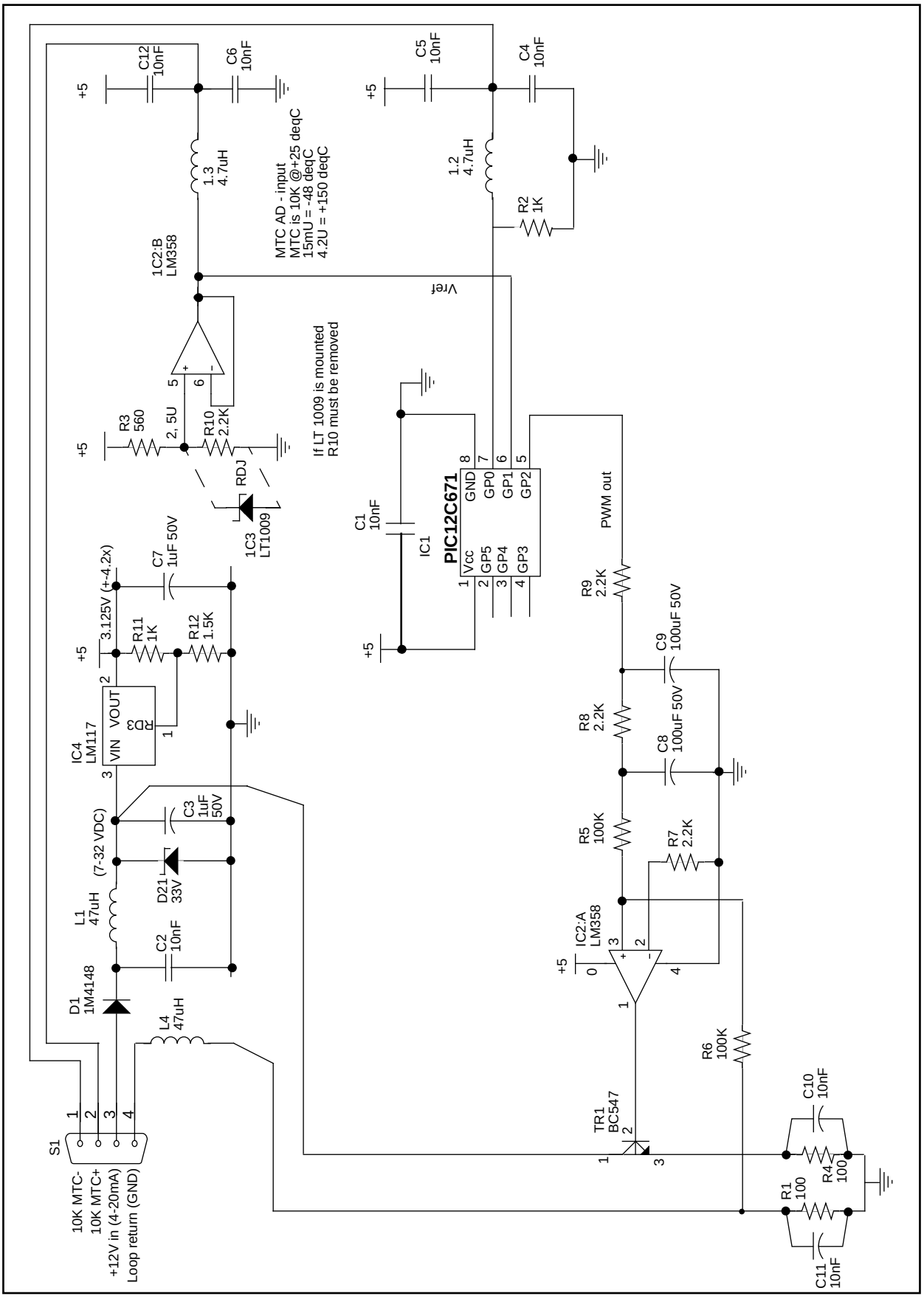
