Displaying POT Values over UART

SoftConsole Standalone Flow Tutorial for SmartFusion cSoC



Table of Contents

Introduction	3
Tutorial Requirements	3
Objective	4
Working with the SoftConsole	5
Step 1 - Launching SoftConsole	5
Step 2 - Configuring MSS Peripherals	5
Step 3 - Programming SmartFusion Board Using FlashPro	15
Step 4 - Building the Project	17
Step 5 - Configuring the Serial Terminal Emulation Program	21
Step 6 - Installing Drivers for the USB to RS232 Bridge	23
Step 7 - Running the Application	23
Step 8 - Debugging the Application	27
Step 9 - Building an Executable Image in Release mode	29
Appendix A – Libero SoC Vault/Repository Settings	31
Appendix B – Firmware Catalog Settings	33
Appendix D – Thimware Catalog Settings	
List of Changes	
Product Support	37
Customer Service	
Customer Technical Support Center	37
Technical Support	37
Website	
Contacting the Customer Technical Support Center	
ITAR Technical Support	38



Introduction

This tutorial demonstrates how to develop an application that can be implemented using SmartFusion[®] customizable system-on-chip (cSoC) device. After completing this tutorial, you will be familiar with the following:

- 1. Creating and implementing a project using SoftConsole and SmartFusion cSoC device
- 2. Configuring the peripherals using SmartDesign
- 3. Configuring the analog compute engine (ACE)
- 4. Generating the programming file to program the SmartFusion cSoC device
- 5. Compiling application code
- 6. Creating and launching a debug session

Tutorial Requirements

Software Requirements

This tutorial requires the following software installed on your PC:

- Libero[®] system-on-chip (SoC) v10.0 (or later) that can be downloaded from the link below: www.microsemi.com/soc/download/software/libero/files.aspx.
- Microsemi SoftConsole v3.3 or later, which is installed as a part of Libero SoC v10.0 installation or can be downloaded from www.microsemi.com/soc/download/software/softconsole/default.aspx
- FlashPro v10.0 or higher, which is often installed as part of the Microsemi Libero SoC installation and can be launched from within Libero SoC or as a standalone.

Hardware Requirements

This tutorial requires the following hardware:

- SmartFusion Evaluation Kit Board or SmartFusion Development Kit Board.
- Two USB cables (programming and communication) one for connecting the programmer to your PC and the other to connect the universal asynchronous receiver/transmitter (UART) interface on the board to PC.

Associated Project Files

You can download the associated project files for this tutorial from the Microsemi website: www.microsemi.com/soc/download/rsc/?f=SmartFusion_SoftConsole_POTlevel_UART_tutorial_DF.

You can download the programming file (*.stp) in release for this tutorial from the Microsemi website: www.microsemi.com/soc/download/rsc/?f=SmartFusion_SoftConsole_POTlevel_UART_tutorial_PF.

MSS Components Used

- ARM[®] Cortex[™]-M3 processor
- Communications matrix
- Clock conditioning circuitry (CCC)
- UART_0
- ACE



Displaying POT Values over UART - SoftConsole Standalone Flow Tutorial for SmartFusion cSoC

Target Board

SmartFusion Evaluation Kit Board (A2F-EVAL-KIT) or SmartFusion Development Kit Board (A2F-DEV-KIT).

Objective

The objective of this tutorial is to explain how to configure the SmartFusion cSoC MSS peripherals using the SmartFusion cSoC MSS configurator outside the Libero SoC V10.0 flow.

Design Steps

Following are the major steps to be executed for this tutorial:

- Configure the SmartFusion cSoC MSS peripherals using SmartDesign MSS
- Generate the programming file and program the SmartFusion cSoC device
- Create a simple project and run the application using the SoftConsole debugger

As a part of MSS configuration, this tutorial demonstrates how to configure SmartFusion analog channels and ACE (used to monitor the voltage across the potentiometer).

The UART_0 is used to send the analog to digital converter (ADC) results to a serial terminal program (HyperTerminal).

The hardware configuration has four flags:

- Over 1.0 V
- Over 1.5 V
- Over 2.0 V
- Over 2.5 V

The design monitors the voltage across a potentiometer (POT) and the four flags are included for voltage monitoring.



Working with the SoftConsole

Step 1 - Launching SoftConsole

- Open SoftConsole by clicking Start > Programs > Microsemi SoftConsole v3.3 > Microsemi SoftConsole IDE.
- The SoftConsole Workspace Launcher is displayed as shown in Figure 1 · . Click Browse on the SoftConsole Workspace Launcher to navigate to your preferred location (where you will create your project), or enter the location of your workspace. For example: C:\Microsemiprj\POT_Uart). Click OK.
 - Note: You can also specify the workspace as a default workspace for all your projects by selecting Use this as the default and do not ask again.

You can switch workspaces within SoftConsole by selecting File > Switch Workspace.

