```

AVRASM ver. 2.2.7  E:\ESE_280\$MyDocuments$\Atmel Studio\7.0\lab_11\ADC_sgnl_conv\ADC_sgnl_conv\main.asm Tue Nov 17 18:18:10 2020

E:\ESE_280\$MyDocuments$\Atmel Studio\7.0\lab_11\ADC_sgnl_conv\ADC_sgnl_conv\main.asm(9): Including file 'C:/Program Files (x86)\Atmel\Studio\7.0\Packs\atmel\ATmega_DFP\1.3.300\avrasm\inc\m4809def.inc'
E:\ESE_280\$MyDocuments$\Atmel Studio\7.0\lab_11\ADC_sgnl_conv\ADC_sgnl_conv\main.asm(9): Including file 'C:/Program Files (x86)\Atmel\Studio\7.0\Packs\atmel\ATmega_DFP\1.3.300\avrasm\inc\m4809def.inc'


                                 ; ADC_sgnl_conv.asm
                                 ;
                                 ; Created: 11/17/2020 2:02:20 PM
                                 ; Author : Judah Ben-Eliezer
                                 ;

                                 .list

                                 .equ PERIOD_EXAMPLE_VALUE = 25

                                 .dseg
002800                           led_display: .byte 4
002804                           digit_num: .byte 1


                                 .cseg

                                 reset:
000000 940c 0010                    jmp start

                                 .org TCA0_OVF_vect
00000e 940c 0038                    jmp post_display_ISR

                                 start:
                                    ;configure inputs and outputs
000010 9881                          cbi VPORTE_DIR, 1
000011 ef0f                         ldi r16, $FF
000012 b908                         out VPORTC_DIR, r16
000013 b90c                         out VPORTD_DIR, r16
000014 9500                         com r16
000015 b909                         out VPORTC_OUT, r16
000016 b90d                         out VPORTD_OUT, r16

                                    ;configure TCA0
000017 e000                         ldi r16, TCA_SINGLE_WGMODE_NORMAL_gc    ;WGMODE normal
000018 9300 0a01                    sts TCA0_SINGLE_CTRLB, r16
```

```

                                        ;enable overflow interrupt
00001a e001                             ldi r16, TCA_SINGLE_OVF_bm
00001b 9300 0a0a                        sts TCA0_SINGLE_INTCTRL, r16

                                        ;load period low byte then high byte
00001d e109                             ldi r16, LOW(PERIOD_EXAMPLE_VALUE)
00001e 9300 0a26                        sts TCA0_SINGLE_PER, r16
000020 e000                             ldi r16, HIGH(PERIOD_EXAMPLE_VALUE)
000021 9300 0a27                        sts TCA0_SINGLE_PER + 1, r16

                                        ;set clock and start timer
000023 e00d                             ldi r16, TCA_SINGLE_CLKSEL_DIV256_gc | TCA_SINGLE_ENABLE_bm
000024 9300 0a00                        sts TCA0_SINGLE_CTRLA, r16

                                        ;set voltage reference
000026 e200                             ldi r16, VREF_ADC0REFSEL_2V5_gc
000027 9300 00a0                        sts VREF_CTRLA, r16

                                        ;select PE1/ AIN9
000029 e009                             ldi r16, ADC_MUXPOS_AIN9_gc
00002a 9300 0606                        sts ADC0_MUXPOS, r16

                                        ;enable internal reference and set prescaler to div 64
00002c e005                             ldi r16, ADC_PRESC_DIV64_gc | ADC_REFSEL_INTREF_gc
00002d 9300 0602                        sts ADC0_CTRLC, r16

                                        ;set resolution to 10 bit and enable adc
00002f e001                             ldi r16, ADC_RESSEL_10BIT_gc | ADC_ENABLE_bm;
000030 9300 0600                        sts ADC0_CTRLA, r16

                                        ;start conversion
000032 e001                             ldi r16, ADC_STCONV_bm;
000033 9300 0608                        sts ADC0_COMMAND, r16

                                        ;enable interrupts
000035 9478                             sei
000036 940c 0047                        jmp wait_for_post

                                    ;***************************************************************************
                                    ;*
                                    ;* "post_display" - title
                                    ;*
                                    ;* Description: toggles value for all PORTC
```

```
                    pins.  Since PORTC is used to multiplex the led display, this will
                                 ;* turn the LED display on and off
                                 ;* Author: Judah Ben-Eliezer
                                 ;* Version:    1.0
                                 ;* Last updated:   11/17
                                 ;* Target: ATmega4809
                                 ;* Number of words:    13
                                 ;* Number of cycles:   6
                                 ;* Low registers modified:
                                 ;* High registers modified:
                                 ;* Parameters: none
                                 ;* Returns:        none
                                 ;*
                                 ;* Notes:
                                 ;*
                                 ;*************************************************************************
                                 post_display_ISR:
000038 930f                         push r16
000039 b70f                         in r16, CPU_SREG
00003a 930f                         push r16
00003b 931f                         push r17

00003c ef1f                         ldi r17, $FF
00003d b109                         in r16, VPORTC_OUT
00003e 2701                         eor r16, r17
00003f b909                         out VPORTC_OUT, r16