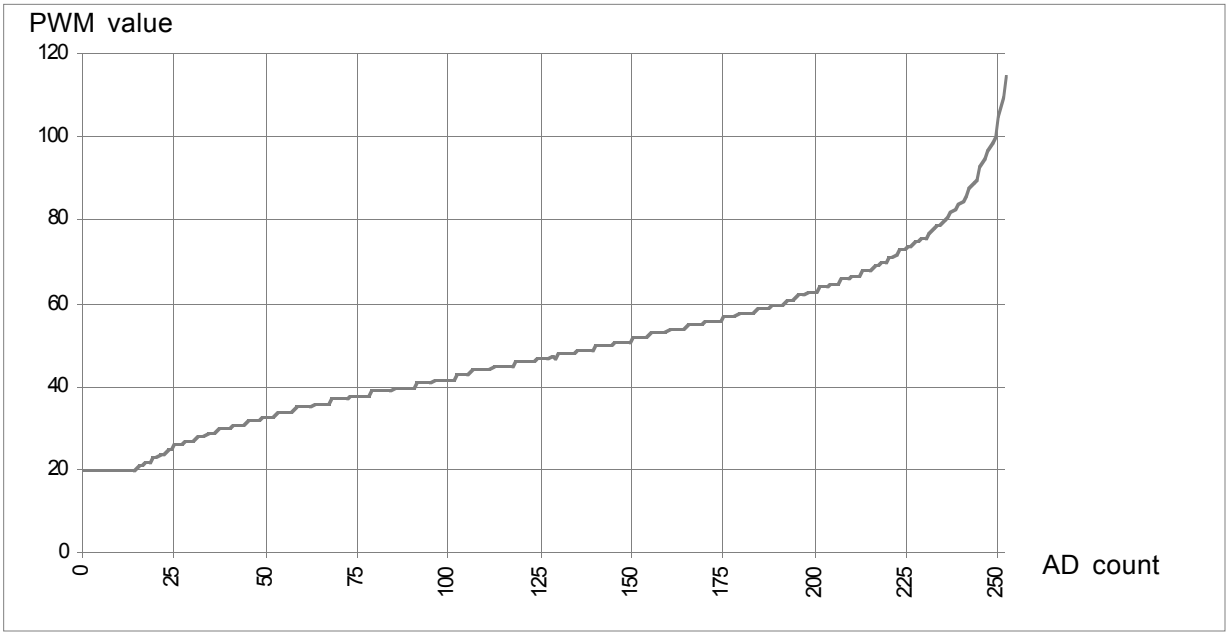
Temperature Transmitter