SC Workspace Launcher	×
Select a workspace	
Microsemi SoftConsole IDE v3.3 stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: C:\Microsemiprj\POT_Uart Browse	
Use this as the default and do not ask again	
OK Cancel	

Figure 1 · Specifying the SoftConsole Workspace

Step 2 - Configuring MSS Peripherals

- 1. Open the External Tools dialog box using Run > External Tools > External Tools Configurations.
- 2. Double-click Program to create a new configuration.
- 3. Enter MSS_Configurator in the Name field.
- 4. Click Browse File System in the Location field. The Open dialog box is displayed.
- 5. Navigate to the Libero.exe.
- 6. In the Arguments field, enter the following:
 - STARTED_BY:SoftConsole
 - PROJECT_LOCATION:C:\Microsemiprj\POT_Uart\Voltage_Monitor
 - DESIGN_NAME:Voltage_Monitor
 - Click Apply and then click Run.



SC External Tools Configurations	Example a second se
Create, manage, and run confi Run a program	gurations Or
Image: Second Secon	Name: New_configuration Main Refresh Build Environment Common Location: C./Microsemi/Libero_v10.0\Designer/bin\libero.exe Browse Workspace Browse File System Variables Working Directory: Browse Workspace Browse File System Variables Arguments: STARTED BY:SoftConsole A
	PROJECT_LOCATION:C:\Microsemiprj\POT_Uart\Voltage_Monitor DESIGN_NAME:Voltage_Monitor Variableg Note: Enclose an argument containing spaces using double-quotes (").
Filter matched 4 of 4 items	Apply Reyert
?	Run Close

Figure 2 · Running SmartFusion MSS Configurator

- 7. The Libero SoC appears with a **New Project** dialog box, as shown in Figure 3 · below. Select the following:
 - Die: If you are using SmartFusion Evaluation Kit Board, enter A2F200M3F; if you are using SmartFusion Development Kit Board, enter A2F500M3F.
 - Package: 484 FBGA
 - Speed: STD
 - Die Voltage: 1.5



Name:	Voltage_Monitor			
ocation:	C:\Microsemipri\POT_Uart\\oli	ana Marikar		Browse
		age_monicor		browse
Prefered HDL type:	 Verilog O VHDL 			
Description:				
Edit Tool Profiles				
Device				
Family:	SmartFusion	~		
Die:	A2F200M3F	*		
Package:	484 FBGA	~		
Speed:	STD 💌			
Die Voltage:	1.5 💌			
Operating Conditions:	сом 💌			
Temper	ature (in degrees Celsius)	Best	Typical 25	Worst 85
VCCA V	oltage (in volts) 5 Voltage (in volts)	1.575	1.5	1.425
VCCI 1.	8 Voltage (in volts)	1.9	1.8	1.7 2.3
	5 Voltage (in volts) 3 Voltage (in volts)	2.7 3.6	3.3	3
esign Template				
🗸 Use template				
	Core			Versio
SmartFusion Microcontrol	er Subsystem (MSS)			2.5.10
			v s	now only latest versio

The Project Name and Project Directory parameters are already specified.

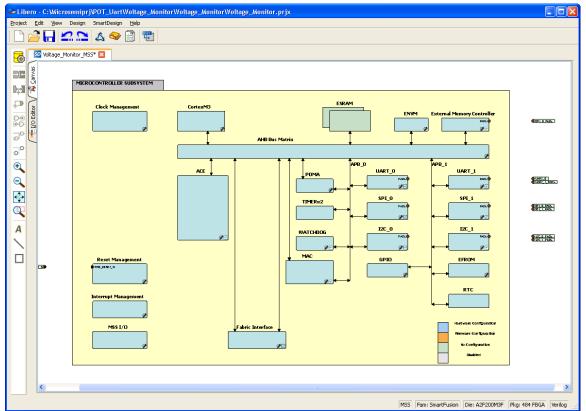
Figure 3 · New Project Dialog Window

Note: If you do not see the latest MSS version (v2.5.106 or later), you need to change your repositories settings. Steps for setting up repositories are described in Appendix A – Libero SoC Vault/Repository Settings.

If your vault does not have MSS core then, double click on MSS name in the Design Template to download the core.



Displaying POT Values over UART - SoftConsole Standalone Flow Tutorial for SmartFusion cSoC



8. Click OK. The MSS canvas is displayed, as shown in Figure 4 · .

Figure 4 · MSS in the SmartDesign Canvas

The enabled MSS peripherals are highlighted in blue, and can be configured in hardware. The disabled peripherals are shown in gray.

To disable a peripheral that is not required, select the peripheral, right-click, and clear the **Enabled** check box in the lower right corner of the **peripheral** box. The box turns grey to indicate that the peripheral has been disabled. Disabled peripherals can be enabled by repeating the procedure.

An enabled peripheral looks as shown in Figure 5 \cdot .

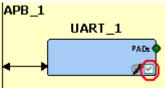
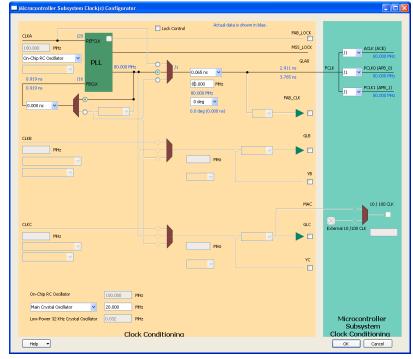


Figure 5 · Enabling the Peripheral

For this design we will be using the analog compute engine (ACE), UART_0, and clock management peripherals. Other peripherals must be disabled.

