```
toggle_every_xxx_us.elf:     file format elf32-avr

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .data         00000000  00804000  00804000  0000019c  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  1 .text         00000148  00000000  00000000  00000054  2**1
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .comment      00000030  00000000  00000000  0000019c  2**0
                  CONTENTS, READONLY
  3 .note.gnu.avr.deviceinfo 00000040  00000000  00000000  000001cc  2**2
                  CONTENTS, READONLY
  4 .debug_aranges 00000020  00000000  00000000  0000020c  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_info   00003021  00000000  00000000  0000022c  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_abbrev 00002d5f  00000000  00000000  0000324d  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_line   00000348  00000000  00000000  00005fac  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .debug_frame  00000024  00000000  00000000  000062f4  2**2
                  CONTENTS, READONLY, DEBUGGING
  9 .debug_str    00001663  00000000  00000000  00006318  2**0
                  CONTENTS, READONLY, DEBUGGING
 10 .debug_loc    0000002a  00000000  00000000  0000797b  2**0
                  CONTENTS, READONLY, DEBUGGING
 11 .debug_ranges 00000010  00000000  00000000  000079a5  2**0
                  CONTENTS, READONLY, DEBUGGING

Disassembly of section .text:

00000000 <__vectors>:
   0:   0c 94 7a 00     jmp 0xf4    ; 0xf4 <__ctors_end>
   4:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
   8:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
   c:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  10:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  14:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  18:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  1c:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  20:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  24:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  28:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  2c:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  30:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  34:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  38:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  3c:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  40:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  44:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  48:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
```

```
  4c:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  50:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  54:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  58:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  5c:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  60:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  64:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  68:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  6c:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  70:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  74:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  78:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  7c:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  80:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  84:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  88:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  8c:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  90:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  94:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  98:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  9c:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  a0:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  a4:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  a8:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  ac:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  b0:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  b4:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  b8:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  bc:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  c0:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  c4:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  c8:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  cc:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  d0:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  d4:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  d8:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  dc:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  e0:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  e4:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  e8:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  ec:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>
  f0:   0c 94 84 00     jmp 0x108   ; 0x108 <__bad_interrupt>

000000f4 <__ctors_end>:
  f4:   11 24           eor r1, r1
  f6:   1f be           out 0x3f, r1    ; 63
  f8:   cf ef           ldi r28, 0xFF   ; 255
  fa:   cd bf           out 0x3d, r28   ; 61
  fc:   df e7           ldi r29, 0x7F   ; 127
  fe:   de bf           out 0x3e, r29   ; 62
 100:   0e 94 86 00     call    0x10c   ; 0x10c <main>
 104:   0c 94 a2 00     jmp 0x144   ; 0x144 <_exit>
```

```
00000108 <__bad_interrupt>:
 108:   0c 94 00 00     jmp 0   ; 0x0 <__vectors>

0000010c <main>:

#define PULL_UP_VAL 0x08

int main(void)
{
    PORTC_DIR = PIN3_bm;  //Configure PC3 as the output
 10c:   88 e0           ldi r24, 0x08   ; 8
 10e:   80 93 40 04     sts 0x0440, r24 ; 0x800440 <__TEXT_REGION_LENGTH__+0x7e0440>

    PORTC_PIN0CTRL = PULL_UP_VAL;    //Enable internal pull up resistor for PC0
 112:   80 93 50 04     sts 0x0450, r24 ; 0x800450 <__TEXT_REGION_LENGTH__+0x7e0450>
    PORTC_PIN1CTRL = PULL_UP_VAL;   //Enable internal pull up resistor for PC1
 116:   80 93 51 04     sts 0x0451, r24 ; 0x800451 <__TEXT_REGION_LENGTH__+0x7e0451>

    while (1)
    {
        PORTC_OUT |= PIN3_bm;   //Set PC3, XOR
 11a:   a4 e4           ldi r26, 0x44   ; 68
 11c:   b4 e0           ldi r27, 0x04   ; 4
 11e:   8c 91           ld  r24, X
 120:   88 60           ori r24, 0x08   ; 8
 122:   8c 93           st  X, r24
        //For two bits to be used for the DIP SWITCH
        //PC0 = 0 and PC1 = 0 for DIP SWITCH


        if((~(PIN0_bm & PORTC_IN)) && (~(PIN1_bm & PORTC_IN))){
 124:   e8 e4           ldi r30, 0x48   ; 72
 126:   f4 e0           ldi r31, 0x04   ; 4
 128:   80 81           ld  r24, Z
 12a:   80 81           ld  r24, Z
    can be achieved.
*/
void
_delay_loop_1(uint8_t __count)
{
    __asm__ volatile (
 12c:   81 e2           ldi r24, 0x21   ; 33
 12e:   98 2f           mov r25, r24
 130:   9a 95           dec r25
 132:   f1 f7           brne    .-4         ; 0x130 <main+0x24>
        else if(((PIN0_bm & PORTC_IN)) && (~(PIN1_bm & PORTC_IN))){
            _delay_loop_1(137);
        }


        PORTC_OUT &= ~PIN3_bm;   //Clear PC3
```

```
 134:   9c 91           ld  r25, X
 136:   97 7f           andi    r25, 0xF7   ; 247
 138:   9c 93           st  X, r25

        //For two bits to be used for the DIP SWITCH
        //PC0 = 0 and PC1 = 0 for DIP SWITCH
        if((~(PIN0_bm & PORTC_IN)) && (~(PIN1_bm & PORTC_IN))){
 13a:   90 81           ld  r25, Z
 13c:   90 81           ld  r25, Z
 13e:   8a 95           dec r24
 140:   f1 f7           brne    .-4         ; 0x13e <main+0x32>
 142:   eb cf           rjmp    .-42        ; 0x11a <main+0xe>

00000144 <_exit>:
 144:   f8 94           cli

00000146 <__stop_program>:
 146:   ff cf           rjmp    .-2         ; 0x146 <__stop_program>
```