporal ,      1
350     sts      Modo_Standby ,      Reg_Temporal
351     ldi      r29 ,              255          ;PWM
352     sts      PWM ,              r29
353     rcall    SET_PWM
354     ldi      r27 ,              100

```

```

357     rcll    ESPERA
358     clr     Reg_Temporal
359     sts     Modo_Standby,      Reg_Temporal
360     ret
361
362 ;-----
363 ;Transmit
364 ;Transmite por el puerto paralelo el dato pasado como parametro
365 ;Reg_Temporal: valor a transmitir
366
367 USART_Transmit:
368     sbis     _SFR_IO_ADDR(UCSRA), UDRE          ;Espero a que se libere el UDRE
369     rjmp     USART_Transmit
370
371     out      _SFR_IO_ADDR(UDR),  Reg_Temporal
372     ret
373
374 ;-----
375 ;Uart init
376 ;Inicializa el USART para poder enviar datos
377
378 USART_Init:
379     ldi      Reg_Temporal,      (1<<TXEN)          ;enable
380     out      _SFR_IO_ADDR(UCSRB), Reg_Temporal
381
382     ldi      Reg_Temporal,      (1<<URSEL)|(3<<UCSZ0) ;8bits, 1bit de stop, sin bit de
383     out      _SFR_IO_ADDR(UCSRC), Reg_Temporal
384
385     ldi      Reg_Temporal,      0xC              ;Baud 38400 (Clock de 8Mhz)
386     out      _SFR_IO_ADDR(UBRR1), Reg_Temporal
387
388     ret
389
390 ;-----
391 ;Read adc
392 ;Lee un dato del conversor adc y lo devuelve
393 ;Reg_Temporal: canal del cual leer
394 ;Reg_Temporal: valor leído devuelto
395
396 READ_ADC:
397
398     out      _SFR_IO_ADDR(ADMUX), Reg_Temporal    ; Canal a ser leído, 7 y 6 en 1 para v
399     ldi      Reg_Temporal,      0b11001111        ; habilita ADC, inicia la conversion, a
400     out      _SFR_IO_ADDR(ADCSRA), Reg_Temporal    ; buts 2:0 Factor de division 128
401
402 WAIT_ADC:
403     ;Espera a que finalice la lectura
404     in       Reg_Temporal,      _SFR_IO_ADDR(ADCSRA)
405     sbrs     Reg_Temporal,      4
406     rjmp     WAIT_ADC
407
408     sbi      _SFR_IO_ADDR(ADCSRA), 4              ; Reinicia el interrupt flag
409
410     in       Reg_Temporal,      _SFR_IO_ADDR(ADCL)
411     in       r17,               _SFR_IO_ADDR(ADCH)
412     lsl      r17
413     ror      Reg_Temporal
414     lsl      r17
415     ror      Reg_Temporal

```

```

417     ret

419 ;-----
419 ;PWM init
421 ;Inicializa los puertos de salida del pum
PWM_Init:
423     sbi     _SFR_IO_ADDR(DDRB) , 3           ;(OC2) para salida
424     ldi     Reg_Temporal , 0x71             ;(01110001) Phase correct,
425     out     _SFR_IO_ADDR(TCCR2) ,Reg_Temporal ; no pre escalar, clear on match
426     ret

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

433 TRADUCIR_TERMISTOR:
434     mov     r18 , Reg_Temporal
435     ldi     r17 , 0                         ;tabla termistor
436     ldi     Reg_Temporal , 0                 ;indice

437 LOOP_BUSQUEDA_TERM:
438     rcall   LEER_EEPROM
439     cpi     r25 , 0
440     breq    FIN_TABLA
441     cp      r18 , r25                       ;leído vs valor tabla
442     brsh    END_TERMISTOR
443     inc     Reg_Temporal
444     inc     Reg_Temporal
445     rjmp    LOOP_BUSQUEDA_TERM

447 END_TERMISTOR:
448     inc     Reg_Temporal
449     rcall   LEER_EEPROM
450     mov     Reg_Temporal , r25
451     ret

453 FIN_TABLA:
454     dec     Reg_Temporal
455     dec     Reg_Temporal
456     rjmp    END_TERMISTOR

457 ;-----
459 ;Traducir peltier
459 ;Convierte el valor recibido por parametro en su temperatura equivalente
461 ;entrada: Temp_Disipador, Tension_Salida, Modo_Operacion
461 ;salida: Temp_Peltier

463 TRADUCIR_PELTIER:
464     lds     r18 , Tension_Salida
465     ldi     r17 , 0                         ;tabla termistor
466     lds     r31 , Modo_Operacion
467     cpi     r31 , 0
468     breq    TABLA_FRIO_PELTIER
469 TABLA_CALOR_PELTIER:
470     ldi     Reg_Temporal , Eeprom_Inicio_Calor
471     rjmp    LOOP_BUSQUEDA_TERM_PELTIER
473 TABLA_FRIO_PELTIER:
474     ldi     Reg_Temporal , Eeprom_Inicio_Frio
475     rjmp    LOOP_BUSQUEDA_TERM_PELTIER

```