- 9. Double-click the Clock Management block and configure as shown below:
 - CLKA: On-chip RC Oscillator
 - MSS Clock source: PLL output
 - MSS clock frequency: 80 MHz
 - 10/100CLK: Cleared check box (by default it will be cleared, if MAC is disabled)
 - · Use default settings for all other fields.





10. After completing the configuration, click OK.



11. Click **Project > Save** to save SmartFustion_UART_MSS.

Configuring ACE

To configure ACE, double-click the ACE peripheral block and configure as follows:

 Connect TM0 to the potentiometer (POT) on the SmartFusion Evaluation Kit Board or the SmartFusion Development Kit Board. Configure a voltage monitor to measure the voltage across the POT and also create flags to indicate when the voltage is greater than 1.0 V, 1.5 V, 2.0 V, and 2.5 V. These flags will be used to monitor the voltage.

Configure ACE					
ADC Configuration	· · ·	ngs			
ACLK: 80 MHz	ADC Clock: 10 MHz	Resolution: 12 vits		Advanced Options	
Add service: Active Bipola Prescaler Inpu ADC Direct Input Differential Input Temperature Input Hanalog Comparator Sigma Delta DAC	ADC Block 0 ADC Block 1	Signal	Sample F	iackage Pin SCB	21
Help				OK Cancel	

Figure 7 · MSS ACE Configuration

Displaying POT Values over UART - SoftConsole Standalone Flow Tutorial for SmartFusion cSoC

- 2. Select **ADC Direct Input** and click **Add** (or double-click ADC Direct Input) and enter the parameters as shown in Table 1
 - Signal name: TM0_Voltage
 - Send raw results to DMA: un-checked
 - Acquisition time: 10 µs
 - Filtering factor: None

Table 1 · Configure ADC Direct Input Dialog Box

Flag Name	Flag Type	Threshold (V)	Hysteresis (mV)
over_1p0v	OVER	1	1
over_1p5v	OVER	1.5	1
over_2p0v	OVER	2	1
over_2p5v	OVER	2.5	1

Configure ADC Di	rect Input							×
AT	Signal na	me:				b		
	jrmo_voli	age		📕 Send raw A	ADC result to D	AMA		
Acquisition tim	ne: 10.	000 us						
Digital filterin	ig		_	🗌 Linear tran	sformation			
Filtering fac	tor: Non	e 🔻		Scale factor	:			
Initial value	: 0.0	00 V		Offset:		_		
🗖 Send filt	ered result to	DMA		🔲 Send tra	nsformed resu	ilt to DMA		
		Thre	eshold Detectio	n		*	\times	
F	lag Name	Flag Type	Threshold (V)	Hysteresis (mV)	Assert Samples	De-asse Sample		
	_1p0v	OVER	1	1				
	_1p5v	OVER OVER	1.5	1			_	
	_2p0v _2p5v	OVER	2.5	1			-	
					ОК	C	ancel]

Figure 8 · MSS ADC Direct Input Configuration

- 3. Click OK.
- 4. Assign ADC Direct Input Signal to package pin W8 in the Configure ACE dialog box.



0		1 2	0	
igure ACE				
	Controller F	lags		
ADC Configuration				
ACLK: 80 MHz	ADC Clock: 10 MHz	Resolution: 12 v bits	Advanced Op	tions
Add service:				_/ ×
Active Bipolar Prescaler Inpu ADC Direct Input	Service	Signal	Sample Package Pin	SCB
Current Input Differential Input	ADC Direct Input	TM0_Voltage	10.000 W8(TM0)	0(4)
Temperature Input LVTTL Input				
Analog Comparator Sigma Delta DAC				
Add >>				
	ADC Block 0 ADC Block 1			
Help			ОК	Cancel

5. The **Configure ACE** tab is displayed as shown in Figure 9 · .



The next step in configuring the ACE is to enable the sampling sequence. This configuration dialog is launched by clicking the **Controller** tab (next to the **Configure ACE** tab).

Configure ACE
Configure ACE Controller Flags
Procedures
Name
ADC0_MAIN ADC1_MAIN ADC1_MAIN ADC1_MAIN
) Operating sequence entry: C Auto C Manual
Details of procedure: ADC0_MAIN
Available signals: Sampling rate
Signal Actual Rate (ksps)
Total same Insert operating sequence slot
Operating sequence
Help OK Cancel

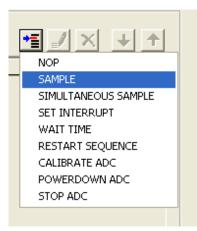
6. Select Manual as the Operating sequence entry in the Controller tab.

- Figure 10 · Insert Operating Sequence Slot
- 7. Click the **Insert operating sequence slot**, highlighted in Figure $10 \cdot .$

🏷 Microsemi.

