```c
/*
 * interrupt_echo.c
 *
 * Created: 3/18/2021 7:00:09 PM
 * Author : Judah Ben-Eliezer
 */

#define BAUD_RATE 4800UL                                                    //  ↵
  baud rate.
#define F_CPU 4000000UL                                                     //  ↵
  clock at 4 MHz.

#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>

uint8_t USART_sw_read();                                                    //  ↵
  read function declaration.
void USART_sw_write(char);                                                  //  ↵
  write function declaration.

char c;

int main(void)
{
    PORTB.DIRCLR = PIN1_bm;                                                 //  ↵
      set PB1 as input.
    PORTB.PIN1CTRL |= PORT_ISC_FALLING_gc;                                  //  ↵
      enable interrupt on falling edge of PB1.
    sei();                                                                  //  ↵
      enable global interrupts.

    while (1)
    {
        asm volatile ("nop");                                               //  ↵
          nop to avoid optimization deletion of while loop.
    }
}

ISR (PORTB_PORT_vect) {
    c = USART_sw_read();                                                    //  ↵
      call USART_sw_read.
    USART_sw_write(c - 0x20);                                               //  ↵
      write uppercase c.
    PORTB.INTFLAGS |= PIN1_bm;                                              //  ↵
      clear interrupt.
}

uint8_t USART_sw_read() {
```

```
39
40      uint8_t d;                                                      //  ↵
          bit time.
41      if (BAUD_RATE == 4800UL) {
42          d = 48;
43      } else if (BAUD_RATE == 9600UL) {
44          d = 99;
45      } else if (BAUD_RATE == 19200UL) {
46          d = 201;
47      } else return 0x00;
48
49      uint8_t data = 0;
50
51      _delay_us(d/2);
52      if ((PORTB_IN & PIN1_bm) != 0) return 0x00;                     //  ↵
          check for false start.
53      _delay_us(d);                                                   //  ↵
          delay for bit time.
54
55      uint8_t i;
56      for (i = 0; i < 8; ++i) {
57          data >>= data | ((PORTB_IN | PIN1_bm) << 6);                //  ↵
            read little endian input into data.
58          _delay_us(d);                                               //  ↵
            delay for bit time.
59      }
60
61      return data;
62  }
63
64  void USART_sw_write(char c) {
65      PORTB.DIRSET = PIN0_bm;                                         //  ↵
          set PB0 as output.
66      uint8_t d;                                                      //  ↵
          bit time.
67      if (BAUD_RATE == 4800L) {
68          d = 48;
69      } else if (BAUD_RATE == 9600L) {
70          d = 99;
71      } else if (BAUD_RATE == 19200L) {
72          d = 201;
73      } else return;
74
75      uint8_t data = (uint8_t) c;
76
77      PORTB_OUT = 0x00 | PIN0_bm;                                     //  ↵
          send start bit.
78      _delay_us(d);                                                   //  ↵
          delay for bit time.
```

```
79
80      uint8_t i;
81      for (i = 0; i < 8; ++i) {
82          PORTB_OUT = data | PIN0_bm;                                    //  ↪
              send lsb of data.
83          data >>= data;                                                 //  ↪
              shift data right.
84          _delay_us(d);                                                  //  ↪
              delay for bit time.
85      }
86
87      PORTB_OUT = PIN0_bm;                                               //  ↪
          send end bit.
88      _delay_us(d);                                                      //  ↪
          delay for bit time.
89  }
90
91
```