

```
1
2 AVRASM ver. 2.2.7 C:\Users\Seyi Olajuyi\Documents\Atmel Studio\7.0\pos_edge_ints\pos_edge_ints\
3 main.asm Thu Nov 21 20:36:43 2019
4
5 C:\Users\Seyi Olajuyi\Documents\Atmel Studio\7.0\pos_edge_ints\pos_edge_ints\main.asm(19):
6 Including file 'C:/Program Files (x86)\Atmel\Studio\7.0\Packs\atmel\ATmega_DFP\1.3.300\
7 avrasm\inc\m324adef.inc'
8 C:\Users\Seyi Olajuyi\Documents\Atmel Studio\7.0\pos_edge_ints\pos_edge_ints\main.asm(19):
9 Including file 'C:/Program Files (x86)\Atmel\Studio\7.0\Packs\atmel\ATmega_DFP\1.3.300\
10 avrasm\inc\m324adef.inc'
11
12
13          ;*
14          ;* Title:          pos_edge_ints
15          ;* Author:        Seyi Olajuyi & Bassel El Amine
16          ;* Version:       1.0
17          ;* Last updated:   11/21/19
18          ;* Target:        ATmega324A
19          ;*
20          ;* DESCRIPTION
21          ;* This program counts the number of times a key (any key) on the keypad
22          ;* is pressed and the number of times the pushbutton is pressed.
23          ;*
24          ;*
25          ;* VERSION HISTORY
26          ;* 1.0 Original version
27          ;*****
28
29          .list
30
31          .dseg
32 000100    key_presses:    .byte 1
33 000101    pb_presses:     .byte 1
34
35
```

```
36      .cseg
37      reset:
38      .org RESET      ;reset interrupt vector
39      000000 c004      rjmp start      ;program starts here at reset
40      .org INT0addr    ;INT0 interrupt vector
41      000002 c018      rjmp keypress_ISR
42      .org INT1addr
43      000004 c022      rjmp pb_press_ISR
44
45
46      start:
47      000005 9852      cbi DDRD, 2      ; Set pin 2 on PORTD to input
48      000006 9853      cbi DDRD, 3      ; Set pin 3 on PORTD to input
49
50      000007 9a0a      sbi DDRA, 2      ; Set pin 2 on PORTA to output
51      000008 9a3f      sbi DDRC, 7      ; Set pin 7 on PORTC to output
52
53      000009 980c      cbi DDRA, 4      ; Set pin 4 on PORTA to output
54
55      00000a e000      ldi r16, 0      ; Clear the variables
56      00000b 9300 0100 sts key_presses, r16
57      00000d 9300 0101 sts pb_presses, r16
58
59      00000f ef0f      ldi r16, LOW(RAMEND) ;initialize SP to point to top of stack
60      000010 bf0d      out SPL, r16
61      000011 e008      ldi r16, HIGH(RAMEND)
62      000012 bf0e      out SPH, r16
63
64      000013 e00f      ldi r16, (1 << ISC00) | (1 << ISC01) | (1 << ISC10) | (1 << ISC 11)
65      000014 9300 0069 sts EICRA, r16
66      000016 e003      ldi r16, $03      ; Enable interrupt request at INTO & INT1
67      000017 bb0d      out EIMSK, r16
68
69      000018 9478      sei      ;set global interrupt enable
70
```

```

71          main_loop:
72  000019 0000          nop                ;stub for background task
73  00001a cffe          rjmp main_loop      ;jump back to main_loop
74
75
76          ;*****
77          ;*
78          ;* "keypress_ISR" - Count Interrupts at INT0
79          ;*
80          ;* Description: Counts rising edges at INT0 (PD2)
81          ;*
82          ;* Author:          Ken Short
83          ;* Version:
84          ;* Last updated:    10/23/17
85          ;* Target:         ATmega324A
86          ;* Number of words:
87          ;* Number of cycles: 16
88          ;* Low registers modified: none
89          ;* High registers modified: none
90          ;*
91          ;* Parameters: Uses PORTB register to hold the count and drive LED s
92          ;* connected to that port.
93          ;*
94          ;* Notes:
95          ;*
96          ;*****
97
98          ;INT0 interrupt service routine
99  keypress_ISR:
100 00001b b70f          in r16, SREG          ;save SREG
101 00001c 930f          push r16
102
103 00001d 9100 0100          lds r16, key_presses          ;increment count
104 00001f 9503          inc r16
105 000020 9300 0100          sts key_presses, r16

