

```

1
2 ic_test_v2.elf:      file format elf32-avr
3
4 Sections:
5   Idx Name          Size      VMA      LMA      File off  Algn
6   0  .data          00000018  00804000  00000268  000002fc  2**0
7                      CONTENTS, ALLOC, LOAD, DATA
8   1  .text          00000268  00000000  00000000  00000094  2**1
9                      CONTENTS, ALLOC, LOAD, READONLY, CODE
10  2  .bss            00000002  00804018  00804018  00000314  2**0
11                      ALLOC
12  3  .comment        00000030  00000000  00000000  00000314  2**0
13                      CONTENTS, READONLY
14  4  .note.gnu.avr.deviceinfo 00000040  00000000  00000000  00000344  2**2
15                      CONTENTS, READONLY
16  5  .debug_aranges  00000020  00000000  00000000  00000384  2**0
17                      CONTENTS, READONLY, DEBUGGING
18  6  .debug_info     0000311e  00000000  00000000  000003a4  2**0
19                      CONTENTS, READONLY, DEBUGGING
20  7  .debug_abbrev   00002ddb  00000000  00000000  000034c2  2**0
21                      CONTENTS, READONLY, DEBUGGING
22  8  .debug_line     0000041d  00000000  00000000  0000629d  2**0
23                      CONTENTS, READONLY, DEBUGGING
24  9  .debug_frame    00000024  00000000  00000000  000066bc  2**2
25                      CONTENTS, READONLY, DEBUGGING
26 10  .debug_str       0000169c  00000000  00000000  000066e0  2**0
27                      CONTENTS, READONLY, DEBUGGING
28 11  .debug_loc      00000091  00000000  00000000  00007d7c  2**0
29                      CONTENTS, READONLY, DEBUGGING
30 12  .debug_ranges   00000010  00000000  00000000  00007e0d  2**0
31                      CONTENTS, READONLY, DEBUGGING
32
33 Disassembly of section .text:
34
35 00000000 <__vectors>:
36   0:  0c 94 7a 00      jmp 0xf4      ; 0xf4 <__ctors_end>
37   4:  0c 94 99 00      jmp 0x132     ; 0x132 <__bad_interrupt>
38   8:  0c 94 99 00      jmp 0x132     ; 0x132 <__bad_interrupt>
39   c:  0c 94 99 00      jmp 0x132     ; 0x132 <__bad_interrupt>
40  10:  0c 94 99 00      jmp 0x132     ; 0x132 <__bad_interrupt>
41  14:  0c 94 99 00      jmp 0x132     ; 0x132 <__bad_interrupt>
42  18:  0c 94 99 00      jmp 0x132     ; 0x132 <__bad_interrupt>
43  1c:  0c 94 99 00      jmp 0x132     ; 0x132 <__bad_interrupt>
44  20:  0c 94 99 00      jmp 0x132     ; 0x132 <__bad_interrupt>
45  24:  0c 94 99 00      jmp 0x132     ; 0x132 <__bad_interrupt>
46  28:  0c 94 99 00      jmp 0x132     ; 0x132 <__bad_interrupt>
47  2c:  0c 94 99 00      jmp 0x132     ; 0x132 <__bad_interrupt>
48  30:  0c 94 99 00      jmp 0x132     ; 0x132 <__bad_interrupt>
49  34:  0c 94 99 00      jmp 0x132     ; 0x132 <__bad_interrupt>

```

```
50 38: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
51 3c: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
52 40: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
53 44: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
54 48: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
55 4c: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
56 50: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
57 54: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
58 58: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
59 5c: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
60 60: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