```

477 LOOP_BUSQUEDA_TERM_PELTIER:
    rcall    LEER_EEPROM
479    cpi     r25,          0xFF
    breq     FIN_TABLA_PELTIER
481    cp      r25,          r18          ; valor tabla vs leído
    brsh     END_PELTIER
483    inc     Reg_Temporal
    inc     Reg_Temporal
485    rjmp    LOOP_BUSQUEDA_TERM_PELTIER
END_PELTIER:
487    inc     Reg_Temporal
    rcall    LEER_EEPROM
489    lds     r17,          Temp_Disipador
    lds     r18,          Modo_Operacion
491
    mov      Reg_Temporal,    r25
493
    sbrc     r18,          0
495    rjmp    CALCULO_CALOR
    rjmp    CALCULO_FRIO
497
CALCULO_CALOR:
499    add     Reg_Temporal,    r17
    sts     Temp_Peltier,    Reg_Temporal
501    ret

503 CALCULO_FRIO:
    sub     r17,          Reg_Temporal
505    mov     Reg_Temporal,    r17
    sts     Temp_Peltier,    Reg_Temporal
507    ret

509 FIN_TABLA_PELTIER:
    dec     Reg_Temporal
511    dec     Reg_Temporal
    rjmp    END_PELTIER
513
;-----
515 ;Get mode
;Devuelve el valor del modo en el cual se ejecuta para la temperatura Reg_Temporal
517 ;Reg_Temporal: temperatura leida
;Reg_Temporal: valor leído devuelto
519 GET_MODE:
    ldi     Reg_Temporal,    0b11000000    ; canal 0 temperatura ambiente
521    rcall    READ_ADC
    rcall    TRADUCIR_TERMISTOR    ; obtengo la temperatura
523    cpi     Reg_Temporal,    50          ; temperatura*2 arbitraria para el modo de
    brsh     COLD_MODE
525
HOT_MODE:
527    sbic    _SFR_IO_ADDR(PINB), 0          ; Si el pin esta en 1 cambia de modo
    rjmp    SET_COLD
529
SET_HOT:
531    cbi     _SFR_IO_ADDR(PORTB), 1
    sbi     _SFR_IO_ADDR(PORTB), 2
533
    ldi     Reg_Temporal,    1          ; 1 es modo calor
535    sts     Modo_Operacion,    Reg_Temporal    ; Modo_Operacion modo de operacion iniciado

```

```
537         ret

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

543 SET_COLD:
        sbi     _SFR_IO_ADDR(PORTB) , 1
545         cbi     _SFR_IO_ADDR(PORTB) , 2

547         ldi     Reg_Temporal , 0           ; 0 es el modo frio
        sts     Modo_Operacion , Reg_Temporal ; Modo_Operacion modo de operacion inicia
549
        ret

551
        .section .eeprom
553 .org 0x0000

555 ; Tablas de conversion
        ; Formato: [(Tension medida, Temperatura*2),...]

557 .byte 204 , 30
        .byte 204 , 31
559 .byte 204 , 32
        .byte 204 , 33
561 .byte 204 , 34
        .byte 204 , 35
563 .byte 204 , 36
        .byte 204 , 37
565 .byte 204 , 38
        .byte 201 , 39
567 .byte 199 , 40
        .byte 197 , 41
569 .byte 195 , 42
        .byte 194 , 43
571 .byte 193 , 44
        .byte 192 , 45
573 .byte 190 , 46
        .byte 188 , 47
575 .byte 186 , 48
        .byte 184 , 49
577 .byte 182 , 50
        .byte 181 , 51
579 .byte 172 , 52
        .byte 169 , 53
581 .byte 166 , 54
        .byte 163 , 55
583 .byte 160 , 56
        .byte 159 , 57
585 .byte 158 , 58
        .byte 157 , 59
587 .byte 156 , 60
        .byte 155 , 62
589 .byte 154 , 64
        .byte 153 , 66
591 .byte 152 , 68
        .byte 151 , 70
593 .byte 147 , 72
        .byte 145 , 74
595 .byte 143 , 76
        .byte 141 , 78
```

```
597 .byte    140 ,    80
    .byte    0, 0 ;FIN DE TABLA
599
    ; TABLA PARA MODO CALOR
601 ;Formato: [Tension minima, Dif_Temperatura*2]
    .org      0x0064
603 .byte    0      ,    0
    .byte    1      ,    0
605 .byte    20     ,   10
    .byte    25     ,   15
607 .byte    30     ,   17
    .byte    35     ,   21
609 .byte    40     ,   23
    .byte    50     ,   24
611 .byte    55     ,   18
    .byte    58     ,   20
613 .byte    60     ,   24
    .byte    62     ,   28
615 .byte    70     ,    7
    .byte    0xFF, 0xFF ;FIN DE TABLA
617
    ; TABLA PARA MODO FRIO
619 ;Formato: [Tension minima, Dif_Temperatura*2]