```

```
106
107             restore_values_1:
108
109 000022 910f             pop r16             ;restore SREG
110 000023 bf0f             out SREG,r16
111
112 000024 9847             cbi PORTC, 7
113 000025 9a47             sbi PORTC, 7
114
115 000026 9518             reti             ;return from interrupt
116
117
118 ;*****
119 ;
120 ;* "pb_press_ISR" - Count Interrupts at INT1
121 ;*
122 ;* Description: Counts rising edges at INT1 (PD3)
123 ;*
124 ;* Author:             Ken Short
125 ;* Version:
126 ;* Last updated:       10/23/17
127 ;* Target:             ATmega324A
128 ;* Number of words:
129 ;* Number of cycles:   16
130 ;* Low registers modified: none
131 ;* High registers modified: none
132 ;*
133 ;* Parameters: Uses PORTB register to hold the count and drive LED s
134 ;* connected to that port.
135 ;*
136 ;* Notes:
137 ;*
138 ;*****
139
140             ;INT1 interrupt service routine
```

```

141          pb_press_ISR:
142          wait_for_bounce_1:
143 000027 9904          sbic PINA, 4
144 000028 cffe          rjmp wait_for_bounce_1
145 000029 e604          ldi r16, 100
146 00002a d010          rcall var_delay
147 00002b 9904          sbic PINA, 4
148 00002c cffa          rjmp wait_for_bounce_1
149
150 00002d e002          ldi r16, (1 <<INTF1)
151 00002e bb0c          out EIFR, r16
152 00002f 9812          cbi PORTA, 2          ; Clear Flip-Flop
153 000030 9a12          sbi PORTA, 2
154
155 000031 b70f          in r16, SREG          ;save SREG
156 000032 930f          push r16
157
158 000033 9100 0101          lds r16, pb_presses          ;increment count
159 000035 9503          inc r16
160 000036 9300 0101          sts pb_presses, r16
161
162          restore_value_2:
163 000038 910f          pop r16          ;restore SREG
164 000039 bf0f          out SREG,r16
165
166 00003a 9518          reti          ;return from interrupt
167
168          ;*****
169          ;SUBROUTINE FOR VAR DELAY
170          ;*****
171          var_delay:
172          outer_loop:
173 00003b e210          ldi r17, 32
174          inner_loop:
175 00003c 951a          dec r17

```

```

176 00003d f7f1          brne inner_loop
177 00003e 950a          dec r16
178 00003f f7d9          brne outer_loop
179
180
181 RESOURCE USE INFORMATION
182 -----
183
184 Notice:
185 The register and instruction counts are symbol table hit counts,
186 and hence implicitly used resources are not counted, eg, the
187 'lpm' instruction without operands implicitly uses r0 and z,
188 none of which are counted.
189
190 x,y,z are separate entities in the symbol table and are
191 counted separately from r26..r31 here.
192
193 .dseg memory usage only counts static data declared with .byte
194
195 "ATmega324A" register use summary:
196 x : 0 y : 0 z : 0 r0 : 0 r1 : 0 r2 : 0 r3 : 0 r4 : 0
197 r5 : 0 r6 : 0 r7 : 0 r8 : 0 r9 : 0 r10: 0 r11: 0 r12: 0
198 r13: 0 r14: 0 r15: 0 r16: 29 r17: 2 r18: 0 r19: 0 r20: 0
199 r21: 0 r22: 0 r23: 0 r24: 0 r25: 0 r26: 0 r27: 0 r28: 0
200 r29: 0 r30: 0 r31: 0
201 Registers used: 2 out of 35 (5.7%)
202
203 "ATmega324A" instruction use summary:
204 .lds : 0 .sts : 0 adc : 0 add : 0 adiw : 0 and : 0
205 andi : 0 asr : 0 bclr : 0 bld : 0 brbc : 0 brbs : 0
206 brcc : 0 brcs : 0 break : 0 breq : 0 brge : 0 brhc : 0
207 brhs : 0 brid : 0 brie : 0 brlo : 0 brlt : 0 brmi : 0
208 brne : 2 brpl : 0 brsh : 0 brtc : 0 brts : 0 brvc : 0
209 brvs : 0 bset : 0 bst : 0 call : 0 cbi : 5 cbr : 0
210 clc : 0 clh : 0 cli : 0 cln : 0 clr : 0 cls : 0

```

```

211 clt   : 0 clv   : 0 clz   : 0 com   : 0 cp    : 0 cpc   : 0
212 cpi   : 0 cpse  : 0 dec   : 2 eor   : 0 fmul  : 0 fmuls  : 0
213 fmulsu: 0 icall  : 0 ijmp  : 0 in    : 2 inc   : 2 jmp   : 0
214 ld    : 0 ldd   : 0 ldi   : 8 lds   : 2 lpm   : 0 lsl   : 0
215 lsr   : 0 mov   : 0 movw  : 0 mul   : 0 muls  : 0 mulsu  : 0
216 neg   : 0 nop   : 1 or    : 0 ori   : 0 out   : 6 pop   : 2
217 push  : 2 rcall : 1 ret   : 1 reti  : 2 rjmp  : 6 rol   : 0
218 ror   : 0 sbc   : 0 sbci  : 0 sbi   : 4 sbic  : 2 sbis  : 0
219 sbiw  : 0 sbr   : 0 sbrc  : 0 sbrs  : 0 sec   : 0 seh   : 0
220 sei   : 1 sen   : 0 ser   : 0 ses   : 0 set   : 0 sev   : 0
221 sez   : 0 sleep : 0 spm   : 0 st    : 0 std   : 0 sts   : 5
222 sub   : 0 subi  : 0 swap  : 0 tst   : 0 wdr   : 0

```

223 Instructions used: 19 out of 113 (16.8%)

224

225 "ATmega324A" memory use summary [bytes]:

Segment	Begin	End	Code	Data	Used	Size	Use%
227	-----						
228	[.cseg]	0x000000	0x000082	126	0	126	32768 0.4%
229	[.dseg]	0x000100	0x000102	0	2	2	2048 0.1%
230	[.eseg]	0x000000	0x000000	0	0	0	1024 0.0%

231

232 Assembly complete, 0 errors, 0 warnings

233