ESE 381

Lab 10: Air Monitoring System I - Basic Operation of SCD41 CO2, Humidity, and Temperature Sensor

4/25/2025

Muhammad Sharjeel and Naafiul Hossain

115185427

115107623

Section: L02 Bench 7

Group 16

2) What is the maximum specified serial clock speed at which the SCD41 can be operated? What is the maximum speed and which the SCD41 can be operated in this design? Show your calculations for both.

The maximum serial clock speed specified in the SCD41 datasheet is **400kHz**. *In normal*, the maximum speed can be *100kHz*

The maximum speed can be calculated using the formula found in the data sheet:

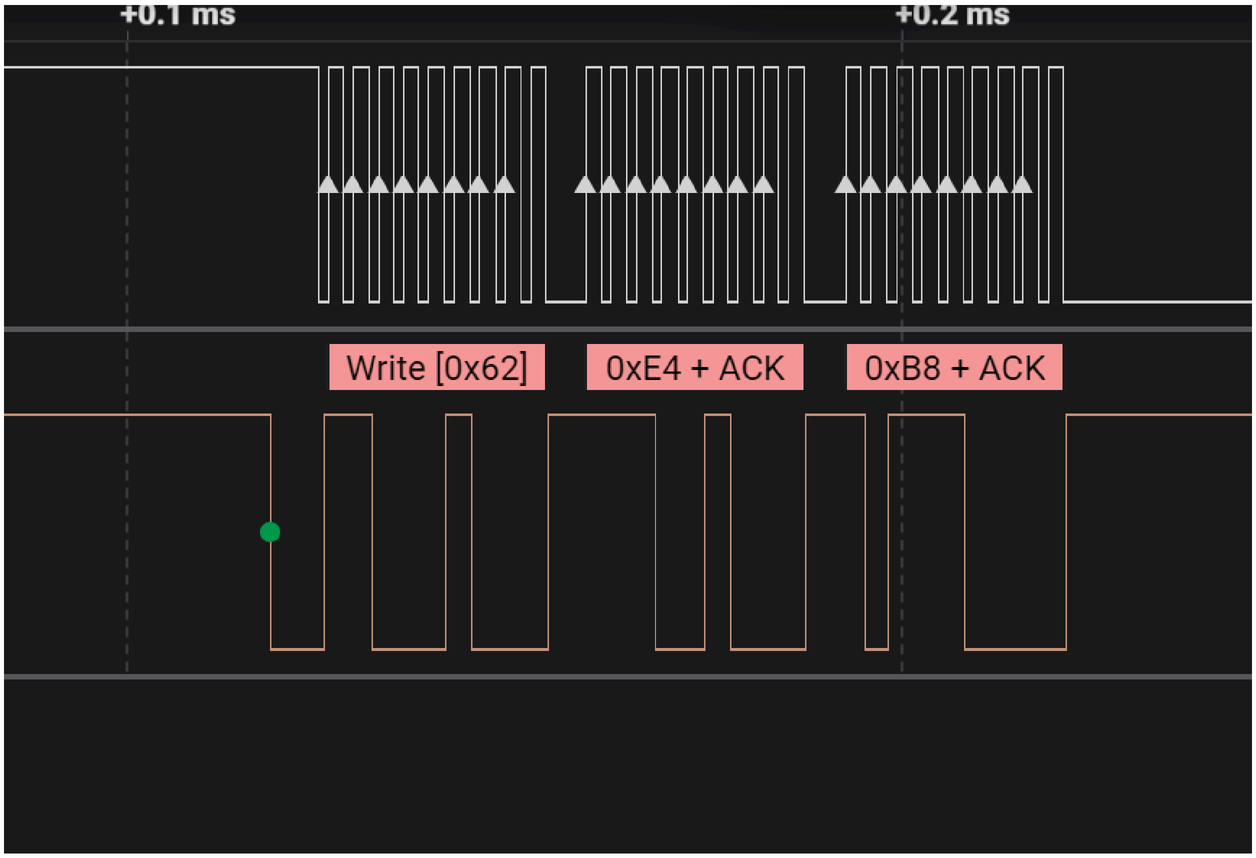
$$f_{SCL} = \frac{f_{CLK\_PER}}{10 + 2 \times BAUD + f_{CLK\_PER} \times T_R}$$