621 .org      0x0094
    .byte    0      ,    0
623 .byte    1      ,    0
    .byte    30     ,   14
625 .byte    35     ,   16
    .byte    40     ,   18
627 .byte    45     ,   20
    .byte    50     ,   22
629 .byte    60     ,   30
    .byte    100    ,   36
631 .byte    110    ,   40
    .byte    115    ,   42
633 .byte    0xFF, 0xFF ;FIN DE TABLA

635 .end
```

A.2. Makefile

```
1  EXE = pulsera
   MICRO = atmega8
3
   MSG_EEPROM = Creando archivo para la EEPROM:
5  FORMAT = ihex

7
   all: $(EXE).hex $(EXE).eep
9
   clean:
11      rm -f *.hex *.o *.elf *.eep *.d *.syb

13  $(EXE).elf: $(EXE).S
      #avr-gcc -Wall -g -g2 -gstabs -O0 -fpack-struct -fshort-enums -funsigned-char -funsi
      -Wa,-as=$(EXE).syb
15      avr-gcc -Wall -Wextra -pedantic -g -mmcu=$(MICRO) $(EXE).S -o $(EXE).elf
      -Wa,-as=$(EXE).syb #O3

17  send: $(EXE).hex $(EXE).eep
      sudo avrdude -c usbtiny -p m8 -U flash:w:$(EXE).hex:i
19
   send_table: $(EXE).hex $(EXE).eep
21      sudo avrdude -c usbtiny -p m8 -U flash:w:$(EXE).hex:i -U eeprom:w:$(EXE).eep:i

23  $(EXE).hex: $(EXE).elf
      avr-objcopy -O ihex $(EXE).elf $(EXE).hex
25
      ##$(EXE).eep: $(EXE).elf
27      # avr-objcopy -j .eeprom --no-change-warnings --change-section-lma .eeprom=0 -O ihex $

29  show_size: $(EXE).elf
      avr-size --format=avr --mcu=atmega8 $(EXE).elf
31

