SmartFusion cSoC: Interfacing with OLED using I2C

Table of Contents

Introduction

The SmartFusion® customizable system-on-chip (cSoC) device contains a hard embedded microcontroller subsystem (MSS), programmable analog circuitry, and FPGA fabric consisting of logic tiles, static random access memory (SRAM), and phase-locked loops (PLLs). The MSS consists of a 100 MHz ARM® Cortex™-M3 processor, advanced high-performance bus (AHB) matrix, system registers, Ethernet MAC, DMA engine, real-time clock (RTC), embedded nonvolatile memory (eNVM), embedded SRAM (eSRAM), fabric interface controller (FIC), the Philips Inter-Integrated Circuit (I2C), serial peripheral interface (SPI), and external memory controller (EMC).

The MSS has two identical I^2C peripherals that perform serial-to-parallel conversion on data originating from serial devices, and perform parallel-to-serial conversion on data from the ARM Cortex-M3 processor to these devices. The ARM Cortex-M3 processor controls the I^2C peripherals through the advanced peripheral bus (APB) interface.

The I^2C peripherals in the SmartFusion cSoC device support I^2C, SMBus, and PMBus data transfers that conform to the I^2C v2.1 specification and support the SMBus v2.0 and PMBus v1.1 specifications. The I^2C peripherals can operate as either a master or a slave. When operating in the Master mode, the I^2C peripherals generate the serial clock and data to the slave device that needs to be accessed. The I^2C peripherals can generate serial clock from 104 KHz to 1.66 MHz by dividing MSS clock, which can be controlled by software. The I^2C peripherals use a 7-bit addressing format and run up to 400 Kbps (Fast mode) data rates nominally. Faster rates can be achieved depending on the external load. Refer to the *SmartFusion Microcontroller Subsystem User's Guide* for more details on I^2C peripherals.

This application note describes how to use the I^2C interface on the SmartFusion cSoC device and to display the text on RiTdisplay 96x16 OLED module. This OLED module consists of 96x16 OLED panel and OLED driver with controller (SSD0300, Solomon Systech Limited). Refer to the *RiTdisplay 96x16 OLED Module datasheet* for more details. A basic understanding of SmartFusion design flow is assumed.

Refer to the *Using UART with a SmartFusion cSoC - Libero SoC and SoftConsole Flow Tutorial* to understand the SmartFusion design flow.

Design Example Overview

This design example demonstrates using I^2C OLED module on the SmartFusion Development Kit Board and the SmartFusion Evaluation Kit Board. These kits have one OLED module — RiTdisplay PMO13701 — that is connected to I2C_0 on both kits.

Figure 1 shows the top level interface signals used in this design example. The UART in the MSS acts as a user interface for display selection through HyperTerminal.

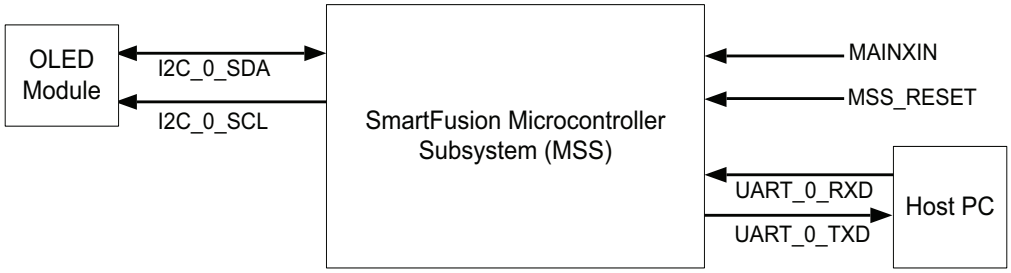

***Figure 1* • Top Level Interface Signals**

Description of the Design Example

The MSS is configured to use one I^2C interface (I2C_0) and one UART interface (UART_0). I2C_0 and UART_0 are clocked by PCLK0 on APB bus 0. PCLK0 is derived from the clock conditioning circuit (CCC) in the MSS that generates 80 MHz clock.