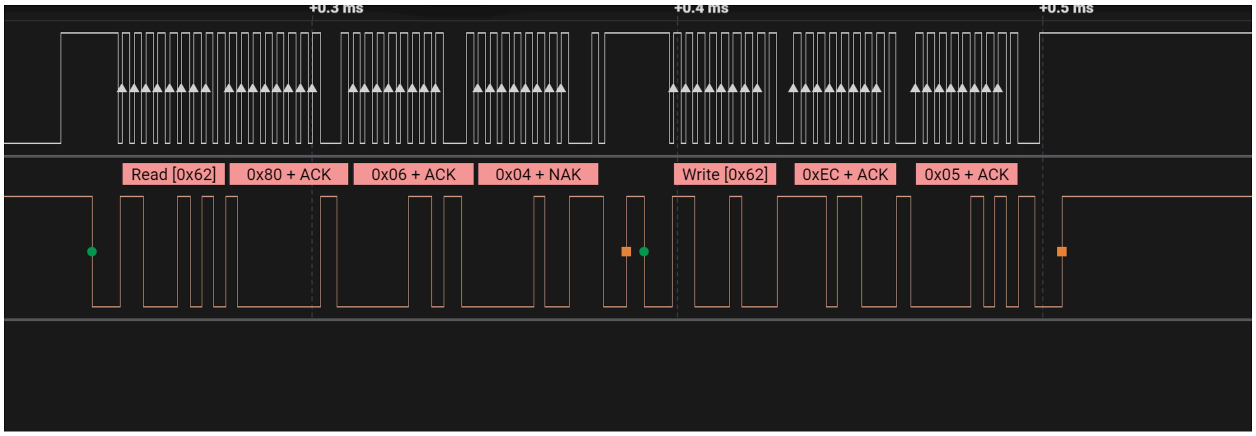
Thus, in our design, with F_CPU = 4 MHz and TWI0.MBAUD = 1, the actual SCL speed is:

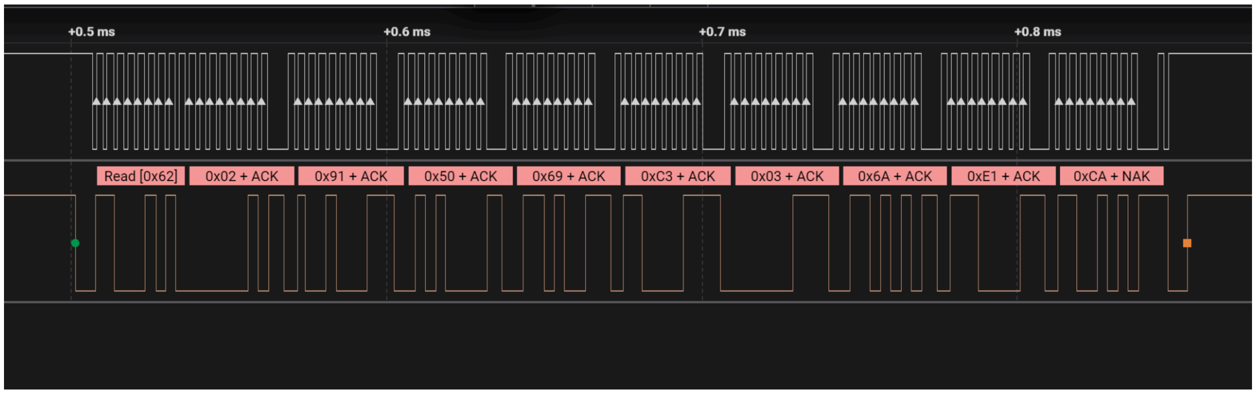
$$f_{SCL} = \frac{4{,}000{,}000}{10 + 2(1) + 4{,}000{,}000 \cdot 300 \cdot 10^{-9}} \approx \boxed{303 \text{ kHz}}$$

(Note in some data sheets, they use the formula of 100khz as f_CPU aka the normal time or standard mode. In problem, we assumed for maximum speed its to best to use fast mode.)