33  $(EXE).eep: $(EXE).elf
      @echo
35      @echo $(MSG_EEPROM) $@
      -avr-objcopy -j .eeprom --set-section-flags=.eeprom="alloc,load" \
37      --change-section-lma .eeprom=0 -O $(FORMAT) $< $@
```

B. Presupuesto

Componente	Precio xU	Cantidad	Precio total
Celda Peltier 10 W 15 mmx15 mm	\$120.00	1	\$120.00
Atmega8-L	\$42.00	1	\$42.00
L298	\$47.00	1	\$47.00
LM358	\$8.00	1	\$8.00
LM7850	\$5.00	1	\$5.00
Disipador	\$12.00	3	\$36.00
Termistor	\$7.00	2	\$14.00
Resistencia	\$0.50	5	\$2.50
Capacitor	\$0.50	5	\$2.50
Bateria	\$160.00	1	\$160.00
Interruptor	\$5.00	1	\$5.00
Placa perforada	\$26.00	1	\$26.00
Adaptador UART/USB	\$152.00	1	\$152.00
Total			\$620.00

C. Datasheets

Features

- High-performance, Low-power Atmel® AVR® 8-bit Microcontroller
- Advanced RISC Architecture
 - 130 Powerful Instructions – Most Single-clock Cycle Execution
 - 32 × 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16MIPS Throughput at 16MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
 - 8Kbytes of In-System Self-programmable Flash program memory
 - 512Bytes EEPROM
 - 1Kbyte Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler, one Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Three PWM Channels
 - 8-channel ADC in TQFP and QFN/MLF package
 - Eight Channels 10-bit Accuracy
 - 6-channel ADC in PDIP package
 - Six Channels 10-bit Accuracy
 - Byte-oriented Two-wire Serial Interface
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-lead PDIP, 32-lead TQFP, and 32-pad QFN/MLF
- Operating Voltages
 - 2.7V - 5.5V (ATmega8L)
 - 4.5V - 5.5V (ATmega8)
- Speed Grades
 - 0 - 8MHz (ATmega8L)
 - 0 - 16MHz (ATmega8)
- Power Consumption at 4Mhz, 3V, 25°C
 - Active: 3.6mA
 - Idle Mode: 1.0mA
 - Power-down Mode: 0.5µA



**8-bit Atmel with
8KBytes In-
System
Programmable
Flash**

**ATmega8
ATmega8L**

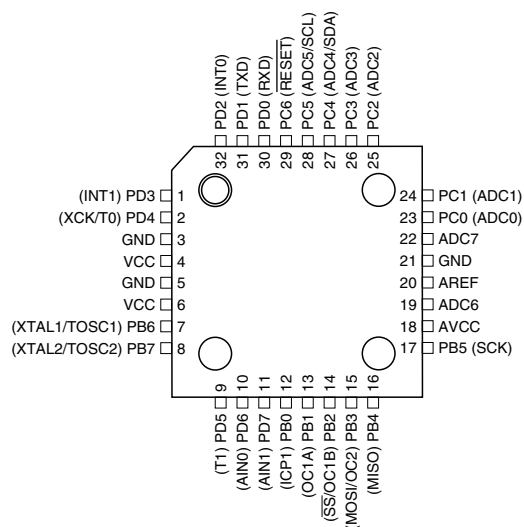
Rev.2486AA-AVR-02/2013

Pin Configurations

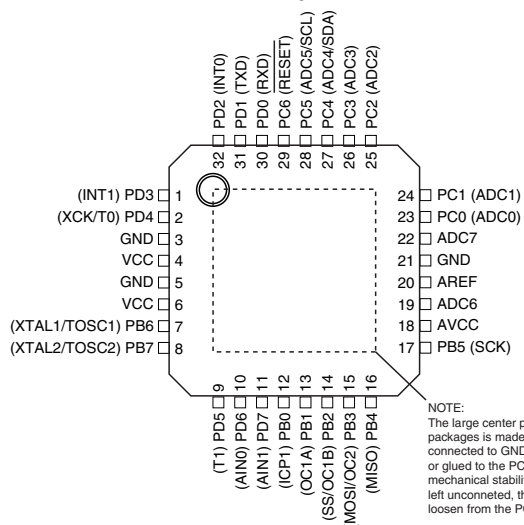
PDIP

(RESET) PC6	1	28	PC5 (ADC5/SCL)