61 64: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
62 68: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
63 6c: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
64 70: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
65 74: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
66 78: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
67 7c: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
68 80: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
69 84: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
70 88: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
71 8c: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
72 90: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
73 94: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
74 98: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
75 9c: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
76 a0: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
77 a4: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
78 a8: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
79 ac: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
80 b0: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
81 b4: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
82 b8: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
83 bc: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
84 c0: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
85 c4: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
86 c8: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
87 cc: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
88 d0: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
89 d4: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
90 d8: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
91 dc: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
92 e0: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
93 e4: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
94 e8: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
95 ec: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
96 f0: 0c 94 99 00 jmp 0x132 ; 0x132 <__bad_interrupt>
97
98 000000f4 <__ctors_end>:
```

```

99    f4:    11 24                eor r1, r1
100   f6:    1f be                out 0x3f, r1    ; 63
101   f8:    cf ef                ldi r28, 0xFF    ; 255
102   fa:    cd bf                out 0x3d, r28    ; 61
103   fc:    df e7                ldi r29, 0x7F    ; 127
104   fe:    de bf                out 0x3e, r29    ; 62
105
106 00000100 <__do_copy_data>:
107 100:    10 e4                ldi r17, 0x40    ; 64
108 102:    a0 e0                ldi r26, 0x00    ; 0
109 104:    b0 e4                ldi r27, 0x40    ; 64
110 106:    e8 e6                ldi r30, 0x68    ; 104
111 108:    f2 e0                ldi r31, 0x02    ; 2
112 10a:    00 e0                ldi r16, 0x00    ; 0
113 10c:    0b bf                out 0x3b, r16    ; 59
114 10e:    02 c0                rjmp    .+4      ; 0x114 <__do_copy_data+0x14>
115 110:    07 90                elpm    r0, Z+
116 112:    0d 92                st X+, r0
117 114:    a8 31                cpi r26, 0x18    ; 24
118 116:    b1 07                cpc r27, r17
119 118:    d9 f7                brne    .-10     ; 0x110 <__do_copy_data+0x10>
120
121 0000011a <__do_clear_bss>:
122 11a:    20 e4                ldi r18, 0x40    ; 64