The I^2C peripheral in the MSS is configured as master that operates in the Fast mode. The APB bus clock (PCLK0) divider is set to 256 to generate 312.5 KHz serial clocks (I2C_0_SCL).

The I^2C Master device initiates a write transaction by sending a START bit as soon as the bus becomes free. The START bit is followed by the 7-bit serial address of the slave device and write bit. The slave acknowledges receipt of its address with an acknowledge bit. This design example uses 96x16 OLED module as slave device with serial address set to 0x78.

The design example uses UART as a user interface for OLED display selection through HyperTerminal. You can select the following options through HyperTerminal to display on OLED panel.

1. Plain text display
2. Vertical scrolling
3. Horizontal scrolling
4. Continuous horizontal and vertical scrolling
5. Sine wave
6. Clear the OLED data

Verilog Libero projects and SoftConsole project are provided in the design files attached with this design example.

Interface Description

Table 1 illustrates the top level interface signal descriptions.

Table 1 • **Interface Description**

Signal	Direction	Description
MSS_RESET_N	Input	Active Low reset signal for the MSS.
MAINXIN	Input	Main crystal oscillator circuit. Input to the crystal oscillator circuit. Pin for connecting an external crystal, ceramic resonator, or RC network.
UART_0_TXD	Output	UART Transmit data
UART_0_RXD	Input	UART Receive data
I2C_0_SDA	Input	Serial data
I2C_0_SCL	Output	Serial clock

Software Implementation

The software design (main.c) mainly consists of two sections:

1. UART Initialization and Configuration: This section initializes and configures the UART for user interface and prints the following OLED display selections on HyperTerminal:
   – Plain text display
   – Vertical scrolling
   – Horizontal scrolling
   – Continuous horizontal and vertical scrolling
   – Sine wave
   – Clear the OLED data
2. OLED driver Application Program Interface (APIs): This section executes different OLED APIs based on the user display selection input.

The OLED driver APIs described in Table 2 are used in the example design to display text on I^2C OLED module.

Table 2 • **API Description**

Function	Description
OLED_init ()	This function initializes and configures the I^2C peripheral for data transfer. It configures the I^2C controller with serial clock speed and OLED Slave address. It also initializes the OLED module and sets the cursor on first character in the first line.
OLED_write_data ()	This function writes the data to OLED based on the parameters passed to this function. The data parameter is a pointer to a data structure that holds different fields of data to be required for OLED.
OLED_vertical_scroll ()	This function enables vertical scrolling by one line.
OLED_horizontal_scroll ()	This function enables horizontal scrolling.
put_pixel ()	This function displays one pixel on the OLED.
OLED_clear_display ()	This function clears the content of the display.

Refer to "Appendix B - I^2C OLED Driver's APIs" on page 6 for more details on OLED initialization and configuration, OLED data structure, vertical scroll, horizontal scroll, and clear display.

Refer to the *SmartFusion MSS I2C Driver User's Guide* for more details on I^2C driver's API.

Running the Design

Board Settings

The design example works on the SmartFusion Development Kit Board and the SmartFusion Evaluation Kit Board with default board settings. Refer to the following user's guides for default board settings:

- *SmartFusion Development Kit User's Guide*
- *SmartFusion Evaluation Kit User's Guide*

Program the Design and Running the Application

Program the SmartFusion Development Kit Board with the generated/provided *.stp file (refer to "Appendix A - Design and Programming Files" on page 6) using FlashPro and then power cycle the board.

Invoke the SoftConsole IDE, by clicking on the **Write Application code** under **Develop Firmware** in Libero® System-on-Chip (SoC) (refer to "Appendix A - Design and Programming Files" on page 6) and launch the debugger. Start HyperTerminal with 57600 baud rate, 8 data bits, 1 stop bit, no parity, and no flow control. If your computer does not have HyperTerminal program, use any free serial terminal emulation program like PuTTY or Tera Term. Refer to the *Configuring Serial Terminal Emulation Programs* tutorial for configuring the HyperTerminal, Tera Term, and PuTTY.