(RXD) PD0	2	27	PC4 (ADC4/SDA)
(TXD) PD1	3	26	PC3 (ADC3)
(INT0) PD2	4	25	PC2 (ADC2)
(INT1) PD3	5	24	PC1 (ADC1)
(XCK/T0) PD4	6	23	PC0 (ADC0)
VCC	7	22	GND
GND	8	21	AREF
(XTAL1/TOSC1) PB6	9	20	AVCC
(XTAL2/TOSC2) PB7	10	19	PB5 (SCK)
(T1) PD5	11	18	PB4 (MISO)
(AIN0) PD6	12	17	PB3 (MOSI/OC2)
(AIN1) PD7	13	16	PB2 (SS/OC1B)
(ICP1) PB0	14	15	PB1 (OC1A)

TQFP Top View



MLF Top View



NOTE:
The large center pad underneath the MLF packages is made of metal and internally connected to GND. It should be soldered or glued to the PCB to ensure good mechanical stability. If the center pad is left unconnected, the package might loosen from the PCB.

NTC Thermistor : TTC05 Series



Φ5 mm Lead Type for Temperature Sensing/Compensation

■ Features

1. RoHS compliant
2. Halogen-Free (HF) series are available
3. Body size: Φ5mm
4. Radial lead resin coated
5. Operating temperature range: -30°C~+125°C
6. Wide resistance range
7. Cost effective
8. Agency recognition: UL / cUL / CSA / TUV / CQC



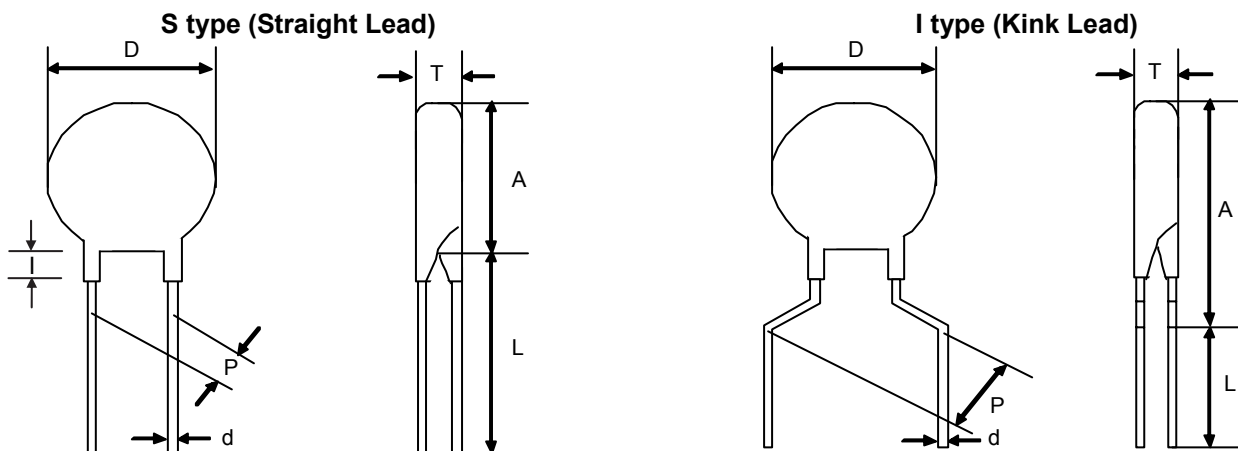
■ Recommended Applications

1. Home appliances
2. Automotive electronics
3. Computers
4. Switch mode power supplies
5. Adapters

■ Part Number Code

1	2	3	4	5	6	7	8	9	10	11	12	13	14
Product Type		Body Size		Zero Power Resistance at 25°C (R ₂₅)		Tolerance of R ₂₅		Appearance		Optional Suffix			
TTC	THINKING NTC Thermistor TTC Series	05	Φ5mm	R ₂₅ < 100Ω 005: 5Ω 015: 15Ω 050: 50Ω	R ₂₅ ≥ 100Ω 101: 100Ω 682: 6800Ω 474: 470000Ω	J ±5% K ±10% L ±15%	S Straight lead I Kink lead	Y RoHS Compliant E RoHS & HF Compliant					

■ Structure and Dimensions



(Unit: mm)

Type	D max.	P	d	I max.	A max.	L min.	T max.
S Type	6.5	3.5± 0.5	0.5±0.02	3	6.5	31	5
I Type	6.5	5± 0.5	0.5±0.02	—	10	29	5

NTC Thermistor : TTC05 Series



Φ5 mm Lead Type for Temperature Sensing/Compensation

■ Electrical Characteristics

Part No.	Zero Power Resistance at 25°C	Tolerance of R ₂₅	B _{25/50} Value	Max. Power Dissipation at 25°C	Dissipation Factor	Thermal Time Constant	Operating Temperature Range	Safety Approvals			