Displaying POT Values over UART - SoftConsole Standalone Flow Tutorial for SmartFusion cSoC

8. Select SAMPLE.





9. The Configure SAMPLE window is displayed. Select **TM0_Voltage** from the drop-down list and click **OK**.

Configure 'SAMPLE'
Analog Pad: TM0_Voltage
Description
Sample Analog Peripheral. Only the peripherals that are assigned to this ADC can be selected.
Help OK Cancel

Figure 12 · Configure SAMPLE

- 10. Click Insert operating sequence slot again and select RESTART SEQUENCE.
- 11. Click Calculate Actual Rate. The final Controller tab window is displayed as shown in Figure 13 · .

Configure ACE	×
Configure ACE Controller Flags	
Procedures	
Name ADC Block 0	
ADC Block 0 ADC Block 1 ADC Block 1	
Operating sequence entry: C Auto C Manual	
Details of procedure: ADC0_MAIN Available signals: Sampling rate	
Signal Actual Rate (ksps)	
->>> TM0_Voltage 84.200	
Calculate Actual Rate Iotal sampling rate: 84.200 ksps	
Operating sequence	
Sample TM0_Voltage	
Restarts the execution sequence for this timeslot	
Help OK Cancel	

Figure 13 · MSS ACE Configuration: Final Controller Tab

12. Click the Flags tab in the Configure ACE window. This tab lists the Flags set from PPE registers.



13. Click the + sign to expand the Flag Registers group. The PPE_FLAGSn registers contain the userdefined flags.

Select PPE_FLAGS0 (FLAGBANK0). Observe that PPE_FLAGS0 contains the 4 threshold flags assigned earlier. These are the flags that were defined when the direct input voltage service was configured. The flag register can be read by the Cortex-M3 processor. The flags also generate interrupts to the Cortex-M3 processor.

Configure ACE Controller	Flags				
elect a register to view ACE flag mapping:	Available bits of PPE_F	FLAG50 register:			
- Fabric - Flag Registers	Bit	Source	ACE	Cortex-M3 Interrupt	^
PPE_FLAG50 (0x40021450)	0 TM0_Voltage	e:over_1p0v	54	118	
	1 TM0_Voltage		55	119	
	2 TM0_Voltage	e:over_2p0v	56	120	
	3 TM0_Voltage	e:over_2p5v	57	121	
PPE_SFFLAGS (0×40021460)	4 <none></none>		58	122	
	5 <none></none>		59	123	
	6 <none></none>		60	124	
	7 <none></none>		61	125	
	8 <none></none>		62	126	
	9 <none></none>		63	127	
	10 <none></none>		64	128	
	11 <none></none>		65	129	
	12 <none></none>		66	130	
	13 <none></none>		67	131	
	14 <none></none>		68 69	132	
	15 <none></none>		69 70	133	
	16 <none></none>		70	134 135	
			72		
	18 <none></none>		72	136	
	19 <none></none>		73	137 138	
	20 <none></none>		74		
	21 <none></none>		75	139 140	
	22 (none)		76	140	
	23 <none> 24 <none></none></none>		77	141	
	25 <none></none>		79 80	143	
	26 <none> 27 <none></none></none>			144	
			81 82	145	
	28 <none></none>		82	146 147	
	29 <none></none>		83	147	~
	,				

Figure 14 · ACE Flag Mapping: PPE Flag Registers

14. Click **OK** to close the ACE configuration box.

Configuring the General Purpose Input/Output (GPIO) Peripheral

Note: If you are not using SmartFusion Evaluation Kit Board – REV E, skip the configuration of GPIO. Double-click the GPIO block in the MSS component, configure as shown in Figure $15 \cdot$, and click **OK**.

55	Config	uring MSS_GF	PIO_0 (MSS_GPIO	- 1.0.101)				
	Config	uration						<u> </u>
		-Multiplexed With	12C1 Peripheral Dedic	ated I/Os				
		GPIO_31:	Use as MSS I/O Pad	Output 😽	• U20	or connect to Fabric	Not Used 🔍	
		GPIO_30:	Use as MSS I/O Pad	Output 💊	v22	or connect to Fabric	Not Used 🔍	
		-Multiplexed With	n UART1 Peripheral Dec	licated I/Os				
		GPIO_29:	Use as MSS I/O Pad	Output 💌	W22	or connect to Fabric	Not Used 🛛 👻	
		GPIO_28:	Use as MSS I/O Pad	Output 💌	V20	or connect to Fabric	Not Used 🕑	
		-Multiplexed With	n SPI1 Peripheral Dedica	ated I/Os				
		GPIO_27:	Use as MSS I/O Pad	Not Used 🛛 👻	W21	or connect to Fabric	Not Used 🔽	
		GPIO 26:	Lise as MSS I/O Pad	Not Used 🔍	AA22	or connect to Fabric	Not Lised 🔍	✓
	Help	•					ОК	Cancel



This example requires GPIO_31, GPIO_30, GPIO_29, and GPIO_26 to be connected to LED_8 to LED_5 on the SmartFusion Evaluation Kit Board (A2F_EVAL-KIT Rev E).