Lab10 Saleae Waveforms



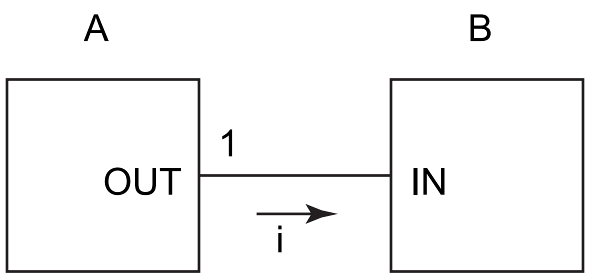



+0.5 ms
+0.6 ms
+0.7 ms
+0.8 ms
Read [0x62]
0x02 + ACK
0x91 + ACK
0x50 + ACK
0x69 + ACK
0xC3 + ACK
0x03 + ACK
0x6A + ACK
0xE1 + ACK
0xCA + NAK

Interface Checklist (revised 10/09/17)
ESE280 Ken Short

Naafiul Hossain-Group 16

A=AVR128DB48
B=SDC41



$V_{OHmin(A)} > V_{IHmin(B)}$

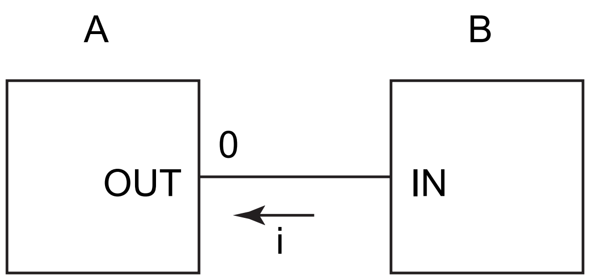
3.29V > 2.145V

$I_{OHmax(A)} > I_{IHmax(B)}$

115.5uA > 1nA

VOH min (LA) > VIH min (CB)
→ 3.29 V [3.3 − 10k × 1 μA]
→ 2.145 V [0.65 × VDD

IOH min (A) > IIH max (B)
→ (3.3 − 2.145) / 10k
→ Assume 1 μA



$V_{OLmax(A)} < V_{ILmax(B)}$

0.99v < 0.99v

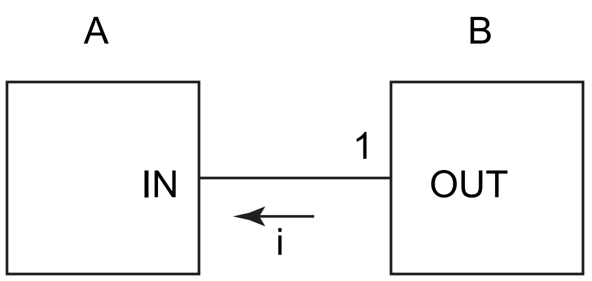
$I_{OLmax(A)} > I_{ILmax(B)}$

6mA > 1uA+0.33mA

VOL max (LA) < VIL max (CB)
→ 0.99 V
→ 0.99 V

IOL max (A) > IIL max (B)
→ 6 mA
→ 1 mA + (3.3 / 10k)

[VOL max (I2C) = 0.33 × VDD @ 6mA]]



$V_{IHmin(A)} < V_{OHmin(B)}$

2.31V < 3.29v

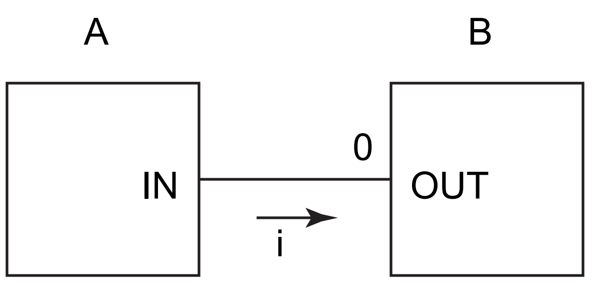
$I_{IHmax(A)} < I_{OHmax(B)}$

1uA < 99uA

Calculation:

VIH(min)(A) = 2.31 V

VOH(min)(B) = 3.29 V
2.31 V < 3.29 V → Satisfies logic high voltage level requirement



$V_{ILmax(A)} > V_{OLmax(B)}$

0.99V > 0.66V

$I_{ILmax(A)} < I_{OLmax(B)}$

0.33mA+5nA < 3mA

IIL(max)(A) = 3 mA + 5 nA ≈ 3.000005 mA

IOL(max)(B) = 3 mA

lab10_task3_display_sensor_data_file

AUTHOR
Version 1.1

Table of Contents

File Index

File List

Here is a list of all files with brief descriptions:

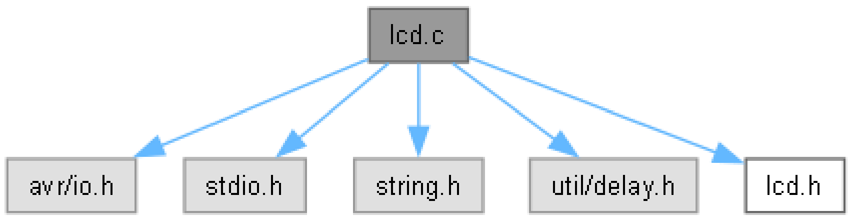
File Documentation

lcd.c File Reference

LCD display management for the temperature measurement system.

```
#include <avr/io.h>
#include <stdio.h>
#include <string.h>
#include <util/delay.h>
#include "lcd.h"
```

Include dependency graph for lcd.c:



Functions

- void **init_spi0_SerLCD** (void)
  Initializes the SPI interface for LCD communication.
- void **write_spi0_SerLCD** (unsigned char data)
  Sends a byte of data to the LCD via SPI.
- void **select_SS** (void)
  Selects the LCD as the SPI slave device.
- void **deselect_SS** (void)
  Deselects the LCD as the SPI slave device.
- void **update_SerLCD** (void)
  Updates the content displayed on the LCD.
- void **clear_display_buffs** (void)
  Clears the display buffers.

Variables

- char **dsp_buff1** [21]
- char **dsp_buff2** [21]
- char **dsp_buff3** [21]
- char **dsp_buff4** [21]

Detailed Description

LCD display management for the temperature measurement system.

This file contains all the functions necessary for initializing and managing the LCD display via SPI communication, including sending data to the display, clearing display buffers, and updating the display with new information.

Author

Naafiul Hossain

Date

2025-04-02

Function Documentation

void clear_display_buffs (void)

Clears the display buffers.

Fills the display buffer arrays with spaces to clear previous content, and sets the end character to null to properly terminate the string for display.

```
clear_display_buffs(); // Clear the display buffers before writing new content
```

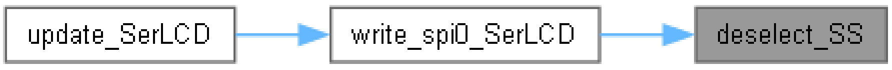
void deselect_SS (void)

Deselects the LCD as the SPI slave device.

Deactivates the slave select (SS) line specific to the LCD to end SPI communication.

```
deselect_SS(); // Deactivate the LCD SS line after sending data
```

Here is the caller graph for this function:



void init_spi0_SerLCD (void)

Initializes the SPI interface for LCD communication.

Sets up the SPI0 hardware module for communication with the LCD using master mode. Configures the direction of SPI pins and initializes SPI control registers.

```
init_spi0_SerLCD();
```

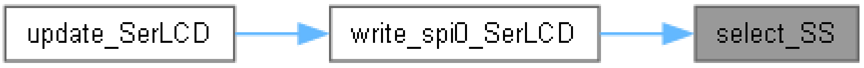
void select_SS (void)

Selects the LCD as the SPI slave device.