	R ₂₅ (Ω)	(±%)	(K)	P _{max} (mW)	δ(mW/°C)	τ (Sec.)	T _L ~T _U (°C)	UL /cUL	CSA	TUV	CQC
TTC05005□	5	10, 15	2400	450	Approx. 4.5	Approx. 20	-30~+125		√	√	√
TTC05010□	10		2800						√	√	√
TTC05015□	15		2800					√	√	√	√
TTC05020□	20		2800					√	√	√	√
TTC05025□	25		2900					√	√	√	√
TTC05045□	45		3100					√	√	√	√
TTC05050□	50		3100					√	√	√	√
TTC05060□	60		3100					√	√	√	√
TTC05085□	85		3200					√	√	√	√
TTC05090□	90		3200					√		√	√
TTC05101□	100		3200					√	√	√	√
TTC05121□	120		3300					√	√	√	√
TTC05151□	150		3300					√	√	√	√
TTC05201□	200		3500					√	√	√	√
TTC05221□	220		3500					√	√	√	√
TTC05251□	250		3500					√	√	√	√
TTC05301□	300		3800					√	√	√	√
TTC05471□	470		3500					√	√	√	√
TTC05501□	500		3700					√	√	√	√
TTC05681□	680		3800					√	√	√	√
TTC05701□	700		3800					√	√	√	√
TTC05102□	1000		3800					√	√	√	√
TTC05152□	1500		3950					√	√	√	√
TTC05202□	2000		4000					√	√	√	√
TTC05222□	2200		4000					√	√	√	√
TTC05252□	2500		4000					√	√	√	√
TTC05302□	3000		4000					√	√	√	√
TTC05332□	3300		4000					√	√	√	√
TTC05402□	4000		4000					√	√	√	√
TTC05472□	4700		4050					√	√	√	√
TTC05502□	5000		3950					√	√	√	√
TTC05602□	6000		4050					√	√	√	√
TTC05682□	6800		4050					√	√	√	√
TTC05802□	8000		4050					√	√	√	√
TTC05103□	10000	5, 10, 15	4050					√	√	√	√
TTC05123□	12000		4050					√	√	√	√
TTC05153□	15000		4150					√	√	√	√
TTC05203□	20000		4250					√	√	√	√
TTC05303□	30000		4250					√	√	√	√
TTC05473□	47000		4300					√	√	√	√
TTC05503□	50000		4300					√	√	√	√
TTC05104□	100000		4400					√	√	√	√
TTC05154□	150000		4500					√	√	√	√
TTC05204□	200000		4600					√	√	√	√
TTC05224□	220000		4600					√		√	√
TTC05474□	470000		4750					√		√	√

Note 1: □ = Tolerance of R₂₅

Note 2: UL/cUL File No: E138827

CSA File No: 97495

TUV File No: R 50050155

CQC File No: CQC05001011991 ; CQC05001011994

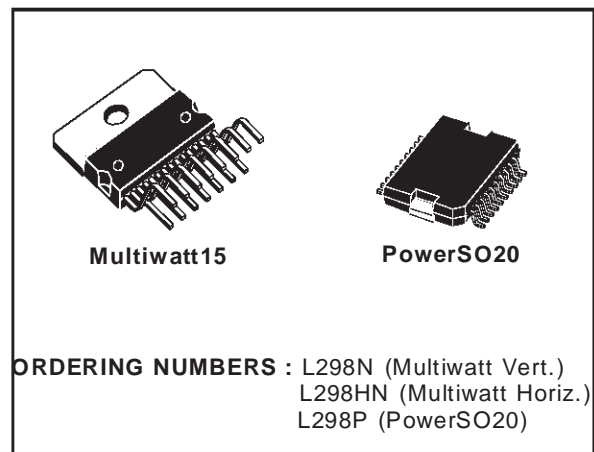
Note 3: Special specifications are available upon request.

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

