```
    REG_PORT_DIRSET1 = 0x80;        /* PB07 = CS for BME680 */

    REG_SERCOM1_SPI_CTRLA = 1;              /* reset SERCOM1 */
    while (REG_SERCOM1_SPI_CTRLA & 1) {}    /* wait for reset to complete */

    REG_SERCOM1_SPI_CTRLA = 0x3030000C;     /* MISO-3, MOSI-0, SCK-1, SS-2, CPOL=1, CPHA=1 */
    REG_SERCOM1_SPI_CTRLB = 0x00022000;     /* Master SS, 8-bit, receiver enabled */
    REG_SERCOM1_SPI_BAUD = 0;               /* SPI clock is 4MHz/2 = 2MzHz */
    REG_SERCOM1_SPI_CTRLA |= 2;             /* enable SERCOM1 */
}

static void init_spi_BME680 (void) {
    user_spi_write (0, 0x60, (void *)0xB6, 1);      //software reset BME680
    user_spi_write(0, 0x73, (void *)0, 1);          //switch to page 0 of memory map
    user_spi_read(0, 0x73, &status, 1);     //read status register
    user_spi_read(0, 0x50, &id, 1);         //read id register
    user_spi_write(0, 0x73, (void *)0x10, 1);               //switch to page 1 of memory map
    user_spi_read(0, 0x73, &status, 1);     //read status register
}

void user_delay_ms (uint32_t period) {
    for (int i = 0; i < 170*period; i++) {      //based off of 30us delay in DOGM163W_A_SERCOM1.c
        __asm("nop");                           //delay by period ms
    }
}

static uint8_t spi_transfer (uint8_t data) {
    uint8_t Rx_data;
    while(!(REG_SERCOM1_SPI_INTFLAG & 1)) {}        //wait until Tx ready
    REG_SERCOM1_SPI_DATA = data;                    //send data byte
    while(!(REG_SERCOM1_SPI_INTFLAG & 2)) {}        //wait until transmit is complete
    while(!(REG_SERCOM1_SPI_INTFLAG & 4)) {}        //wait until receive is complete
    Rx_data = REG_SERCOM1_SPI_DATA;                 //read data register
    return Rx_data;
}

int8_t user_spi_read (uint8_t dev_id, uint8_t reg_addr, uint8_t *reg_data, uint16_t len) {
```