Activates the slave select (SS) line specific to the LCD to initiate SPI communication.

```
select_SS(); // Activate the LCD SS line before sending data
```

Here is the caller graph for this function:



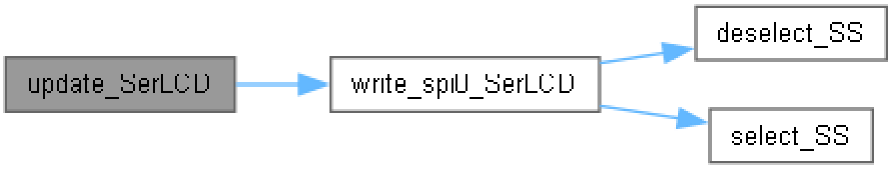
void update_SerLCD (void)

Updates the content displayed on the LCD.

Sends the contents of display buffers to the LCD via SPI, updating each line of the display. This function should be called whenever the display data needs to be refreshed.

```
clear_display_buffs();   // Clear the display buffers
sprintf(dsp_buff1, "Temperature: %dC", temp); // Prepare line 1
sprintf(dsp_buff2, "Status: %s", status);     // Prepare line 2
update_SerLCD();         // Send the updated buffer to the LCD
```

Here is the call graph for this function:



void write_spi0_SerLCD (unsigned char data)

Sends a byte of data to the LCD via SPI.

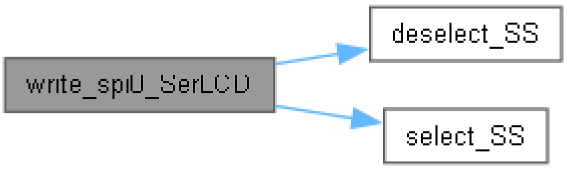
This function transmits a single byte to the LCD using SPI communication, ensuring the slave select line is appropriately managed before and after the transmission.

Parameters

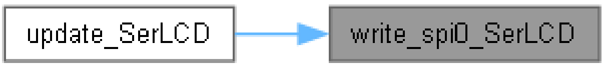
data	The data byte to be sent to the LCD.

```
write_spi0_SerLCD('H'); // Send character 'H' to the LCD
```

Here is the call graph for this function:



Here is the caller graph for this function:



Variable Documentation

char dsp_buff1[21]

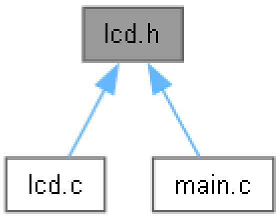
char dsp_buff2[21]

char dsp_buff3[21]

char dsp_buff4[21]

lcd.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void **init_spi0_SerLCD** (void)
  Initializes the SPI interface for LCD communication.
- void **write_spi0_SerLCD** (unsigned char data)
  Sends a byte of data to the LCD via SPI.
- void **update_SerLCD** (void)
  Updates the content displayed on the LCD.
- void **clear_display_buffs** (void)
  Clears the display buffers.
- void **select_SS** (void)
  Selects the LCD as the SPI slave device.
- void **deselect_SS** (void)
  Deselects the LCD as the SPI slave device.

Variables

- char **dsp_buff1** [21]
- char **dsp_buff2** [21]
- char **dsp_buff3** [21]
- char **dsp_buff4** [21]

Function Documentation

void clear_display_buffs (void)

Clears the display buffers.

Fills the display buffer arrays with spaces to clear previous content, and sets the end character to null to properly terminate the string for display.

```
clear_display_buffs(); // Clear the display buffers before writing new content
```

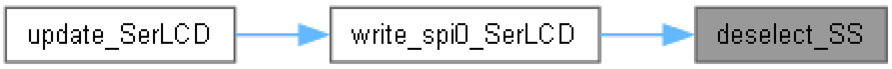
void deselect_SS (void)

Deselects the LCD as the SPI slave device.