Displaying POT Values over UART - SoftConsole Standalone Flow Tutorial for SmartFusion cSoC

Generating the MSS Component

1. The MSS canvas is displayed, as shown in Figure 16 · .

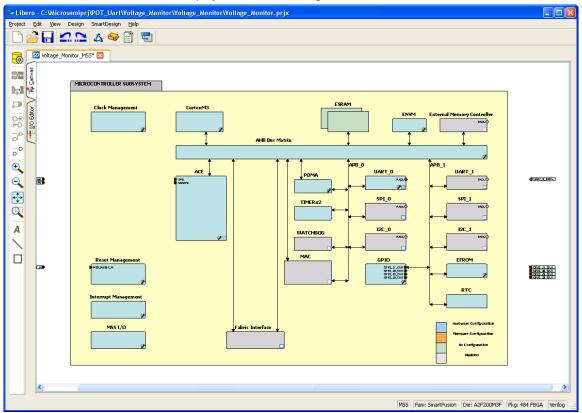


Figure 16 · The MSS Canvas

- 2. Save your design using Project > Save Voltage_Monitor_MSS.
- 3. Click Design > Configure Firmware as shown in Figure 17 · .



Note: Check whether or not you are able to see the latest version of the drivers without any warning or error indicating that firmware is missing from the Vault. If missing, refer to Appendix B – Firmware Catalog Settings to set your Repositories. If you are opening the SmartDesign for the first time then you need to download the all firmware drivers from the **Firmware** tab in the SmartDesign



4. On the **DESIGN_FIRMWARE** tab, clear the **Generate** check boxes for all the peripherals for which you do not need to generate the firmware. Click **Configuration** on the SmartFusion_CMSIS_PAL_0 instance and select **SoftConsole** as the configuration.

So Voltage_Monitor 🗵 So Voltage_Monitor_MSS 🛛 So DESIGN_FIRMWARE* 🔀								
	Generate		In	stance Name	Core Type	Version		Compatible Hardware Instance
1	~	Ø,	HAL_0		HAL	2.1.102		Voltage_Monitor_MSS
2	~		MSS_ACE	_Driver_0	MSS_ACE_Driver	2.2.101	~	Voltage_Monitor_MSS:MSS_ACE_
3	~		MSS_GPI	O_Driver_0	MSS_GPIO_Driver	2.0.105	~	Voltage_Monitor_MSS:MSS_GPIO
4			MSS_IAP	_Driver_0	MSS_IAP_Driver	2.2.101	~	Voltage_Monitor_MSS
5			MSS_NVM	1_Driver_0	MSS_NVM_Driver	2.2.102	~	Voltage_Monitor_MSS:MSS_ENVM
6			MSS_PD	🕵 Configuring	SmartFusion_CMSIS_P/			Voltage_Monitor_MSS:MSS_DMA_
7			MSS_RT		onard anon_onoio_r			Voltage_Monitor_MSS:MSS_RTC_
8			MSS_Tim	Configuration —			~	Voltage_Monitor_MSS:MSS_TIMEI
9	 Image: A start of the start of		MSS_UA	Software T	ool Chain SoftConsole	~	~	Voltage_Monitor_MSS:MSS_UART
10		Ť	SmartFu	Help 🔻	ОК	Cancel	~	Voltage_Monitor_MSS

Figure 18 · Configuring SmartFusion_CMSIS_PAL_0

- 5. Click OK.
- 6. Click **Project > Save to save the Design_Firmware**.
- 7. Save the design and generate the MSS component by clicking Generate Component or clicking SmartDesign > Generate Component.



Figure 19 · Select SoftConsole in the Configuration Window

8. Confirm that the design was successfully generated.

Note: If errors are indicated, open the log window (View > Log Window) to get additional information.

9. Open the memory map for the design (**Design > Reports**). Scroll the window to become familiar with the locations of the peripherals. Close the window when finished.

Step 3 - Programming SmartFusion Board Using FlashPro

Jumper Settings for SmartFusion Evaluation Kit Board

Before you proceed with programming the device, ensure that LCPS or FlashPro4 is properly connected to the board. Use the following details to ensure the correct jumper settings. Refer to the *SmartFusion Evaluation Kit User's Guide* and the *SmartFusion Development Kit User's Guide* for additional information.

- JP10: Connect pin 1 and 2.
- JP7: Connect pin 1 and 2 for LCPS programming mode.
- J6: Connect pins 1 and 2 with the jumper.
- JP6: Connect pins 2 and 3 with the jumper.
- J13: Connect USB cable to the J13 connector. When the cable is connected, the FlashPro4 or FlashPro drivers might be installed if they are not already installed.

• J14: Connect one end of USB mini B cable to J14.