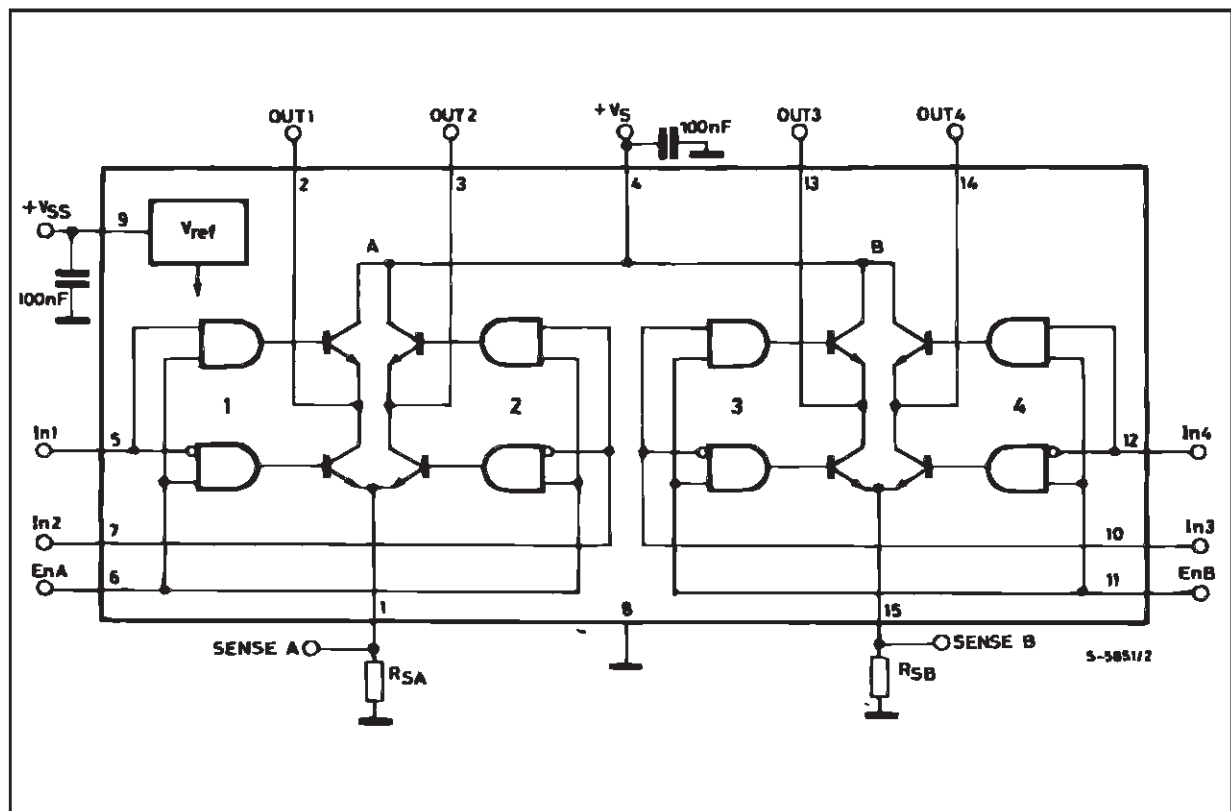
DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

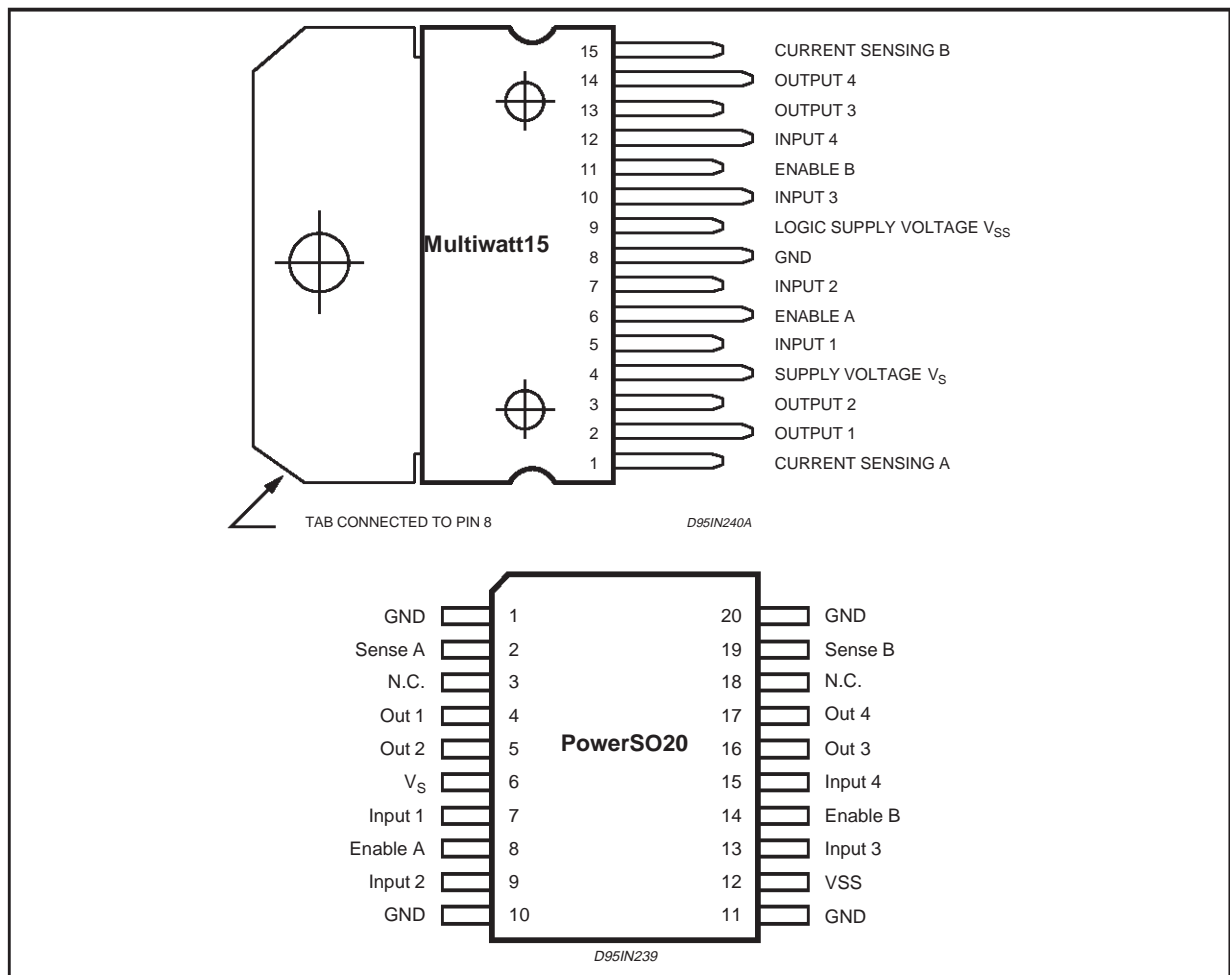
BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_O	Peak Output Current (each Channel)		
	– Non Repetitive ($t = 100\mu s$)	3	A
	– Repetitive (80% on -20% off; $t_{on} = 10ms$)	2.5	A
	– DC Operation	2	A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^\circ C$)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	$^\circ C$
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	$^\circ C$

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th\ j-case}$	Thermal Resistance Junction-case	Max.	–	3	$^\circ C/W$
$R_{th\ j-amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	$^\circ C/W$

(*) Mounted on aluminum substrate

LM258, LM358, LM358A, LM2904, LM2904A, LM2904V, NCV2904, NCV2904V

Single Supply Dual Operational Amplifiers

Utilizing the circuit designs perfected for Quad Operational Amplifiers, these dual operational amplifiers feature low power drain, a common mode input voltage range extending to ground/ V_{EE} , and single supply or split supply operation. The LM358 series is equivalent to one-half of an LM324.

These amplifiers have several distinct advantages over standard operational amplifier types in single supply applications. They can operate at supply voltages as low as 3.0 V or as high as 32 V, with quiescent currents about one-fifth of those associated with the MC1741 (on a per amplifier basis). The common mode input range includes the negative supply, thereby eliminating the necessity for external biasing components in many applications. The output voltage range also includes the negative power supply voltage.

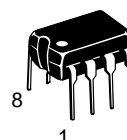
Features

- Short Circuit Protected Outputs
- True Differential Input Stage
- Single Supply Operation: 3.0 V to 32 V
- Low Input Bias Currents
- Internally Compensated
- Common Mode Range Extends to Negative Supply