Deactivates the slave select (SS) line specific to the LCD to end SPI communication.

```
deselect_SS(); // Deactivate the LCD SS line after sending data
```

Here is the caller graph for this function:



void init_spi0_SerLCD (void)

Initializes the SPI interface for LCD communication.

Sets up the SPI0 hardware module for communication with the LCD using master mode. Configures the direction of SPI pins and initializes SPI control registers.

```
init_spi0_SerLCD();
```

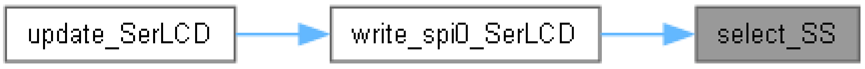
void select_SS (void)

Selects the LCD as the SPI slave device.

Activates the slave select (SS) line specific to the LCD to initiate SPI communication.

```
select_SS(); // Activate the LCD SS line before sending data
```

Here is the caller graph for this function:



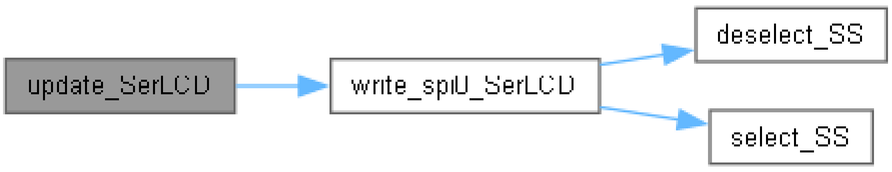
void update_SerLCD (void)

Updates the content displayed on the LCD.

Sends the contents of display buffers to the LCD via SPI, updating each line of the display. This function should be called whenever the display data needs to be refreshed.

```
clear_display_buffs();   // Clear the display buffers
sprintf(dsp_buff1, "Temperature: %dC", temp); // Prepare line 1
sprintf(dsp_buff2, "Status: %s", status);     // Prepare line 2
update_SerLCD();         // Send the updated buffer to the LCD
```

Here is the call graph for this function:



void write_spi0_SerLCD (unsigned char data)

Sends a byte of data to the LCD via SPI.

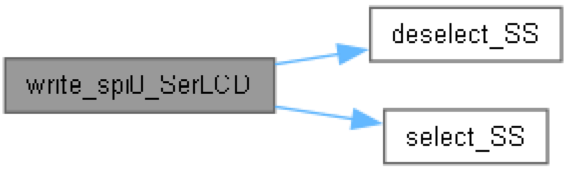
This function transmits a single byte to the LCD using SPI communication, ensuring the slave select line is appropriately managed before and after the transmission.

Parameters

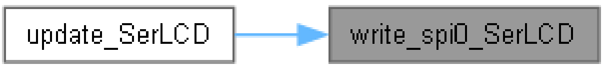
data	The data byte to be sent to the LCD.

```
write_spi0_SerLCD('H'); // Send character 'H' to the LCD
```

Here is the call graph for this function:



Here is the caller graph for this function:



Variable Documentation

char dsp_buff1[21] `[extern]`

char dsp_buff2[21] `[extern]`

char dsp_buff3[21] `[extern]`

char dsp_buff4[21] `[extern]`

lcd.h

Go to the documentation of this file.