                                    ;ldi r16, TCA_SINGLE_OVF_bm ;clear OVF flag
                                    ;sts TCA0_SINGLE_INTFLAGS, r16

000040 911f                         pop r17
000041 910f                         pop r16
000042 bf0f                         out CPU_SREG, r16
000043 910f                         pop r16

000044 9478                         sei
000045 940c 0049                    jmp main_loop

                                 wait_for_post:
000047 0000                         nop
000048 cffe                         rjmp wait_for_post

                                 main_loop:
000049 d012                         rcall multiplex_display
00004a d026                         rcall mux_digit_delay
00004b 9130 060b                    lds r19, ADC0_INTFLAGS
00004d fd30                         sbrc r19, 0
```

```
00004e d001                         rcall read
00004f cff9                         rjmp main_loop

                                 ;***************************************************************************
                                 ;*
                                 ;* "read" - title
                                 ;*
                                 ;* Description: loads ADC0_RES into r17:r16 and calls bin16_to_led
                                 ;*
                                 ;* Author:  Judah Ben-Eliezer
                                 ;* Version:     1.0
                                 ;* Last updated:    11/17/2020
                                 ;* Target:  ATmega4809
                                 ;* Number of words:
                                 ;* Number of cycles:
                                 ;* Low registers modified:  none
                                 ;* High registers modified:         r17:r16
                                 ;*
                                 ;* Parameters:  ADC0_RES
                                 ;* Returns:         r17:r16
                                 ;*
                                 ;* Notes:
                                 ;*
                                 ;***************************************************************************
                                 read:
000050 9110 0611                    lds r17, ADC0_RESH
000052 9100 0610                    lds r16, ADC0_RESL
000054 d023                         rcall bin16_to_led

                                    ;reset interrupt flag
000055 e001                         ldi r16, ADC_RESRDY_bm;
000056 9300 060a                    sts ADC0_INTCTRL, r16

                                    ;restart conversion
000058 e001                         ldi r16, ADC_STCONV_bm;
000059 9300 0608                    sts ADC0_COMMAND, r16

00005b 9508                         ret

                                 ;***************************************************************************
                                 ;*
                                 ;* "multiplex_display" - title
                                 ;*
                                 ;* Description:    outputs values from led_display array to 7 segment display on PORTD driven by
```

```
highest two bits of PORTC
                                 ;*
                                 ;* Author:  Judah Ben-Eliezer
                                 ;* Version:     1.0
                                 ;* Last updated:    11/10/2020
                                 ;* Target:  ATmega4809
                                 ;* Number of words:
                                 ;* Number of cycles:
                                 ;* Low registers modified:
                                 ;* High registers modified:
                                 ;*
                                 ;* Parameters:
                                 ;* Returns:
                                 ;*
                                 ;* Notes:
                                 ;*
                                 ;*****************************************************************************
                                 multiplex_display:
00005c e2d8                         ldi YH, HIGH(led_display)
00005d e0c0                         ldi YL, LOW(led_display)
00005e 9110 2804                    lds r17, digit_num
000060 7013                         andi r17, $03
000061 2f41                         mov r20, r17
000062 0fc1                         add YL, r17
000063 8128                         ld r18, Y
000064 e850                         ldi r21, $80
000065 9543                         inc r20
                                 loop:
000066 9556                         lsr r21
000067 954a                         dec r20
000068 f7e9                         brne loop
000069 0f55                         lsl r21
00006a 9550                         com r21
00006b b959                         out VPORTC_OUT, r21
00006c b92d                         out VPORTD_OUT, r18
00006d 9513                         inc r17
00006e 9310 2804                    sts digit_num, r17
000070 9508                         ret

                                 ;*****************************************************************************
                                 ;*
                                 ;* "mux_digit_delay" - title
                                 ;*
                                 ;* Description: delays 0.1 * r23
                                 ;*
                                 ;* Author:  Judah Ben-Eliezer
                                 ;* Version:     1.0
```

```
                    ;* Last updated:
                    ;* Target:
                    ;* Number of words:
                    ;* Number of cycles:
                    ;* Low registers modified:
                    ;* High registers modified:
                    ;*
                    ;* Parameters:
                    ;* Returns:
                    ;*
                    ;* Notes:
                    ;*
                    ;*****************************************************************************

                    mux_digit_delay:
000071 e078            ldi r23, $08 ; 0.1 * r23 = delay
                    outer_loop:
000072 e086            ldi r24, $06
                    inner_loop:
000073 958a            dec r24
000074 f7f1            brne inner_loop
000075 957a            dec r23
000076 f7d9            brne outer_loop
000077 9508            ret

                    ;*****************************************************************************

                    ;*
                    ;* "bin16_to_led" - title
                    ;*
                    ;* Description:    Converts bin16 input to 7seg output, from bcd_entries array to led_display array
                    ;*
                    ;* Author: Judah Ben-Eliezer
                    ;* Version:    1.0
                    ;* Last updated:    11/17/2020
                    ;* Target: ATmega4809
                    ;* Number of words:
                    ;* Number of cycles:
                    ;* Low registers modified:
                    ;* High registers modified:
                    ;*
                    ;* Parameters: r17:r16 16 bit binary number.
                    ;* Returns:    none
                    ;*
                    ;* Notes:
                    ;*
                    ;*****************************************************************************
```

```

                                 bin16_to_led:
000078 e2b8                         ldi XH, HIGH(led_display)
000079 e0a0                         ldi XL, LOW(led_display)
00007a 2f21                         mov r18, r17
00007b 7f20                         andi r18, $F0
00007c 9522                         swap r18
00007d d00f                         rcall hex_to_7seg
00007e 932d                         st X+, r18
00007f 2f21                         mov r18, r17
000080 702f                         andi r18, $0F
000081 d00b                         rcall hex_to_7seg
000082 932d                         st X+, r18
000083 2f20                         mov r18, r16
000084 7f20                         andi r18, $F0
000085 9522                         swap r18
000086 d006                         rcall hex_to_7seg
000087 932d                         st X+, r18
000088 2f20                         mov r18, r16
000089 702f                         andi r18, $0F
00008a d002                         rcall hex_to_7seg
00008b 932c                         st X, r18
00008c 9508                         ret

                                 ;****************************************************************************
                                 ;*
                                 ;* "hex_to_7seg" - Hexadecimal to Seven Segment Conversion
                                 ;*
                                 ;* Description: Converts a right justified hexadecimal digit to the seven
                                 ;* segment pattern required to display it. Pattern is right justified a
                                 ;* through g. Pattern uses 0s to turn segments on ON.
                                 ;*
                                 ;* Author:                      Ken Short
                                 ;* Version:                         1.0
                                 ;* Last updated:                101620
                                 ;* Target:                      ATmega4809
                                 ;* Number of words:                 8
                                 ;* Number of cycles:            13
                                 ;* Low registers modified:      none
                                 ;* High registers modified:         r19, r18, ZL, ZH
                                 ;*
                                 ;* Parameters: r18: right justified hex digit,
                          high nibble 0
                             ;* Returns: r18: segment values a through g right justified
                             ;*
                             ;* Notes:
                             ;*
                             ;***************************************************************************

                             hex_to_7seg:
00008d 702f                     andi r18, 0x0F              ;clear ms nibble
00008e e0f1                      ldi ZH, HIGH(hextable * 2)  ;set Z to point to start of table
00008f e2ea                      ldi ZL, LOW(hextable * 2)
000090 e030                      ldi r19, $00                ;add offset to Z pointer
000091 0fe2                      add ZL, r18
000092 1ff3                      adc ZH, r19
000093 9124                      lpm r18, Z                  ;load byte from table pointed to by Z
000094 9508                     ret

                                 ;Table of segment values to display digits 0 - F
                                 ;!!! seven values must be added - verify all values
000095 4f01
000096 0612
000097 244c
000098 0f20
000099 0400
00009a 6008
00009b 4231
00009c 3830                  hextable: .db $01, $4F, $12, $06, $4C, $24, $20, $0F, $00, $04, $08, $60, $31, $42, $30, $38


RESOURCE USE INFORMATION
------------------------

Notice:
The register and instruction counts are symbol table hit counts,
and hence implicitly used resources are not counted, eg, the
'lpm' instruction without operands implicitly uses r0 and z,
none of which are counted.