When you run the debugger in SoftConsole, the HyperTerminal window prompts you to enter the option to display the text on OLED. Once you enter the option, the application displays the text on OLED (as either fixed text, vertical scroll, horizontal scroll, or sine wave). Figure 2 shows the screen shot of HyperTerminal with I^2C OLED display option.

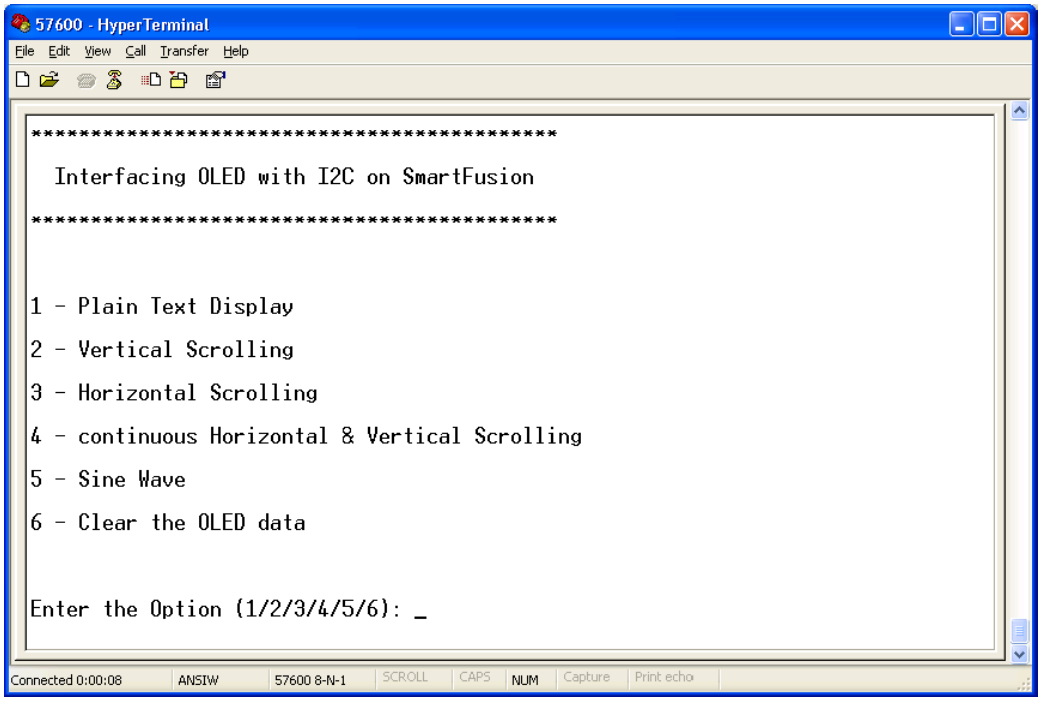

***Figure 2* • Screen Shot of HyperTerminal With I^2C OLED Display**

Release Mode

The release mode programming file (STAPL) is also provided. Refer to the Readme.txt file included in the programming zip file for more information.

Refer to the *Building Executable Image in Release Mode and Loading into eNVM Tutorial* for more information on building an application in release mode.

Conclusion

This application note highlights the features of the SmartFusion cSoC devices along with detailed features of the I^2C peripheral. This application note describes how to use the I^2C interface on the SmartFusion cSoC devices to display the text on RiT display 96x16 OLED module with sample software. It also explains the usage of I^2C OLED driver APIs.

Appendix A - Design and Programming Files

You can download the design files from the Microsemi SoC Products Group website:

www.microsemi.com/soc/download/rsc/?f=A2F_AC347_DF.