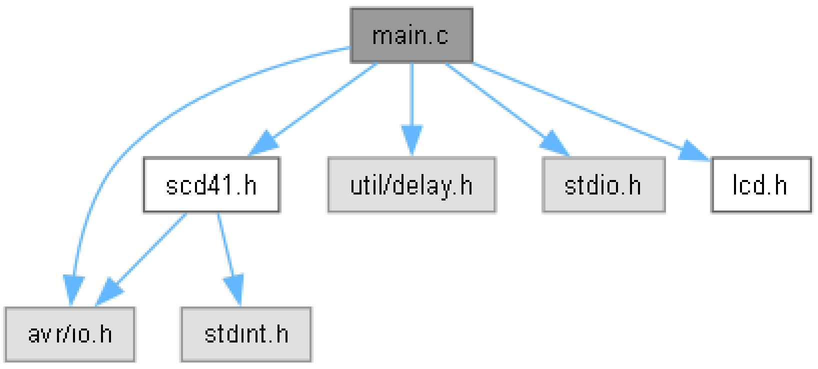
```c
/*
 * lcd.h
 *
 * Created: 4/2/2025 1:20:53 AM
 *  Author: Naafiul Hossain
 */
#ifndef LCD_H
#define LCD_H

 extern char dsp_buff1[21]; // Buffer for LCD display line 1 - Global variable, static storage, no linkage
 extern char dsp_buff2[21]; // Buffer for LCD display line 2 - Global variable, static storage, no linkage
 extern char dsp_buff3[21]; // Buffer for LCD display line 3 - Global variable, static storage, no linkage
 extern char dsp_buff4[21]; // Buffer for LCD display line 4 - Global variable, static storage, no linkage


void init_spi0_SerLCD(void);
void write_spi0_SerLCD(unsigned char data);
void update_SerLCD(void);
void clear_display_buffs(void);
void select_SS(void);  // Add this line
void deselect_SS(void);  // Add this line

#endif // LCD_H
```

main.c File Reference

Task 3 – Display CO2, temperature, and humidity on SPI-based LCD using SCD41 sensor.

```
#include "scd41.h"
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>
#include "lcd.h"
```

Include dependency graph for main.c:



Macros

- #define **F_CPU**   4000000UL

Functions

- int **main** (void)

Detailed Description

Task 3 – Display CO2, temperature, and humidity on SPI-based LCD using SCD41 sensor.

Author: Naafiul Hossain

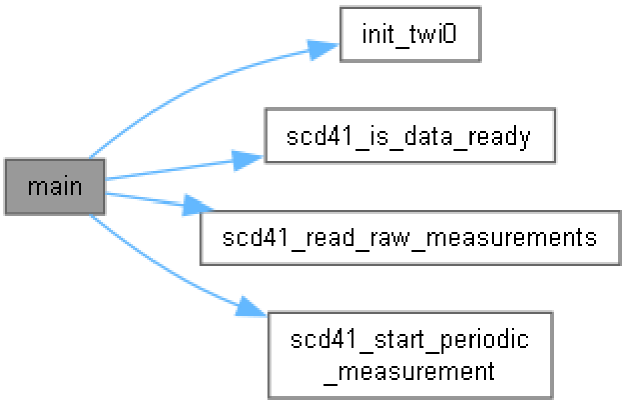
Macro Definition Documentation

#define F_CPU   4000000UL

Function Documentation

int main (void)

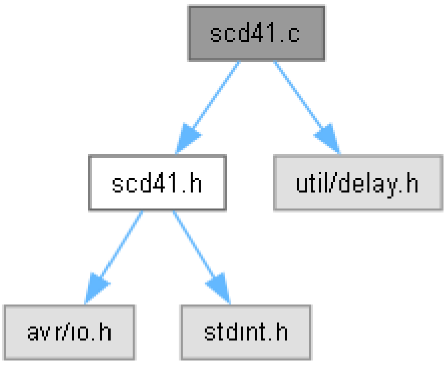
Here is the call graph for this function:


main
init_twi0
scd41_is_data_ready
scd41_read_raw_measurements
scd41_start_periodic_measurement

scd41.c File Reference

```
#include "scd41.h"
#include <util/delay.h>
```

Include dependency graph for scd41.c:



Macros

- #define **SCD41_I2C_ADDR** 0x62
- #define **CMD_START_MEAS** 0x21B1
- #define **CMD_DATA_READY** 0xE4B8
- #define **CMD_READ_MEAS** 0xEC05

Functions

- void **init_twi0** (void)
- void **scd41_start_periodic_measurement** (void)
- uint8_t **scd41_is_data_ready** (void)
- void **scd41_read_raw_measurements** (uint16_t *co2, uint16_t *temp, uint16_t *hum)

