```
 1
 2  AVRASM ver. 2.2.7  E:\ESE_280\$MyDocuments$\Atmel Studio\7.0\lab_9
      \PE0_and_PE2_intrs\PE0_and_PE2_intrs\main.asm Tue Nov 03 19:29:58 2020
 3
 4  E:\ESE_280\$MyDocuments$\Atmel Studio\7.0\lab_9\PE0_and_PE2_intrs
      \PE0_and_PE2_intrs\main.asm(9): Including file 'C:/Program Files (x86)\Atmel
      \Studio\7.0\Packs\atmel\ATmega_DFP\1.3.300\avrasm\inc\m4809def.inc'
 5  E:\ESE_280\$MyDocuments$\Atmel Studio\7.0\lab_9\PE0_and_PE2_intrs
      \PE0_and_PE2_intrs\main.asm(9): Including file 'C:/Program Files (x86)\Atmel
      \Studio\7.0\Packs\atmel\ATmega_DFP\1.3.300\avrasm\inc\m4809def.inc'
 6
 7
 8                                    ; PE0_and_PE2_intrs.asm
 9                                    ;
10                                    ; Created: 10/30/2020 9:44:44 PM
11                                    ; Author : hp
12                                    ;
13
14                                    .list
15
16                                    .dseg
17  002800                            PB1_count: .byte 1  ;pushbutton 1 presses.
18  002801                            PB2_count: .byte 1  ;pushbutton 2 presses.
19
20
21                                    .cseg                   ;start of code segment
22                                    reset:
23  000000 940c 0048                     jmp start           ;reset vector executed
      a power on
24
25                                    .org PORTE_PORT_vect
26  000046 940c 005c                     jmp porte_isr       ;vector for all PORTE
      pin change IRQs
27
28
29                                    start:
30                                        ; Configure I/O ports
31  000048 9880                       cbi VPORTE_DIR, 0   ;PE0 input- gets output
       from PB1
32  000049 9882                       cbi VPORTE_DIR, 2   ;PE2 input- gets output
       from PB2
33
34  00004a e000                       ldi r16, 0x00       ;make initial counts 0
35  00004b 9300 2800                  sts PB1_count, r16
36  00004d 9300 2801                  sts PB2_count, r16
37
38                                    ;Configure interrupts
39  00004f 9100 0490                  lds r16, PORTE_PIN0CTRL ;set ISC for PE0 to
       pos. edge
```

```
000051 6002                     ori r16, 0x02       ;set ISC for rising edge
000052 9300 0490                sts PORTE_PIN0CTRL, r16

000054 9100 0492                lds r16, PORTE_PIN2CTRL ;set ISC for PE2 to pos. edge
000056 6002                     ori r16, 0x02       ;set ISC for rising edge
000057 9300 0492                sts PORTE_PIN2CTRL, r16

000059 9478                     sei         ;enable global interrupts

                              main_loop:    ;main program loop
00005a 0000                     nop
00005b cffe                     rjmp main_loop


                               ;Interrupt service routine for any PORTE pin change IRQ
                              porte_ISR:
00005c 930f                     push r16        ;save r16 then SREG
00005d b70f                     in r16, CPU_SREG
00005e 930f                     push r16
00005f 94f8                     cli             ;clear global interrupt enable

                                ;Determine which pins of PORTE have IRQs
000060 9100 0489                lds r16, PORTE_INTFLAGS ;check for PE0 IRQ flag set
000062 fd00                     sbrc r16, 0
000063 d008                     rcall PB1_sub           ;execute subroutine for PE0

000064 9100 0489                lds r16, PORTE_INTFLAGS ;check for PE2 IRQ flag set
000066 fd02                     sbrc r16, 2
000067 d00d                     rcall PB2_sub           ;execute subroutine for PE2

000068 910f                     pop r16         ;restore SREG then r16
000069 bf0f                     out CPU_SREG, r16
00006a 910f                     pop r16
00006b 9518                     reti            ;return from PORTE pin change ISR


                              ;Subroutines called by porte_ISR
                              PB1_sub:      ;PE0's task to be done
00006c 9100 2800                lds r16, PB1_count      ;get current count
```

```
      for PB1
 79 00006e 9503                      inc r16                 ;increment count
 80 00006f 9300 2800                 sts PB1_count, r16      ;store new count
 81 000071 e001                      ldi r16, PORT_INT0_bm   ;clear IRQ flag for ↪
       PE0
 82 000072 9300 0489                 sts PORTE_INTFLAGS, r16
 83 000074 9508                      ret
 84
 85
 86                                PB2_sub:       ;PE2's task to be done
 87 000075 9100 2801                 lds r16, PB2_count      ;get current count  ↪
      for PB2
 88 000077 9503                      inc r16                 ;increment count
 89 000078 9300 2801                 sts PB2_count, r16      ;store new count
 90 00007a e004                      ldi r16, PORT_INT2_bm   ;clear IRQ flag for ↪
       PE2
 91 00007b 9300 0489                 sts PORTE_INTFLAGS, r16
 92 00007d 9508                      ret
 93
 94
 95
 96 RESOURCE USE INFORMATION
 97 ------------------------
 98
 99 Notice:
100 The register and instruction counts are symbol table hit counts,
101 and hence implicitly used resources are not counted, eg, the
102 'lpm' instruction without operands implicitly uses r0 and z,
103 none of which are counted.