Jumper Settings for SmartFusion Development Kit Board

SW9 must be off (JTAGSEL = H) in order to program the SmartFusion device. SW9 remains in the off
position for Libero SoC and SoftConsole programming. Make the jumper settings as shown in the
following table:

Factory Default	Factory Default	Factory Default
JP1: 1–2	JP12: 1–2	JP21: 1–2
JP2: 1–2	JP13: 1–2	JP22: 2–3
JP4: 1–3; 7–9	JP14: 1–2	JP23: 1–2
JP5: 1–2; 3–4	JP15: 1–2	JP24: 1–2
JP6: 2–3	JP16: 2–3	JP27: 1–2
J7: 2–3; 6–7; 10–11; 14–15	JP17: 2–3	JP28: 1-2
JP7: 1–2	JP18: 1–2	J32: 1–2; 3–4; 5–6
JP8: 3-4; 7-8; 11-12; 15-16	JP19: 2–3	_
JP11: 1-2	JP20: 1–2	_

<u> </u>			
Table 2 · Jumper	Settings for	r Development	Kit Board

Programming the Device

1. Click **Program Device** as shown in Figure $20 \cdot$ or click **Design > Program Device**.



Figure 20 · Select SoftConsole in the Configuration Window

Note: If errors are indicated, open the Reports (**Design > Reports**) to get additional information.

2. Close Libero (File > Exit).



Step 4 - Building the Project

1. Invoke the SoftConsole project generated by Libero SoC v10.0 using **File > Switch Workspace > Other...** and browse to the SoftConsole workspace as shown in Figure 21 · .

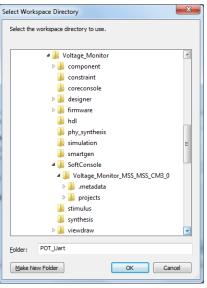


Figure 21 · Specifying the Workspace

2. The SoftConsole window looks as shown in Figure 22 · on page17.

SC C/C++ - Welcome to Microsemi SoftCor	sole IDE v3.3 - Microsemi SoftConsole IDE	v3.3				
<u>Eile E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> rch	<u>Project Run W</u> indow <u>H</u> elp					
📬 • 📰 🗁 🖬 爹 ム 🎯 • 🗐 🗊 ½ × २ × ७ ७ ० ०	88 • 🖻 • 69 • 🕴 🔦 • 📎 • 🗄 🏇 •	0 • 0	i 🥭 🖋 •	🖹 💀 c/c++)	
Project Explorer 🛛 🗖 🗖	🗎 Welcome 🛛		- 8	🗄 Outlin 🛛 💿 M	1ake 🕛 🗖	
□ \$ ~	Microsemi SoftConsole IDE v3.	.3		An outline is not available	e.	
@ 🥵 Voltage_Monitor_MSS_MSS_CM3_0_app ₪ 😰 Voltage_Monitor_MSS_MSS_CM3_0_hw_pl						
	🖹 Problems 🛛 🧔 Tasks 📃 Console 🔲	Properties			~ - 0	
	0 items					
	Description 🔺	Resource	Path	Locat Type		
<						
			:			
з П.						

Figure 22 · SoftConsole Window

Displaying POT Values over UART - SoftConsole Standalone Flow Tutorial for SmartFusion cSoC