Author: *Ken Matiason*
Pottegade 5
DK-6400 Sonderborg
Denmark
mathiase@post5.tele.dk

The following is an example of using PIC12C671 to measure and transmit temperature using a 4-20 mA loop.



Plot of TEMP-FE.S01 (Protel. Schmatic)



Accuracy of 4-20mA temperature transmitter
Accuracy calculation (root mean sqr)

Standard		**X^2**	**LT1009**	**X^2**
Reference voltage	4.2%	0.00180	0.4%	0.00002
R10	1.0%	0.00010	---	---
R3	1.0%	0.00010	---	---
NTC	5.0%	0.00250	1.0%	0.00010
R-shunt, R2	1.0%	0.00010	0.1%	0.00000
PIC, input (8 bit)	0.4%	0.00002	0.4%	0.00002
PIC, output (8 bit)	0.4%	0.00002	0.1%	0.00000
Current out, R1	1.0%	0.00010	0.1%	0.00000
Current out, R4	1.0%	0.00010	0.1%	0.00000
SUM	0.00483			0.00015
Sqr	0.06950			0.01229
Result		± 6.9%		± 1.2%

APPENDIX A: SOURCE CODE

```
;*******************************************************************************
;*
;* File    : c:\mplab\wrk\temp-fe.asm
;* Project : A Microchip $100 project
;* Purpose : 4-20mA loop powered temperature transmitter using PIC12C671
;*
;* Date      Ver.       Name           Note
;* 970716    0.0        Ken Mahtiasen  Initial layout, test with PIC16C71
;*
;*
;*
;*******************************************************************************

            Title      “PIC12C671 4-20mA temperature transmitter”

            list P=16C71, n=74, R=DEC, st=off
            errorlevel -302, -306

            INCLUDE “P16C71.INC”
            INCLUDE “ntc_tab.asm” ; A/D-converter to PWM-output conversion table

;*******************************************************************************
;*
;* Input    : 10k NTC using 2,5V reference voltage
;* Output   : SW-generated PWM signal.
;* Purpose  : Test of software.
;* note     : Please note the code is compiled for PIC16C71 due to
;*               : missing 12C671 MPLAB driver
;*               :
;*               :
;***********************  Data RAM Assignments  **********************
;
            cblock     0Ch            ; variables 0Bh - 2Fh

W_Temp                                ; Register for W during interrupt
Status_Temp                           ; Register for STATUS-register during interrupt

PWMvalue                              ; The PWM puls length setpoint (1-255)

Delay                                 ; Counters for the main loop delay
DelayCnt

            endc

;
;*****************************************************************
;
            ORG     0000              ; Reset vector
            goto       Start

            ORG        0004           ; Interrupt vector
            goto       Interrupt

Start
;           movwf      OSCCAL         ; Running wirh PIC12C671 this line must be adopted



Init        clrf       PORTA          ; Initialize Port A and Port A of the PIC 16C71
            clrf       PORTB

            bsf        STATUS,RP0     ; BANK 1
            movlw      b’00100000’    ;
            movwf      TRISB          ;
```

```
            movlw      b'01000011'    ; TMR0 prescaler is 16 => 00-FF = 4,096ms
            movwf      OPTION_REG

            movlw      b'00011111'    ; RA0 is NTC input, RA3 is Vref
            movwf      TRISA
            movlw      b'00000001'    ; A/D input: RA3 is Vref.
            movwf      ADCON1
            bcf        STATUS,RP0    ; BANK 0
            clrf       ADCON0         ; CH = 0 (RA0). Turn on AD-module (is allways on)
            movlw      b'10100001'    ; Enable Timer interrupt
            movwf      INTCON


main                                  ; MAIN LOOP

            movlw      b'10000001'
            movwf      ADCON0
            bsf        ADCON0,GO      ; start A/D conversion
ADok        btfss      ADCON0,ADIF
            goto       ADok           ; Wait for A/D-converter to finish

            movf       ADRES,W        ; Read the A/D-value

            call       NTC_table      ; Convert the A/D converter signal
                                      ; to the PWN signal (used by IRQ routine).

            movwf      PWMvalue       ; Save the PWM value for the IRQ-routine

            movlw      60             ; Main loop delay of approx. 50ms
            movwf      DelayCnt
Redo_1      movlw      255
            movwf      Delay
Redo_2      decfsz     Delay,F
            goto       Redo_2
            decfsz     DelayCnt,F
            goto       Redo_1

            goto       main


;*************************************************************************
; Interrupt service routine
;*************************************************************************
;
; Input    : PWMvalue (40 => 0,4V output 255 => 3,125V output)
; Output   : SW-generated PWM signal to PortB,0
; note     : Please note the code is compiled for PIC16C71 due to
;               : missing 12C671 MPLAB driver


Interrupt                             ; Variable IRQ-time from 16us to 6,1ms
                                      ; The IRQ routine uses approx. 22us

            movwf      W_Temp         ; Copy W to Temporary register
            swapf      STATUS,W       ; Swap status to be saved into W
            movwf      Status_Temp    ; Sate status to Status_Temp register

                                      ; ON time is equal to TRM0 time
            btfss   PORTB,0           ; level shift: ON to OFF or OFF to ON ?
            goto    OffToOn
OnToOff     bcf        PORTB,0        ; From ON to OFF
            movf       PWMvalue,W     ; Check ONTIME and update TIME
            goto       SetTime
OffToOn     bsf        PORTB,0        ; From OFF to ON
            movf       PWMvalue,W
            sublw      0              ; Ontime = 256 - OFFtime
```

```
            btfsc      STATUS,Z
            movlw      255            ; check for PWMvalue = 0
SetTime movwf TMR0

            swapf      Status_Temp,W  ; Swap Status_Temp register into W
            movwf      STATUS         ; Move W into STATUS register
            swapf      W_Temp,F       ; Swap W_Temp
            swapf      W_Temp,W       ; Swap W_Temp into W

            bcf        INTCON,T0IF    ; enable IRQ

            retfie

   END
```