104
105 x,y,z are separate entities in the symbol table and are
106 counted separately from r26..r31 here.
107
108 .dseg memory usage only counts static data declared with .byte
109
110 "ATmega4809" register use summary:
111 x  :   0 y  :   0 z  :   0 r0 :   0 r1 :   0 r2 :   0 r3 :   0 r4 :   0
112 r5 :   0 r6 :   0 r7 :   0 r8 :   0 r9 :   0 r10:   0 r11:   0 r12:   0
113 r13:   0 r14:   0 r15:   0 r16:  29 r17:   0 r18:   0 r19:   0 r20:   0
114 r21:   0 r22:   0 r23:   0 r24:   0 r25:   0 r26:   0 r27:   0 r28:   0
115 r29:   0 r30:   0 r31:   0
116 Registers used: 1 out of 35 (2.9%)
117
118 "ATmega4809" instruction use summary:
119 .lds  :   0 .sts  :   0 adc   :   0 add   :   0 adiw  :   0 and   :   0
120 andi  :   0 asr   :   0 bclr  :   0 bld   :   0 brbc  :   0 brbs  :   0
121 brcc  :   0 brcs  :   0 break :   0 breq  :   0 brge  :   0 brhc  :   0
122 brhs  :   0 brid  :   0 brie  :   0 brlo  :   0 brlt  :   0 brmi  :   0
123 brne  :   0 brpl  :   0 brsh  :   0 brtc  :   0 brts  :   0 brvc  :   0
```

```
124 brvs  :   0 bset  :   0 bst   :   0 call  :   0 cbi   :   2 cbr   :   0
125 clc   :   0 clh   :   0 cli   :   1 cln   :   0 clr   :   0 cls   :   0
126 clt   :   0 clv   :   0 clz   :   0 com   :   0 cp    :   0 cpc   :   0
127 cpi   :   0 cpse  :   0 dec   :   0 des   :   0 eor   :   0 fmul  :   0
128 fmuls :   0 fmulsu:   0 icall :   0 ijmp  :   0 in    :   1 inc   :   2
129 jmp   :   2 ld    :   0 ldd   :   0 ldi   :   3 lds   :   6 lpm   :   0
130 lsl   :   0 lsr   :   0 mov   :   0 movw  :   0 mul   :   0 muls  :   0
131 mulsu :   0 neg   :   0 nop   :   1 or    :   0 ori   :   2 out   :   1
132 pop   :   2 push  :   2 rcall :   2 ret   :   2 reti  :   1 rjmp  :   1
133 rol   :   0 ror   :   0 sbc   :   0 sbci  :   0 sbi   :   0 sbic  :   0
134 sbis  :   0 sbiw  :   0 sbr   :   0 sbrc  :   2 sbrs  :   0 sec   :   0
135 seh   :   0 sei   :   1 sen   :   0 ser   :   0 ses   :   0 set   :   0
136 sev   :   0 sez   :   0 sleep :   0 spm   :   0 st    :   0 std   :   0
137 sts   :   8 sub   :   0 subi  :   0 swap  :   0 tst   :   0 wdr   :   0
138
139 Instructions used: 19 out of 114 (16.7%)
140
141 "ATmega4809" memory use summary [bytes]:
142 Segment   Begin    End      Code   Data   Used    Size   Use%
143 ----------------------------------------------------------------
144 [.cseg] 0x000000 0x0000fc    116      0    116   49152   0.2%
145 [.dseg] 0x002800 0x002802      0      2      2    6144   0.0%
146 [.eseg] 0x000000 0x000000      0      0      0     256   0.0%
147
148 Assembly complete, 0 errors, 0 warnings
149
```