123 11c:    a8 e1                ldi r26, 0x18    ; 24
124 11e:    b0 e4                ldi r27, 0x40    ; 64
125 120:    01 c0                rjmp    .+2      ; 0x124 <.do_clear_bss_start>
126
127 00000122 <.do_clear_bss_loop>:
128 122:    1d 92                st X+, r1
129
130 00000124 <.do_clear_bss_start>:
131 124:    aa 31                cpi r26, 0x1A    ; 26
132 126:    b2 07                cpc r27, r18
133 128:    e1 f7                brne    .-8      ; 0x122 <.do_clear_bss_loop>
134 12a:    0e 94 9b 00            call    0x136    ; 0x136 <main>
135 12e:    0c 94 32 01            jmp 0x264    ; 0x264 <_exit>
136
137 00000132 <__bad_interrupt>:
138 132:    0c 94 00 00            jmp 0    ; 0x0 <__vectors>
139
140 00000136 <main>:
141
142 uint8_t i;
143
144 int main(void)
145 {
146     PORTA_DIRSET = PA_setup_gm;
147 136:    8f e1                ldi r24, 0x1F    ; 31

```

```

148 138: 80 93 01 04      sts 0x0401, r24 ; 0x800401 <__TEXT_REGION_LENGTH__
      +0x7e0401>
149      PORTB_DIRSET = PB_setup_gm;
150 13c: 10 92 21 04      sts 0x0421, r1  ; 0x800421 <__TEXT_REGION_LENGTH__
      +0x7e0421>
151      PORTC_DIRSET = PC_setup_gm;
152 140: 10 92 41 04      sts 0x0441, r1  ; 0x800441 <__TEXT_REGION_LENGTH__
      +0x7e0441>
153      PORTD_DIRSET = PD_setup_gm;
154 144: 10 92 61 04      sts 0x0461, r1  ; 0x800461 <__TEXT_REGION_LENGTH__
      +0x7e0461>
155      PORTE_DIRSET = PE_setup_gm;
156 148: 87 e0            ldi r24, 0x07  ; 7
157 14a: 80 93 81 04      sts 0x0481, r24 ; 0x800481 <__TEXT_REGION_LENGTH__
      +0x7e0481>
158      PORTF_DIRSET = PF_setup_gm;
159 14e: 85 e0            ldi r24, 0x05  ; 5
160 150: 80 93 a1 04      sts 0x04A1, r24 ; 0x8004a1 <__TEXT_REGION_LENGTH__
      +0x7e04a1>
161
162      PORTA_PIN7CTRL = PORT_PULLUPEN_bm;
163 154: 88 e0            ldi r24, 0x08  ; 8
164 156: 80 93 17 04      sts 0x0417, r24 ; 0x800417 <__TEXT_REGION_LENGTH__
      +0x7e0417>
165      PORTA_PIN6CTRL = PORT_PULLUPEN_bm;
166 15a: 80 93 16 04      sts 0x0416, r24 ; 0x800416 <__TEXT_REGION_LENGTH__
      +0x7e0416>
167      PORTA_PIN5CTRL = PORT_PULLUPEN_bm;
168 15e: 80 93 15 04      sts 0x0415, r24 ; 0x800415 <__TEXT_REGION_LENGTH__
      +0x7e0415>
169      PORTA_PIN4CTRL = PORT_PULLUPEN_bm;
170 162: 80 93 14 04      sts 0x0414, r24 ; 0x800414 <__TEXT_REGION_LENGTH__
      +0x7e0414>
171      PORTA_PIN3CTRL = PORT_PULLUPEN_bm;
172 166: 80 93 13 04      sts 0x0413, r24 ; 0x800413 <__TEXT_REGION_LENGTH__
      +0x7e0413>
173
174      PORTD_OUT &= ~(BARGRAPH_gm | TIP_bm | PASS_bm | FAIL_bm);
175 16a: e4 e6            ldi r30, 0x64  ; 100
176 16c: f4 e0            ldi r31, 0x04  ; 4
177 16e: 80 81            ld  r24, Z
178 170: 87 70            andi  r24, 0x07  ; 7
179 172: 80 83            st  Z, r24
180      #else
181          //round up by default
182          __ticks_dc = (uint32_t)(ceil(fabs(__tmp)));
183      #endif
184
185      __builtin_avr_delay_cycles(__ticks_dc);

```

```

186 174: 2f ef      ldi r18, 0xFF      ; 255
187 176: 84 e3      ldi r24, 0x34      ; 52
188 178: 9c e0      ldi r25, 0x0C      ; 12
189 17a: 21 50      subi  r18, 0x01     ; 1
190 17c: 80 40      sbci  r24, 0x00     ; 0
191 17e: 90 40      sbci  r25, 0x00     ; 0
192 180: e1 f7      brne  .-8           ; 0x17a <main+0x44>
193 182: 00 c0      rjmp  .+0           ; 0x184 <main+0x4e>
194 184: 00 00      nop
195      _delay_ms(1000);