- Single and Split Supply Operation
- ESD Clamps on the Inputs Increase Ruggedness of the Device without Affecting Operation
- NCV Prefix for Automotive and Other Applications Requiring Unique Site and Control Change Requirements; AEC-Q100 Qualified and PPAP Capable
- These Devices are Pb-Free, Halogen Free/BFR Free and are RoHS Compliant



ON Semiconductor®

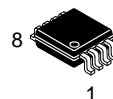
www.onsemi.com



PDIP-8
N, AN, VN SUFFIX
CASE 626

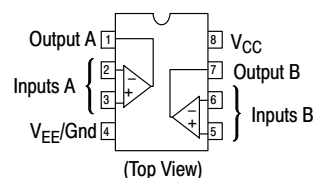


SOIC-8
D, VD SUFFIX
CASE 751



Micro8™
DMR2 SUFFIX
CASE 846A

PIN CONNECTIONS



ORDERING INFORMATION

See detailed ordering and shipping information in the package dimensions section on page 10 of this data sheet.

DEVICE MARKING INFORMATION

See general marking information in the device marking section on page 11 of this data sheet.

LM258, LM358, LM358A, LM2904, LM2904A, LM2904V, NCV2904, NCV2904V



Figure 1.

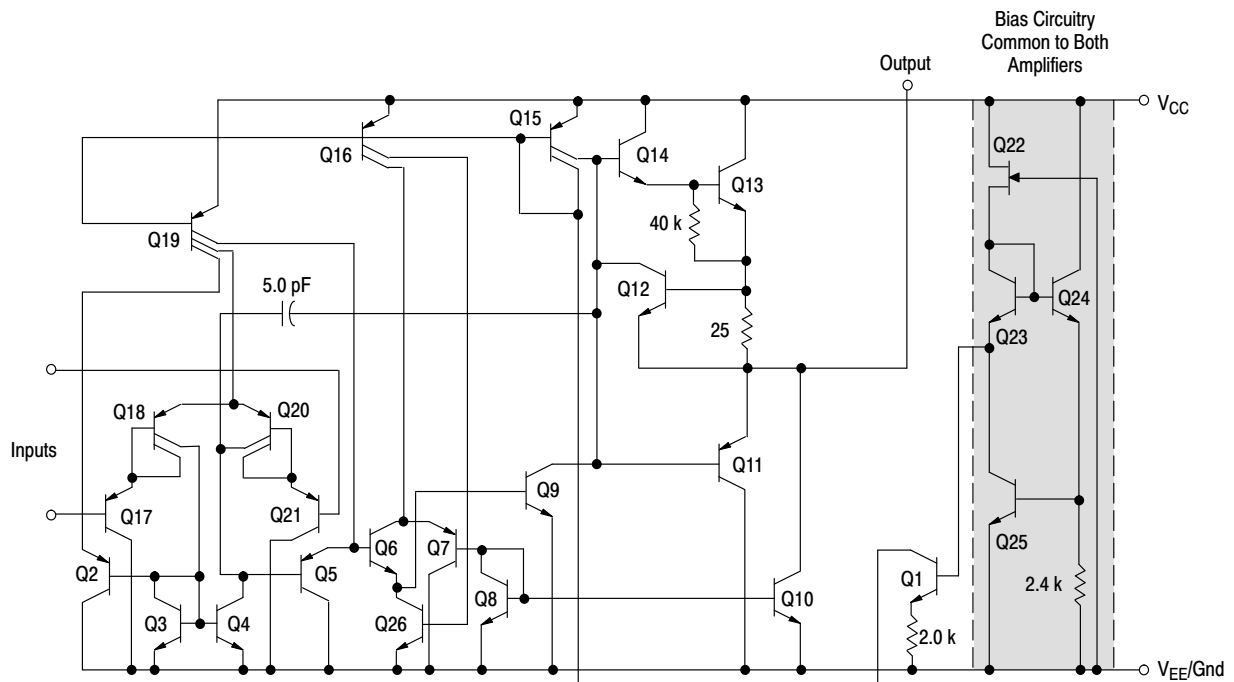


Figure 2. Representative Schematic Diagram
(One-Half of Circuit Shown)



TIP31C

Power transistors

General features

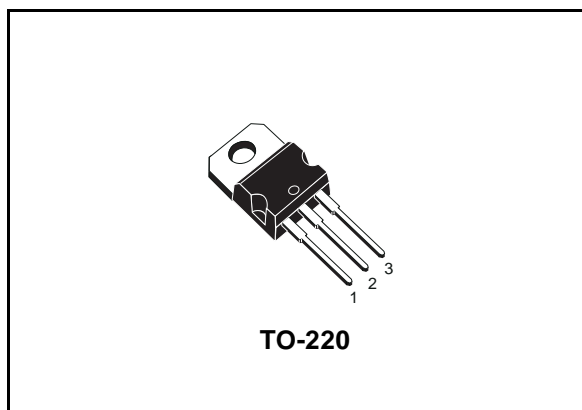
- New enhanced series
- High switching speed
- h_{FE} improved linearity
- h_{FE} Grouping

Applications

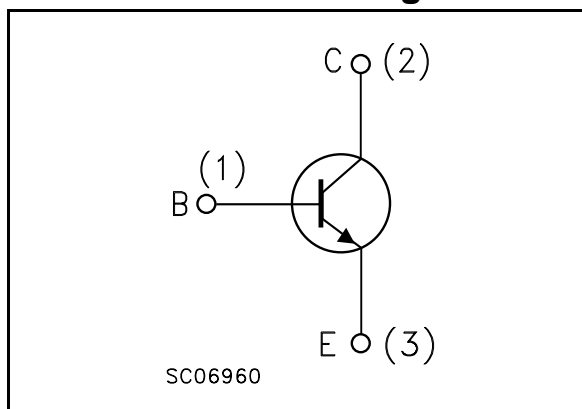
- Linear and switching industrial application

Description

The TIP31C is a base island technology NPN power transistor in TO-220 plastic package with better performances than the industry standard TIP31C that make this device suitable for audio, power linear and switching applications. The PNP type is TIP32C.



Internal schematic diagram



Order codes

Part Number	Marking	Package	Packing
TIP31C <i>Note: on page 4</i>	TIP31C R TIP31C O TIP31C Y	TO-220	Tube

Contents

1 **Electrical ratings** 3

2 **Electrical characteristics** 4

 2.1 Electrical characteristics (curve) 5

3 **Package mechanical data** 6

4 **Revision history** 8

D. Referencias

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_l_datasheet.pdf