3. Enter the code provided below in main.c file in the Voltage_Monitor_MSS_MSS_CM3_0_app project.

Sample C code

```
#include "mss uart.h"
#include "mss_ace.h"
#include "mss gpio.h"
#include <stdio.h>
#define Microsemi logo \
"\n\r \
**
       ** ******
                     ***** ****
                                      ****
                                              ****
                                                              **
                                                                     * *
                                                                        ****** \n\r \
      * *
              *
                             *
                                  *
                                           *
                                            *
                                                      *
                                                              * *
                                                                    * *
                                                                            *
                                                                                 \n\r \
   * *
       *
              *
                    *
                             * * * * *
                                     *
                                          *
                                               ****
                                                      *****
                                                              *
                                                                 * * *
                                                                            *
                                                                                 \n\r \
                                                  *
                                                      +
                                                                                 \n\r \
                                              ****
          ******
                    *****
                                      * * * *
                                                                      * ******
                             *
                                  *
                                                      *****
                                                              *
int main()
{
       const uint8 t greeting[] = "\n\rWelcome to Microsemi's SmartFusion Voltage
Monitor\n\r";
    const uint8_t * channel_name;
/*Initialize and Configure GPIO*/
    MSS GPIO init();
    MSS GPIO config( MSS GPIO 31 , MSS GPIO OUTPUT MODE );
    MSS_GPIO_config( MSS_GPIO_30 , MSS_GPIO_OUTPUT_MODE );
    MSS GPIO config( MSS GPIO 29 , MSS GPIO OUTPUT MODE );
    MSS GPIO config( MSS GPIO 28 , MSS GPIO OUTPUT MODE );
    /*Initialize UART 0*/
    MSS UART init(
        &g_mss_uart0,
        MSS UART 57600 BAUD,
        MSS UART DATA 8 BITS | MSS UART NO PARITY | MSS UART ONE STOP BIT );
    /*Initialize ACE*/
    ACE init( );
    MSS UART polled tx string( &g mss uart0, (const uint8 t*)Microsemi logo );
    MSS UART polled tx( &g mss uart0, greeting, sizeof(greeting) );
    channel_name = ACE_get_channel_name( TM0_Voltage );
    for (;;)
    {
       uint8_t display_buffer[32];
        uint16 t adc result;
```



```
int32 t adc value mv;
       adc result = ACE get ppe sample( TMO Voltage );
       adc value mv = ACE convert to mV( TMO Voltage, adc result );
       if ( adc value mv < 0 )
               {
                       snprintf( (char *) display buffer, sizeof(display buffer),
               "%s : -%.3fV\r\b", channel name, ((float)(-adc value mv) /
(float)(1000)));
       else
            {
                       snprintf( (char *) display buffer, sizeof(display buffer),
               "%s : %.3fV\r\b", channel name, ((float)(adc value mv) / (float)(1000)));
            }
MSS UART polled tx string( &g mss uart0, display buffer );
       /* checking the status of Voltage flags */
       int32_t flag_status_2p5v = ACE_get_flag_status(TM0_Voltage_over_2p5v);
               int32 t flag status 2p0v = ACE get flag status(TMO Voltage over 2p0v);
               int32_t flag_status_1p5v = ACE get flag status(TMO Voltage over 1p5v);
               int32 t flag status 1p0v = ACE get flag status(TMO Voltage over 1p0v);
               /* Voltage flags are displayed on the LEDs through GPIO */
               uint32 t gpio output;
               if ( flag status 2p5v == FLAG ASSERTED )
                       gpio output = ~(
                                      MSS GPIO 28 MASK |
                      MSS GPIO 29 MASK |
                      MSS GPIO 30 MASK |
                      MSS GPIO 31 MASK );
               else
               if ( flag status 2p0v == FLAG ASSERTED )
                       gpio output = ~(
                                      MSS GPIO 28 MASK |
                      MSS GPIO 29 MASK |
                      MSS GPIO 30 MASK );
               else
               if ( flag_status_1p5v == FLAG_ASSERTED )
                       gpio output = ~(
                                      MSS GPIO 28 MASK |
                      MSS GPIO 29 MASK );
               else
               if ( flag status 1p0v == FLAG ASSERTED )
                       gpio output = ~(
```

Wicrosemi. Displaying POT Values over UART - SoftConsole Standalone Flow Tutorial for SmartFusion cSoC

```
MSS GPIO 28 MASK );
                                 else
                                             gpio output = (
                                                                    MSS GPIO 28 MASK |
                                             MSS GPIO 29 MASK |
                                             MSS GPIO 30 MASK |
                                             MSS_GPIO_31_MASK );
                            MSS GPIO set outputs ( gpio output );
                   }
                return 0;
          }
          4. The SoftConsole window looks similar to Figure 23 · on page 20.
SC C/C++ - Voltage_Monitor_MSS_MSS_CM3_0_app/main.c - Microsemi SoftConsole IDE v3.3
                                                                                                                                 <u>File Edit Source Refactor Navigate Search Project Run Window Help</u>
   1 • 📃 👜 | 🗟 | 爹 | 🛆 | 👩 • 🚳 • 👩 • 🎯 •
                                                           i 🔦 • 🕲 • i 🏇 • 🔾 • 🗛 • i 🕭 🛷 • i 🌛 🗉 🔳
                                                                                                                     🔛 🔤 C/C++
 🐓 🗸 🖓 🗸 🏷 🔶 🖓
🎦 Project Explorer 🛛 📄 🔄 🔽 🗆 🔂 🔝 main.c 😒
                                                                                                          🗝 🗖 🗄 Outlin 🛛 💿 Make
                                                                                                                                     - -
 🖃 😂 Voltage_Monitor_MSS_MSS_CM3_0_app
                                       26#define Microsemi_logo \
                                                                                                           ^
                                                                                                                       📲 😿 🖋 🛛 💥
   ■ foldge_nen
■ foldge_nen
■ foldge_nen
■ foldge_nen
■ foldge_nen
■ foldge_nen
                                       27"\n\r_\
                                                                                                                       .\drivers\mss_uart\mss_ua
                                                                                                                    -
                                                                                                     ****1
                                       28**
                                                                           ****
                                                                                     ****
                                                                                             ****
                                                                                                                    .\drivers\mss_ace\mss_ace
                                       29* *
 E 😂 Voltage_Monitor_MSS_MSS_CM3_0_hw_pla
                                                                                                                    ų,
                                                                                                                       .\drivers\mss_gpio\mss_gp
   🗄 前 Includes
                                             * *
                                                                ÷
                                                                          *****
                                                                                                     *****
                                       30*
                                                 *
                                                                                             ****
                                                                                                                    stdio.h
   🗉 👝 CMSIS
                                       31*
                                                  *
                                                         ....
                                              *
                                                                                                     .....
                                                                                                                    0
                                                                                                                       main() : int
                                                     ******
                                                                 *****
   🗄 🧀 drivers
                                       32*
                                                                          *
                                                                                     ****
                                                                                            ****
                                                                                                     ****
                                                                                                                    # Microsemi_logo
                                       33
   😟 🧀 drivers_config
    🗄 👝 hal
                                       34
   🖮 脑 Voltage_Monitor_hw_platform.h
                                       35
                                              const uint8_t greeting[] = "\n\rWelcome to Microsemi's
                                       36
                                              const uint8_t * channel_name;
                                       37
                                       38
                                              /*Initialize and Configure GPIO*/
                                       39
                                              MSS_GPIO_init();
                                              HSS_GPIO_IDI();
HSS_GPIO_config( MSS_GPIO_27 , MSS_GPIO_OUTPUT_MODE );
HSS_GPIO_config( MSS_GPIO_26 , MSS_GPIO_OUTPUT_MODE );
HSS_GPIO_config( MSS_GPIO_25 , MSS_GPIO_OUTPUT_MODE );
HSS_GPIO_config( MSS_GPIO_24 , MSS_GPIO_OUTPUT_MODE );
                                       40
                                       41
                                       42
                                                                                                        ~
                                       43
                                          <
                                                                                                                                    >
                                                                                                                                   - - -
                                    🖹 Problems 🕱 🛛 🖉 Tasks 📮 Console 🔲 Properties
                                    0 items
                                                                        Resource Path
                                                                                                  Locat... Type
                                      Description 🔺
                                                                   Writable
                                                                                           54:76
                                                                                Smart Insert
```