196      PORTD_OUT = BARGRAPH_gm | TIP_bm | PASS_bm | FAIL_bm;
197 186: 88 ef      ldi r24, 0xF8      ; 248
198 188: 80 83      st  Z, r24
199
200      while (1)
201      {
202          while (!(PORTA_IN & START_PB_bm)) {}
203 18a: 80 91 08 04  lds r24, 0x0408 ; 0x800408 <__TEXT_REGION_LENGTH__
+0x7e0408>
204 18e: 84 ff      sbrs  r24, 4
205 190: fc cf      rjmp  .-8           ; 0x18a <main+0x54>
206          while (PORTA_IN & START_PB_bm) {}
207 192: 80 91 08 04  lds r24, 0x0408 ; 0x800408 <__TEXT_REGION_LENGTH__
+0x7e0408>
208 196: 84 fd      sbrc  r24, 4
209 198: fc cf      rjmp  .-8           ; 0x192 <main+0x5c>
210
211          PORTD_OUT = BARGRAPH_gm | TIP_bm | PASS_bm | FAIL_bm;
212 19a: e4 e6      ldi r30, 0x64      ; 100
213 19c: f4 e0      ldi r31, 0x04      ; 4
214 19e: 88 ef      ldi r24, 0xF8      ; 248
215 1a0: 80 83      st  Z, r24
216
217          PORTD_OUT &= ~TIP_bm;
218 1a2: 80 81      ld  r24, Z
219 1a4: 8f 7e      andi  r24, 0xEF     ; 239
220 1a6: 80 83      st  Z, r24
221
222          gate_type = PORTA_IN >> 5;
223 1a8: a8 e0      ldi r26, 0x08      ; 8
224 1aa: b4 e0      ldi r27, 0x04      ; 4
225 1ac: 8c 91      ld  r24, X
226 1ae: 82 95      swap r24
227 1b0: 86 95      lsr  r24
228 1b2: 87 70      andi  r24, 0x07     ; 7
229 1b4: 80 93 18 40  sts 0x4018, r24 ; 0x804018 <__data_end>
230
231          PORTD_OUT &= ~(PORTA_IN & DIP_SW_gm);
232 1b8: 8c 91      ld  r24, X

```

```

233 1ba: 80 7e      andi    r24, 0xE0    ; 224
234 1bc: 90 e0      ldi    r25, 0x00    ; 0
235 1be: 80 95      com    r24
236 1c0: 90 95      com    r25
237 1c2: 90 81      ld     r25, Z
238 1c4: 89 23      and    r24, r25
239 1c6: 80 83      st     Z, r24
240
241          if (gate_type == 4) {
242 1c8: 80 91 18 40 lds    r24, 0x4018 ; 0x804018 <__data_end>
243 1cc: 84 30      cpi    r24, 0x04    ; 4
244 1ce: 49 f4      brne   .+18        ; 0x1e2 <main+0xac>
245          //enable pullups
246          PORTA_PIN7CTRL = PORT_PULLUPEN_bm;
247 1d0: 88 e0      ldi    r24, 0x08    ; 8
248 1d2: 80 93 17 04 sts    0x0417, r24 ; 0x800417 <__TEXT_REGION_LENGTH__
+0x7e0417>
249          PORTA_PIN6CTRL = PORT_PULLUPEN_bm;
250 1d6: 80 93 16 04 sts    0x0416, r24 ; 0x800416 <__TEXT_REGION_LENGTH__
+0x7e0416>
251          PORTA_PIN5CTRL = PORT_PULLUPEN_bm;
252 1da: 80 93 15 04 sts    0x0415, r24 ; 0x800415 <__TEXT_REGION_LENGTH__
+0x7e0415>
253          PORTA_PIN4CTRL = PORT_PULLUPEN_bm;
254 1de: 80 93 14 04 sts    0x0414, r24 ; 0x800414 <__TEXT_REGION_LENGTH__
+0x7e0414>
255      }
256
257          //turn DUT pin 14 on
258          PORTE_OUT |= PIN3_bm;
259 1e2: e4 e8      ldi    r30, 0x84    ; 132
260 1e4: f4 e0      ldi    r31, 0x04    ; 4
261 1e6: 80 81      ld     r24, Z
262 1e8: 88 60      ori    r24, 0x08    ; 8
263 1ea: 80 83      st     Z, r24
264
265          for (uint8_t i = 0; i < 4; ++i) {
266 1ec: 40 e0      ldi    r20, 0x00    ; 0
267 1ee: 20 c0      rjmp   .+64        ; 0x230 <__EEPROM_REGION_LENGTH__
+0x30>
268          PORTC_OUT = stimulus[i];
269 1f0: 24 2f      mov    r18, r20
270 1f2: 30 e0      ldi    r19, 0x00    ; 0
271 1f4: f9 01      movw   r30, r18
272 1f6: ec 5e      subi   r30, 0xEC    ; 236
273 1f8: ff 4b      sbci   r31, 0xBF    ; 191
274 1fa: 80 81      ld     r24, Z
275 1fc: 80 93 44 04 sts    0x0444, r24 ; 0x800444 <__TEXT_REGION_LENGTH__
+0x7e0444>

```

```

276     can be achieved.