The design file consists of Libero SoC, Verilog, SoftConsole software project, and programming files (*.stp) for A2F500-DEV-KIT and A2F-EVAL-KIT. Refer to the Readme.txt file included in the design file for the directory structure and description.

You can download the programming files (*.stp) in release mode from the Microsemi SoC Products Group website: www.microsemi.com/soc/download/rsc/?f=A2F_AC347_PF.

The programming zip file consists of STAPL programming file (*.stp) for A2F500-DEV-KIT and A2F-EVAL-KIT, and a Readme.txt file.

Appendix B - I2C OLED Driver's APIs

This section describes the software driver APIs used in this design to carry out transactions with I2C OLED module. These drivers are included in the design files with this design example.

Function Description Of Data Structures

Enum: oled_no_of_line

This enum represents the number of lines to be written on OLED.

- FIRST LINE: The OLED cursor is set to line number 1 and only 1 line is printed on OLED.
- SECOND_LINE: The OLED cursor is set to line number 2 and only 1 line is printed on OLED.
- FIRST_TWO_LINES: The OLED cursor is set to line number 1 and line 1 and line 2 are printed on OLED.

```
typedef enum {
    FIRST_LINE = 0,
    SECOND_LINE,
    THIRD_LINE,
    FOURTH_LINE,
    FIFTH_LINE,
    FIRST_TWO_LINES,
    ALL_LINES
} ;
```

Structure: oled_data

This is the OLED data structure that holds different fields of data to be required for OLED module. A pointer to an instance of the oled_data structure is passed as the parameter OLED driver functions.

```
Struct oled_data
{
    /* Represents line number 1, where String 1 has to be printed*/
    uint8_t line1;

    /*Represents character offset within line 1, from where String 1 has to be printed*/
    uint8_t char_offset1;

    /* Represents line number 2, where String 2 has to be printed*/
    uint8_t line2;

   /* Represents character offset within line 2, from where String 2 has to be printed*/
    uint8_t char_offset2;

    /* String 1 holds the data to be displayed on line 1 of OLED. It has to be less than
49 characters*/
    char *string1;

    /* String 2 holds the data to be displayed on line 2 of OLED. It has to be less than
49 characters*/
```

```
    char *string2;

    /* Represents line number 3, where String 3 has to be stored in GDRAM*/
    uint8_t line3;

    /* Represents character offset within the line 3, from where String 3 has to be
stored in GDRAM*/
    uint8_t char_offset3;

    /* String 3 holds the data to be stored on line 3 of GDRAM. It has to be less than
49 characters*/
    char *string3;

    /* Represents line number 4, where String 4 has to be stored in GDRAM*/
    uint8_t line4;

    /* Represents character offset within the line 4, from where String 4 has to be
stored in GDRAM*/
    uint8_t char_offset4;


    /* String 4 holds the data to be stored on line 4 of GDRAM. It has to be less than
49 characters*/
    char *string4;

    /* Represents line number 5, where String 5 has to be stored in GDRAM*/
    uint8_t line5;

    /* Represents character offset within the line 5, from where String 5 has to be
stored in GDRAM */
    uint8_t char_offset5;

    /* String 5 holds the data to be stored on line 5 of GDRAM. It has to be less than
49 characters*/
    char *string5;

    /* Holds the contrast value to be set for String 1 and String 2*/
    uint8_t contrast_val;

    /* Represents ON or OFF for horizontal scrolling*/
    uint8_t hor_scr_on_off;


    /* Represents number of coumns scrolls per step for horizontal scroll*/
    unsigned char column_scrool_per_step;

    /* Represents start page for horizontal scroll*/
    unsigned char start_page;

    /* Represents time interval for horizontal scroll*/
    unsigned char time_intrval_btw_scroll_step;

    /* Represents end page for horizontal scroll*/
    unsigned char end_page;

    /* Represents ON or OFF for vertical scrolling*/
     uint8_t ver_scr_on_off;

   /* Display Offset*/
     uint8_t ver_off_set;
};
```

Function Description of API

void OLED_init(void);

This function initializes and configures the I^2C peripheral for data transfer. It configures the I^2C controller with serial clock speed and OLED Slave address. It also initializes the OLED module and sets the cursor on the first character in the first line.

void OLED_clear_display(oled_no_of_line LINES);

This function clears the content of the display RAM-based on the LINES input.