Macro Definition Documentation

#define CMD_DATA_READY   0xE4B8

#define CMD_READ_MEAS   0xEC05

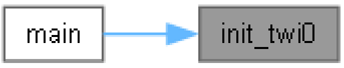
#define CMD_START_MEAS   0x21B1

#define SCD41_I2C_ADDR   0x62

Function Documentation

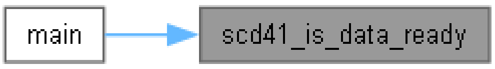
void init_twi0 (void)

Here is the caller graph for this function:



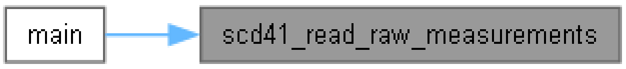
uint8_t scd41_is_data_ready (void)

Here is the caller graph for this function:



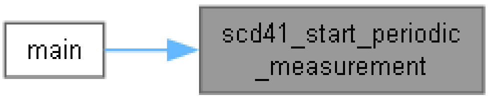
void scd41_read_raw_measurements (uint16_t * co2, uint16_t * temp, uint16_t * hum)

Here is the caller graph for this function:



void scd41_start_periodic_measurement (void)

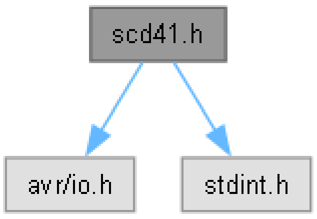
Here is the caller graph for this function:



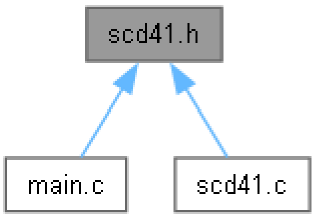
scd41.h File Reference

```
#include <avr/io.h>
#include <stdint.h>
```

Include dependency graph for scd41.h:



This graph shows which files directly or indirectly include this file:



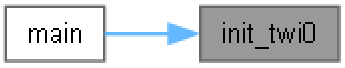
Functions

- void **init_twi0** (void)
- void **scd41_start_periodic_measurement** (void)
- uint8_t **scd41_is_data_ready** (void)
- void **scd41_read_raw_measurements** (uint16_t *co2, uint16_t *temp, uint16_t *hum)

Function Documentation

void init_twi0 (void)

Here is the caller graph for this function:



uint8_t scd41_is_data_ready (void)

Here is the caller graph for this function:

main
scd41_is_data_ready

void scd41_read_raw_measurements (uint16_t * co2, uint16_t * temp, uint16_t * hum)

Here is the caller graph for this function:

main
scd41_read_raw_measurements

void scd41_start_periodic_measurement (void)

Here is the caller graph for this function:

main
scd41_start_periodic
_measurement

scd41.h

Go to the documentation of this file.

```c
/*
 * scd41.h
 *
 * Interface for communicating with the SCD41 CO?, temperature, and humidity sensor via TWI0.
 * Used in Lab 10 Task 3 for a modular multifile implementation.
 *
 * Author: Naafiul Hossain
 */

#ifndef SCD41_H
#define SCD41_H

#include <avr/io.h>
#include <stdint.h>

// === Public API ===
void init_twi0(void);
void scd41_start_periodic_measurement(void);
uint8_t scd41_is_data_ready(void);
void scd41_read_raw_measurements(uint16_t *co2, uint16_t *temp, uint16_t *hum);

#endif // SCD41_H
```

Index

INDEX

Signature Confirmation Inbox ×

bryant.gonzaga@stonybrook.edu
to me

4/25/2025 16:33:13,victor.morales.1@stonybrook.edu,Muhammad Sharjeel,115185427,Sig. 1: LT1 - Run read_data,100,

Signature Confirmation Inbox ×

bryant.gonzaga@stonybrook.edu
to me

4/25/2025 16:58:14,dylan.wong@stonybrook.edu,Muhammad Sharjeel,115185427,Sig. 2: LT2 - Run multi-file program,100,