```

AVRASM ver. 2.2.7  C:\Users\Seyi Olajuyi\Documents\Atmel Studio\7.0\neg_level_ints\neg_level_ints
\main.asm Thu Nov 21 20:34:32 2019

C:\Users\Seyi Olajuyi\Documents\Atmel Studio\7.0\neg_level_ints\neg_level_ints\main.asm(19):
 Including file 'C:/Program Files (x86)\Atmel\Studio\7.0\Packs\atmel\ATmega_DFP\1.3.300\
 avrasm\inc\m324adef.inc'
C:\Users\Seyi Olajuyi\Documents\Atmel Studio\7.0\neg_level_ints\neg_level_ints\main.asm(19):
 Including file 'C:/Program Files (x86)\Atmel\Studio\7.0\Packs\atmel\ATmega_DFP\1.3.300\avrasm\
 inc\m324adef.inc'


                                 ;*
                                 ;* Title:           neg_level_ints
                                 ;* Author:          Seyi Olajuyi & Bassel El Amine
                                 ;* Version:         1.0
                                 ;* Last updated:    11/21/19
                                 ;* Target:          ATmega324A
                                 ;*
                                 ;* DESCRIPTION
                                 ;* This program counts the number of times a key (any key) on the k eypad
                                 ;* is pressed and the number of times the pushbutton is pressed.
                                 ;*
                                 ;*
                                 ;* VERSION HISTORY
                                 ;* 1.0 Original version
                                 ;****************************************************************** *********

                                 .list

                                 .dseg
000100                           key_presses:    .byte 1
000101                           pb_presses:         .byte 1


```

```
                                 .cseg
                                 reset:
                                 .org RESET              ;reset interrupt vector
000000 c004                          rjmp start          ;program starts here at reset
                                 .org INT0addr           ;INT0 interrupt vector
000002 c01c                          rjmp keypress_ISR
                                 .org INT1addr
000004 c026                         rjmp pb_press_ISR


                                 start:
000005 9852                          cbi DDRD, 2                 ; Set pin 2 on PORTD to input
000006 9853                         cbi DDRD, 3             ; Set pin 3 on PORTD to input

000007 9a0a                         sbi DDRA, 2             ; Set pin 2 on PORTA to output
000008 9a3f                         sbi DDRC, 7             ; Set pin 7 on PORTC to output

000009 980c                         cbi DDRA, 4             ; Set pin 4 on PORTA to output

00000a 9812                         cbi PORTA, 2            ; Clear Flip-Flop
00000b 9a12                         sbi PORTA, 2

00000c 9847                         cbi PORTC, 7            ; Clear Flip-Flop
00000d 9a47                         sbi PORTC, 7

00000e e000                         ldi r16, 0              ; Clear the variables
00000f 9300 0100                    sts key_presses, r16
000011 9300 0101                    sts pb_presses, r16

000013 ef0f                          ldi r16, LOW(RAMEND)    ;initialize SP to point to top of stack
000014 bf0d                          out SPL, r16
000015 e008                          ldi r16, HIGH(RAMEND)
000016 bf0e                          out SPH, r16

000017 e000                          ldi r16, $00    ;interrupt sense control bits
```

```
000018 9300 0069                     sts EICRA, r16
00001a e003                         ldi r16, $03         ; Enable interrupt request at INTO & INT1
00001b bb0d                          out EIMSK, r16

00001c 9478                          sei                 ;set global interrupt enable

                                 main_loop:
00001d 0000                          nop                 ;stub for background task
00001e cffe                          rjmp main_loop      ;jump back to main_loop


                                 ;****************************************************************** *********
                                 ;*
                                 ;* "keypress_ISR" - Count Interrupts at INT0
                                 ;*
                                 ;* Description: Counts rising edges at INT0 (PD2)
                                 ;*
                                 ;* Author:                  Ken Short
                                 ;* Version:
                                 ;* Last updated:            10/23/17
                                 ;* Target:                  ATmega324A
                                 ;* Number of words:
                                 ;* Number of cycles:        16
                                 ;* Low registers modified:  none
                                 ;* High registers modified: none
                                 ;*
                                 ;* Parameters:  Uses PORTB register to hold the count and drive LED s
                                 ;* connected to that port.