@param oled_no_of_line: This parameter enum holds the number of lines.

- If FIRST_LINE is passed as parameter to this function, this function clears only first line.
- If SECOND_LINE is passed as parameter to this function, this function clears only second line.
- If FIRST_TWO_LINES is passed as parameter to this function, this function clears entire OLED display.

*void OLED_write_data(struct oled_data * data, oled_no_of_line flag);*

This function writes the data to OLED based on the parameters passed to this function.

@param data: The data parameter is a pointer to an oled_data structure that holds different fields of data to be required for OLED (see the oled_data structure definition).

@param oled_no_of_line: This parameter enum holds the number of lines.

- If FIRST_LINE is passed as parameter to this function, this function writes string 1 at first line.
- If SECOND_LINE is passed as parameter to this function, this functions writes string 1 at second line.
- If FIRST_TWO_LINES is passed as parameter to this function, this function writes string 1 and string 2 at first line and second line respectively.

*void OLED_horizontal_scroll(struct oled_data * horiz_scroll);*

This function enables horizontal scrolling.

@param data: This parameter is a pointer to an oled_data structure that holds different fields of data to be required for OLED (see the oled_data structure definition).

*void OLED_vertical_scroll(struct oled_data * ver_scroll);*

This function enables vertical scrolling by one step.

@param data: This parameter is a pointer to an oled_data structure that holds different fields of data required for OLED (see the oled_data structure definition).

void OLED_contrast(uint8_t color_contrast);

This function sets the contrast to the data displayed on the OLED.

@param color_contrast: This parameter that holds contrast value. The color_contrast values should be in the range of 1 to 256.

void put_pixel(float value);

This function displays one pixel on the OLED.

@param value: The value parameter that holds pixel value. This value should be in the range of 0 to 3.

List of Changes

The following table lists critical changes that were made in each revision of the document.

Revision*	Changes	Page
Revision 5 (January 2013)	Added "Board Settings" section and modified "Running the Design" section (SAR 43469).	4
Revision 4 (November 2012)	Modified the section "Description of the Design Example" (SAR 42901).	2
	Modified the section "Software Implementation" (SAR 42901).	3
	The Figure 2 is updated (SAR 42901).	4
Revision 3 (February 2012)	Removed ".zip" extension in the Design and Programming files link (SAR 36763).	6
Revision 2 (January 2012)	Modified the section "Introduction" (SAR 35787).	1
	Modified the section "Description of the Design Example" (SAR 35787).	2
	Modified the section "Running the Design" (SAR 35787).	4
	Added the section "Release Mode" (SAR 35787).	4
	Modified the section "Appendix A - Design and Programming Files" (SAR 35787).	6
Revision 1 (August 2010)	Modified the section "Running the Design" (SAR 27473).	4
	Modified the section "Appendix A - Design and Programming Files" (SAR 27473).	6
	Removed Table 3 • Design Files Description (SAR 27473).	

*Note: *The revision number is located in the part number after the hyphen. The part number is displayed at the bottom of the last page of the document. The digits following the slash indicate the month and year of publication.*

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo CA 92656 USA
Within the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

Microsemi Corporation (NASDAQ: MSCC) offers a comprehensive portfolio of semiconductor solutions for: aerospace, defense and security; enterprise and communications; and industrial and alternative energy markets. Products include high-performance, high-reliability analog and RF devices, mixed signal and RF integrated circuits, customizable SoCs, FPGAs, and complete subsystems. Microsemi is headquartered in Aliso Viejo, Calif. Learn more at **www.microsemi.com**.

51900215-5/01.13