

```

1
2 clk_main_32768Hz.elf:      file format elf32-avr
3
4 Sections:
5  Idx Name                  Size      VMA      LMA      File off  Algn
6    0 .data                 00000000  00804000  00804000  0000018c  2**0
7                          CONTENTS, ALLOC, LOAD, DATA
8    1 .text                 00000138  00000000  00000000  00000054  2**1
9                          CONTENTS, ALLOC, LOAD, READONLY, CODE
10   2 .comment              00000030  00000000  00000000  0000018c  2**0
11                          CONTENTS, READONLY
12   3 .note.gnu.avr.deviceinfo 00000040  00000000  00000000  00000000  000001bc  2**2
13                          CONTENTS, READONLY
14   4 .debug_aranges        00000020  00000000  00000000  000001fc  2**0
15                          CONTENTS, READONLY, DEBUGGING
16   5 .debug_info           00003403  00000000  00000000  0000021c  2**0
17                          CONTENTS, READONLY, DEBUGGING
18   6 .debug_abbrev         00002d85  00000000  00000000  0000361f  2**0
19                          CONTENTS, READONLY, DEBUGGING
20   7 .debug_line           00000346  00000000  00000000  000063a4  2**0
21                          CONTENTS, READONLY, DEBUGGING
22   8 .debug_frame          00000024  00000000  00000000  000066ec  2**2
23                          CONTENTS, READONLY, DEBUGGING
24   9 .debug_str            000019f0  00000000  00000000  00006710  2**0
25                          CONTENTS, READONLY, DEBUGGING
26  10 .debug_ranges         00000010  00000000  00000000  00008100  2**0
27                          CONTENTS, READONLY, DEBUGGING
28
29 Disassembly of section .text:
30
31 00000000 <__vectors>:
32  0: 0c 94 7a 00      jmp 0xf4      ; 0xf4 <__ctors_end>
33  4: 0c 94 84 00      jmp 0x108     ; 0x108 <__bad_interrupt>
34  8: 0c 94 84 00      jmp 0x108     ; 0x108 <__bad_interrupt>
35  c: 0c 94 84 00      jmp 0x108     ; 0x108 <__bad_interrupt>
36 10: 0c 94 84 00      jmp 0x108     ; 0x108 <__bad_interrupt>
37 14: 0c 94 84 00      jmp 0x108     ; 0x108 <__bad_interrupt>
38 18: 0c 94 84 00      jmp 0x108     ; 0x108 <__bad_interrupt>
39 1c: 0c 94 84 00      jmp 0x108     ; 0x108 <__bad_interrupt>
40 20: 0c 94 84 00      jmp 0x108     ; 0x108 <__bad_interrupt>
41 24: 0c 94 84 00      jmp 0x108     ; 0x108 <__bad_interrupt>
42 28: 0c 94 84 00      jmp 0x108     ; 0x108 <__bad_interrupt>
43 2c: 0c 94 84 00      jmp 0x108     ; 0x108 <__bad_interrupt>
44 30: 0c 94 84 00      jmp 0x108     ; 0x108 <__bad_interrupt>
45 34: 0c 94 84 00      jmp 0x108     ; 0x108 <__bad_interrupt>
46 38: 0c 94 84 00      jmp 0x108     ; 0x108 <__bad_interrupt>
47 3c: 0c 94 84 00      jmp 0x108     ; 0x108 <__bad_interrupt>
48 40: 0c 94 84 00      jmp 0x108     ; 0x108 <__bad_interrupt>
49 44: 0c 94 84 00      jmp 0x108     ; 0x108 <__bad_interrupt>