                                 ;*
                                 ;* Notes:
                                 ;*
                                 ;****************************************************************** *********

                                     ;INT0 interrupt service routine
                                 keypress_ISR:
                                    ;cli                    ; Disable global interrupt
00001f b70f                          in r16, SREG       ;save SREG
000020 930f                          push r16

000021 9100 0100                     lds r16, key_presses       ;increment count
000023 9503                          inc r16
000024 9300 0100                    sts key_presses, r16

                                 restore_values_1:

000026 910f                          pop r16            ;restore SREG
000027 bf0f                          out SREG,r16

000028 9847                         cbi PORTC, 7
000029 9a47                         sbi PORTC, 7

00002a 9518                          reti               ;return from interrupt


                                 ;****************************************************************** *********
                                 ;*
                                 ;* "pb_press_ISR" - Count Interrupts at INT1
                                 ;*
                                 ;* Description: Counts rising edges at INT1 (PD3)
                                 ;*
                                 ;* Author:                    Ken Short
                                 ;* Version:
                                 ;* Last updated:              10/23/17
                                 ;* Target:                    ATmega324A
                                 ;* Number of words:
                                 ;* Number of cycles:          16
                                 ;* Low registers modified:    none
                                 ;* High registers modified: none
                                 ;*
                                 ;* Parameters:  Uses PORTB register to hold the count and drive LED s
                                 ;* connected to that port.
                                 ;*
                                 ;* Notes:
                                 ;*
                                 ;****************************************************************** *********

                                     ;INT1 interrupt service routine
                                 pb_press_ISR:
                                 wait_for_bounce_1:
00002b b70f                         in r16, SREG        ;save SREG
00002c 930f                          push r16

00002d 9904                         sbic PINA, 4
00002e cffc                         rjmp wait_for_bounce_1
00002f e604                         ldi r16, 100
000030 d00e                         rcall var_delay
000031 9904                         sbic PINA, 4
000032 cff8                         rjmp wait_for_bounce_1

000033 e002                         ldi r16, (1 <<INTF1)
000034 bb0c                         out EIFR, r16
000035 9812                         cbi PORTA, 2            ; Clear Flip-Flop
000036 9a12                         sbi PORTA, 2



000037 9100 0101                     lds r16, pb_presses       ;increment count
000039 9503                          inc r16
00003a 9300 0101                    sts pb_presses, r16

                                 restore_value_2:
00003c 910f                          pop r16             ;restore SREG
00003d bf0f                          out SREG,r16

00003e 9518                          reti                ;return from interrupt
```

```

                                 ;***********************************************
                                 ;SUBROUTINE FOR VAR DELAY
                                 ;***********************************************
                                 var_delay:
                                    outer_loop:
00003f e210                             ldi r17, 32
                                    inner_loop:
000040 951a                             dec r17
000041 f7f1                             brne inner_loop
000042 950a                             dec r16
000043 f7d9                             brne outer_loop


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

"ATmega324A" register use summary:
x  :   0 y  :   0 z  :   0 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:  29 r17:   2 r18:   0 r19:   0 r20:   0
r21:   0 r22:   0 r23:   0 r24:   0 r25:   0 r26:   0 r27:   0 r28:   0
r29:   0 r30:   0 r31:   0
Registers used: 2 out of 35 (5.7%)
```

```

"ATmega324A" instruction use summary:
.lds  :   0 .sts  :   0 adc   :   0 add   :   0 adiw  :   0 and   :   0
andi  :   0 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  :   2 brpl  :   0 brsh  :   0 brtc  :   0 brts  :   0 brvc  :   0
brvs  :   0 bset  :   0 bst   :   0 call  :   0 cbi   :   7 cbr   :   0
clc   :   0 clh   :   0 cli   :   0 cln   :   0 clr   :   0 cls   :   0
clt   :   0 clv   :   0 clz   :   0 com   :   0 cp    :   0 cpc   :   0
cpi   :   0 cpse  :   0 dec   :   2 eor   :   0 fmul  :   0 fmuls :   0
fmulsu:   0 icall :   0 ijmp  :   0 in    :   2 inc   :   2 jmp   :   0
ld    :   0 ldd   :   0 ldi   :   8 lds   :   2 lpm   :   0 lsl   :   0
lsr   :   0 mov   :   0 movw  :   0 mul   :   0 muls  :   0 mulsu :   0
neg   :   0 nop   :   1 or    :   0 ori   :   0 out   :   6 pop   :   2
push  :   2 rcall :   1 ret   :   1 reti  :   2 rjmp  :   6 rol   :   0
ror   :   0 sbc   :   0 sbci  :   0 sbi   :   6 sbic  :   2 sbis  :   0
sbiw  :   0 sbr   :   0 sbrc  :   0 sbrs  :   0 sec   :   0 seh   :   0
sei   :   1 sen   :   0 ser   :   0 ses   :   0 set   :   0 sev   :   0
sez   :   0 sleep :   0 spm   :   0 st    :   0 std   :   0 sts   :   5
sub   :   0 subi  :   0 swap  :   0 tst   :   0 wdr   :   0
Instructions used: 19 out of 113 (16.8%)

"ATmega324A" memory use summary [bytes]:
Segment   Begin    End      Code   Data   Used    Size   Use%
---------------------------------------------------------------
[.cseg] 0x000000 0x00008a    134      0    134   32768   0.4%
[.dseg] 0x000100 0x000102      0      2      2    2048   0.1%
[.eseg] 0x000000 0x000000      0      0      0    1024   0.0%

Assembly complete, 0 errors, 0 warnings

```