x,y,z are separate entities in the symbol table and are
counted separately from r26..r31 here.
```

```

.dseg memory usage only counts static data declared with .byte

"ATmega4809" register use summary:
x  :   4 y  :   1 z  :   1 r0 :   0 r1 :   0 r2 :   0 r3 :   0 r4 :   0 
r5 :   0 r6 :   0 r7 :   0 r8 :   0 r9 :   0 r10:   0 r11:   0 r12:   0 
r13:   0 r14:   0 r15:   0 r16:  42 r17:  13 r18:  19 r19:   4 r20:   3 
r21:   5 r22:   0 r23:   2 r24:   2 r25:   0 r26:   1 r27:   1 r28:   2 
r29:   1 r30:   2 r31:   2 
Registers used: 17 out of 35 (48.6%)

"ATmega4809" instruction use summary:
.lds  :   0 .sts  :   0 adc   :   1 add   :   2 adiw  :   0 and   :   0 
andi  :   6 asr   :   0 bclr  :   0 bld   :   0 brbc  :   0 brbs  :   0 
brcc  :   0 brcs  :   0 break :   0 breq  :   0 brge  :   0 brhc  :   0 
brhs  :   0 brid  :   0 brie  :   0 brlo  :   0 brlt  :   0 brmi  :   0 
brne  :   3 brpl  :   0 brsh  :   0 brtc  :   0 brts  :   0 brvc  :   0 
brvs  :   0 bset  :   0 bst   :   0 call  :   0 cbi   :   1 cbr   :   0 
clc   :   0 clh   :   0 cli   :   0 cln   :   0 clr   :   0 cls   :   0 
clt   :   0 clv   :   0 clz   :   0 com   :   2 cp    :   0 cpc   :   0 
cpi   :   0 cpse  :   0 dec   :   3 des   :   0 eor   :   1 fmul  :   0 
fmuls :   0 fmulsu:   0 icall :   0 ijmp  :   0 in    :   2 inc   :   2 
jmp   :   4 ld    :   1 ldd   :   0 ldi   :  24 lds   :   4 lpm   :   2 
lsl   :   1 lsr   :   1 mov   :   5 movw  :   0 mul   :   0 muls  :   0 
mulsu :   0 neg   :   0 nop   :   1 or    :   0 ori   :   0 out   :   8 
pop   :   3 push  :   3 rcall :   8 ret   :   5 reti  :   0 rjmp  :   2 
rol   :   0 ror   :   0 sbc   :   0 sbci  :   0 sbi   :   0 sbic  :   0 
sbis  :   0 sbiw  :   0 sbr   :   0 sbrc  :   1 sbrs  :   0 sec   :   0 
seh   :   0 sei   :   2 sen   :   0 ser   :   0 ses   :   0 set   :   0 
sev   :   0 sez   :   0 sleep :   0 spm   :   0 st    :   4 std   :   0 
sts   :  13 sub   :   0 subi  :   0 swap  :   2 tst   :   0 wdr   :   0 

Instructions used: 30 out of 114 (26.3%)

"ATmega4809" memory use summary [bytes]:
Segment   Begin    End      Code   Data   Used    Size   Use%
---------------------------------------------------------------
[.cseg] 0x000000 0x00013a    274     16    290   49152   0.6%
[.dseg] 0x002800 0x002805      0      5      5    6144   0.1%
[.eseg] 0x000000 0x000000      0      0      0     256   0.0%

Assembly complete, 0 errors, 0 warnings

```