```
;*********************************************************************
;*          Author: Chris Sluder
;*          Note: code can be used for Atiny2616 uC
;*          This program sense the output voltage (0-5VDC) Via Pin 4 (ADC7)
;*          and convert it into a digital format (8 bit Resolution)
;*          Date:03/14/2008    Firmware ver. A   (COMPLETED TEST)!!!
;*               03/25/2008    Optimized the bite reversal
;*          FUSE SETTING:
;*                  Brown-out detection level at VCC=2.7V
;*                  Brown-out detection enabled
;*                  Int. RC Osc. 4 Mhz Start-up time: 6CK+0ms
;*********************************************************************

.include "tn261def.inc" ; Change for each selected controller
.def Temp    =r16   ; Temporary register
.def Temp2   =r20   ; Temporary register
.def Temp3   =r21   ; Temporary registor
.def Temp4   =r22   ; Temporary registor
.def Temp5   =r23   ; Temporary registor
.def Temp6   =r24   ; Temporary registor
.def Temp7   =r25   ; Temporary registor
.def Delay   =r17   ; Delay variable 1
.def Delay2  =r18   ; Delay variable 2
.def Delay3  =r19   ; Delay variable 3

.org 0x0000          ;Places the following code to address 0x0000
    rjmp RESET

.org OVF0addr       ;Interrupt for Timer overflow
    rjmp TIMEROF

.org USI_OVFaddr    ;Interrupt for USI overflow
    rjmp USIOF

;***** Initialization*****************************

RESET:
LDI Temp4, $DF
OUT $3D, Temp4 ;Set stack Pointer to Highest

    SBI ADCSR, 7 ;Set ADCSR bit 7 (ADEN) to enable

    LDI Temp, 0b11111111
    OUT DDRA, Temp  ;Set all pins on port A as outputs

    LDI Temp, 0b00
    OUT PORTA,Temp  ;Output logic low on all pins of port A

    LDI Temp, 0b00000011
    OUT DDRB, Temp ;set all pins on port B as input except pin 1 ,2 DI,DO

    LDI Temp, 0b00000011
    OUT PORTB, Temp ;Input logic low on output pins and no pull-up resistors on input pins


MUX:
    SBI ADCSR, 7 ;Set ADCSR bit 7 (ADEN) to enable
    LDI Temp,$27
    OUT ADMUX, Temp ; LOAD ADMUX (REFS 1,0= 0:0;(AVCC Voltage Ref 5.00VDC))
                    ; ADLAR = 1 (LEFT Adjust)
                    ; MUX 4,3,2,1,0 = 0:0:1:1:1 ( input pin 7 (ADC7))

START:
    SBI ADCSR, 6 ;Set bit 6 to start ADC conversion
```

```
CONTINUE:
    SBIC ADCSR, 6;Check if ADC conversion is finished
RJMP CONTINUE
    IN Temp,$05; LOAD ADCH INTO Temp
CALL REVERSE
    IN Temp,$04; LOAD ADCL INTO Temp
CALL REVERSE
    LDI Temp, $0D ; Send carriage return
CALL REVERSE

RJMP START

REVERSE:;REVERSE THE BIT SO MSB = LSB, LSB = MSB

    LDI Temp3, 0x08
rev1:
    LSL Temp
    ROR Temp2
    DEC Temp3
    BRNE rev1
    MOV Temp, Temp2

MOVE_SHIFT: ;Move and shift the ADC Data into two separate registers

    MOV Temp4, Temp;MOVE DATE INTO SECOND FRAME
    MOV Temp7, Temp;MOVE DATE INTO FIRST FRAME
    LSL Temp4
    LSL Temp4
    LSL Temp4
    LSL Temp4
    LSL Temp4
    LSL Temp4;SHIFT REGISTER LEFT 6 TIMES
    SBR Temp4,$3F;SET STOP BITS IN SECOND FRAME
    LSR Temp7
    LSR Temp7;SHIFT REGISTER RIGHT 2 TIMES
    CBR Temp7,$40;SET START BITS IN FIRST FRAME
    SBR Temp7,$80

    LDI Delay3, $05
DLY:
    dec Delay
    brne DLY
    dec Delay2
    brne DLY
    dec Delay3
    brne DLY

TRANSFER: ; Initialize USI UART Tx

    LDI Temp6, (1<<USIOIE)+(0<<USIWM1)+(1<<USIWM0)+(0<<USICS1)+(1<<USICS0)+(0<<USICLK)+(0<
<USITC)
                        ; Enable USI Counter OVF
                        ; Select THREE Wire mode.
                        ; Select Timer0 OVF as USI Clock source.


    ;For the first frame, set USI counter to (15-(7 data bits))= 8
    ;Once the USI counts to (15-->0) then an USI interrupt is generated to reload the seco
nd frame
    ;to continue the data transmission.
    LDI Temp2,(1<<USIOIF)+(1<<USICNT3)+(0<<USICNT2)+(0<<USICNT1)+(0<<USICNT0)
    OUT USISR, Temp2 ;LOAD USI COUNTER TO 8

    LDI Temp2,(0<<CS02)+(1<<CS01)+(0<<CS00); Reset the prescaler(CK/8).
```

```
    OUT TCCR0, Temp2

    ;Preload Timer0
    LDI Temp, $CC ; reset timer with 204 and add with current value
    OUT TCNT0,Temp ; set Timer/counter

    LDI Temp,(1<<TOIE0) ; timer overflow interrupt enable
    OUT TIMSK,Temp

    OUT USIDR, Temp7;LOAD FRIST FRAME IN BUFFER

    SET ; Set the T flag(This is used to differentiate the frames during the interrupts)
    SEI ; Enable interrupts


START_TRANSFER_FRAME:

    OUT USICR, Temp6

CLEAR:

RET

;***************SERVICE INTERRUPT ROUTINES**********************
TIMEROF:

    LDI Temp, $CC ; reset timer with 204 and add with current value
    IN Temp2, TCNT0
    ADD Temp, Temp2
    OUT TCNT0,Temp ; set Timer/counter

reti

USIOF:

    BRTC CLEAR ;Is this the last frame?

    LDI Temp2, (1<<USIOIF)+(1<<USICNT3)+(1<<USICNT2)+(1<<USICNT1)+(0<<USICNT0)
    OUT USISR, Temp2 ;LOAD USI COUNTER TO 14 (16 - (2 bit))

    OUT USIDR, Temp4;LOAD SECOND FRAME IN BUFFER

    CLT  ; CLEAR T-FLAG (first frame has been sent)

reti
```