```

```

50 48: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
51 4c: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
52 50: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
53 54: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
54 58: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
55 5c: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
56 60: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
57 64: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
58 68: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
59 6c: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
60 70: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
61 74: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
62 78: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
63 7c: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
64 80: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
65 84: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
66 88: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
67 8c: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
68 90: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
69 94: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
70 98: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
71 9c: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
72 a0: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
73 a4: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
74 a8: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
75 ac: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
76 b0: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
77 b4: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
78 b8: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
79 bc: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
80 c0: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
81 c4: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
82 c8: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
83 cc: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
84 d0: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
85 d4: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
86 d8: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
87 dc: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
88 e0: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
89 e4: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
90 e8: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
91 ec: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
92 f0: 0c 94 84 00 jmp 0x108 ; 0x108 <__bad_interrupt>
93
94 000000f4 <__ctors_end>:
95 f4: 11 24 eor r1, r1
96 f6: 1f be out 0x3f, r1 ; 63
97 f8: cf ef ldi r28, 0xFF ; 255
98 fa: cd bf out 0x3d, r28 ; 61

```

```

99  fc:  df e7          ldi r29, 0x7F    ; 127
100 fe:  de bf          out 0x3e, r29    ; 62
101 100: 0e 94 86 00    call 0x10c      ; 0x10c <main>
102 104: 0c 94 9a 00    jmp 0x134    ; 0x134 <_exit>
103
104 00000108 <__bad_interrupt>:
105 108: 0c 94 00 00    jmp 0      ; 0x0 <__vectors>
106
107 0000010c <main>:
108
109 #define CLKCTRL_PDIV_disable 0x00
    // mask for disable prescaler.
110
111 int main(void)
112 {
113     PORTA.DIRSET = PIN7_bm;
    // enable output on pin 7.
114 10c: 80 e8          ldi r24, 0x80    ; 128
115 10e: 80 93 01 04    sts 0x0401, r24 ; 0x800401 <__TEXT_REGION_LENGTH__
    +0x7e0401>
116
117     CPU_CCP = CCP_IOREG_gc;
118 112: 88 ed          ldi r24, 0xD8    ; 216
119 114: 84 bf          out 0x34, r24    ; 52
120     CLKCTRL.MCLKCTRLA = CLKCTRL_CLKOUT_bm | CLKCTRL_CLKSEL_OSC32K_gc;
    // main clock enabled on CLKOUT, main clock set to internal 32.768
    kHz oscillator.
121 116: e0 e6          ldi r30, 0x60    ; 96
122 118: f0 e0          ldi r31, 0x00    ; 0
123 11a: 91 e8          ldi r25, 0x81    ; 129
124 11c: 90 83          st Z, r25
125     CPU_CCP = CCP_IOREG_gc;
126 11e: 84 bf          out 0x34, r24    ; 52
127     CLKCTRL.MCLKCTRLB = CLKCTRL_PDIV_disable;
    // prescaler division disabled.
128 120: 11 82          std Z+1, r1 ; 0x01
129     CPU_CCP = CCP_IOREG_gc;
130 122: 84 bf          out 0x34, r24    ; 52
131     CLKCTRL.MCLKCTRLC = CLKCTRL_CFD SRC_CLKMAIN_gc | CLKCTRL_CFDEN_bm;
    // clock failure source set to main clock, clock failure detection
    disabled.
132 124: 91 e0          ldi r25, 0x01    ; 1
133 126: 92 83          std Z+2, r25    ; 0x02
134     CPU_CCP = CCP_IOREG_gc;
135 128: 84 bf          out 0x34, r24    ; 52
136     CLKCTRL.MCLKINTCTRL = CLKCTRL_INTTYPE_INT_gc;
    // regular interrupt type.
137 12a: 13 82          std Z+3, r1 ; 0x03
138     CPU_CCP = CCP_IOREG_gc;

```

```
139 12c: 84 bf          out 0x34, r24      ; 52
140      CLKCTRL.OSCHFCTRLA = CLKCTRL_RUNSTBY_bm | CLKCTRL_FREQSEL_4M_gc |
      CLKCTRL_AUTOTUNE_bm;          // run on standby enabled, autotune
      enabled.
141 12e: 8d e8          ldi r24, 0x8D      ; 141
142 130: 80 87          std Z+8, r24      ; 0x08
143 132: ff cf          rjmp    .-2          ; 0x132 <main+0x26>
144
145 00000134 <_exit>:
146 134: f8 94          cli
147
148 00000136 <__stop_program>:
149 136: ff cf          rjmp    .-2          ; 0x136 <__stop_program>
150
```