277 */
278 void
279 _delay_loop_1(uint8_t __count)
280 {
281     __asm__ volatile (
282 200: 82 e0          ldi r24, 0x02    ; 2
283 202: 8a 95          dec r24
284 204: f1 f7          brne    .-4      ; 0x202 <__EEPROM_REGION_LENGTH__  ↗
        +0x2>
285
286         _delay_loop_1(2);
287
288         if (!((PORTE_IN & GATES_OUT_gm) == verify[gate_type][i])) break;
289 206: 50 91 88 04      lds r21, 0x0488 ; 0x800488 <__TEXT_REGION_LENGTH__  ↗
        +0x7e0488>
290 20a: 90 91 18 40      lds r25, 0x4018 ; 0x804018 <__data_end>
291 20e: 89 2f             mov r24, r25
292 210: 90 e0             ldi r25, 0x00    ; 0
293 212: 88 0f             add r24, r24
294 214: 99 1f             adc r25, r25
295 216: 88 0f             add r24, r24
296 218: 99 1f             adc r25, r25
297 21a: 80 50             subi    r24, 0x00    ; 0
298 21c: 90 4c             sbci    r25, 0xC0    ; 192
299 21e: fc 01             movw    r30, r24
300 220: e2 0f             add r30, r18
301 222: f3 1f             adc r31, r19
302 224: 90 81             ld r25, Z
303 226: 85 2f             mov r24, r21
304 228: 8f 70             andi    r24, 0x0F    ; 15
305 22a: 89 13             cpse    r24, r25
306 22c: 03 c0             rjmp     .+6      ; 0x234 <__EEPROM_REGION_LENGTH__  ↗
        +0x34>
307     }
308
309     //turn DUT pin 14 on
310     PORTE_OUT |= PIN3_bm;
311
312     for (uint8_t i = 0; i < 4; ++i) {
313 22e: 4f 5f             subi    r20, 0xFF    ; 255
314 230: 44 30             cpi r20, 0x04    ; 4
315 232: f0 f2             brcs    .-68      ; 0x1f0 <main+0xba>
        _delay_loop_1(2);
316
317         if (!((PORTE_IN & GATES_OUT_gm) == verify[gate_type][i])) break;
318     }
319
320     PORTD_OUT |= TIP_bm;
321

```

```

322 234: e4 e6      ldi r30, 0x64      ; 100
323 236: f4 e0      ldi r31, 0x04      ; 4
324 238: 80 81      ld r24, Z
325 23a: 80 61      ori r24, 0x10      ; 16
326 23c: 80 83      st Z, r24
327
328          if (i == 4) PORTD_OUT &= ~PASS_bm;
329 23e: 80 91 19 40 lds r24, 0x4019 ; 0x804019 <i>
330 242: 84 30      cpi r24, 0x04      ; 4
331 244: 21 f4      brne .+8          ; 0x24e <__EEPROM_REGION_LENGTH__
+0x4e>
332 246: 80 81      ld r24, Z
333 248: 87 7f      andi r24, 0xF7      ; 247
334 24a: 80 83      st Z, r24
335 24c: 05 c0      rjmp .+10          ; 0x258 <__EEPROM_REGION_LENGTH__
+0x58>
336          else PORTD_OUT &= ~FAIL_bm;
337 24e: e4 e6      ldi r30, 0x64      ; 100
338 250: f4 e0      ldi r31, 0x04      ; 4
339 252: 80 81      ld r24, Z
340 254: 8f 7b      andi r24, 0xBF      ; 191
341 256: 80 83      st Z, r24
342
343          //turn DUT pin 14 off
344          PORTE_OUT &= ~PIN3_bm;
345 258: e4 e8      ldi r30, 0x84      ; 132
346 25a: f4 e0      ldi r31, 0x04      ; 4
347 25c: 80 81      ld r24, Z
348 25e: 87 7f      andi r24, 0xF7      ; 247
349 260: 80 83      st Z, r24
350      }
351 262: 93 cf      rjmp .-218      ; 0x18a <main+0x54>
352
353 00000264 <_exit>:
354 264: f8 94      cli
355
356 00000266 <__stop_program>:
357 266: ff cf      rjmp .-2          ; 0x266 <__stop_program>
358

```