Figure 23 · SoftConsole workspace

5. Perform a clean build by selecting **Project** > **Clean**. Accept the default settings in the **Clean** dialog box and click **OK**.



SC Clean							
Clean will discard all build problems and built states. The next time a build occurs the projects will be rebuilt from scratch.							
○ Clean all projects							
Image: Source of the second projects geneticed below Imag							
Start a <u>b</u> uild immediately							
Build the entire workspace Devild only the selected projects							
OK Cancel							

Figure 24 · Settings for a Clean Build

- 6. Make sure there are no errors and warnings.
 - Note: If there are any compilation errors regarding the file or directory missing then use the SoftConsole directories including feature from Project options > C/C++ Build > settings > GNU
 C Compiler > Directories to include the header file directories. This enables the compiler to look in to added directories for the included header files.

Step 5 - Configuring the Serial Terminal Emulation Program

Prior to running the application program, you need to configure the terminal emulator program (HyperTerminal, included with Windows[®]) on your computer. Perform the following steps to use the SmartFusion Evaluation Kit Board or the SmartFusion Development Kit Board:

- Connect a second mini USB cable between the USB connector on the SmartFusion Evaluation Kit Board (or the SmartFusion Development Kit Board) and a USB port of your computer. If Windows prompts you to connect to Windows Update, select No, not at this time and click Next.
- If the SFE USB to RS232 Controller drivers are automatically detected (this can be verified in Device Manager), as shown in Figure 25 · , proceed to next step; otherwise follow the Step 6 - Installing Drivers for the USB to RS232 Bridge.

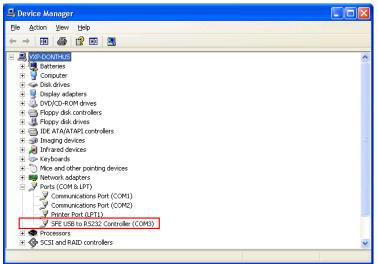


Figure 25 · Device Manager Listing SFE USB to RS232 Controller Drivers

- 3. From the Windows **Start** menu, select **Programs > Accessories > Communications > HyperTerminal**. This opens HyperTerminal. If your computer does not have HyperTerminal, 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.
- 4. Enter Hyperterminal in the Name field in the Connection Description dialog box and click OK.

Connection Description	<u>?</u> ×
New Connection	
Enter a name and choose an icon for the connection:	
<u>N</u> ame:	
Hyperterminal	
<u>l</u> con:	
	2
OK Car	ncel

Figure 26 · New Connection

5. Select the appropriate COM port (to which USB-Rs232 drivers are pointed) from the **Connect using** drop-down list and click **OK**.

Connect To	? 🔀							
Nyperterminal								
Enter details for the phone number that you want to dial:								
<u>C</u> ountry/region:	India (91) 💌							
Ar <u>e</u> a code:	91							
Phone number:								
Co <u>n</u> nect using:	СОМЗ							
	OK Cancel							

Figure 27 · Selecting the COM Port

- 6. Set the following in the COM Properties window and click OK:
 - Bits per second: 57600
 - Data bits: 8
 - Parity: None
 - Stop Bits: 1
 - Flow control: None



COM3 Properties		? ×
Port Settings		
		_
<u>B</u> its per second:	57600	
<u>D</u> ata bits:	8	
<u>P</u> arity:	None	
<u>S</u> top bits:	1	
<u>F</u> low control:	None 🗸	
	<u>R</u> estore Defau	lts
	K Cancel A	spply

Figure 28 · Setting the COM Properties

Click OK to close the Hyperterminal Properties dialog box.
 Next time you can directly open HyperTerminal (without configuring) by selecting, Programs > Accessories > Communications > HyperTerminal > Hyperterminal.

Step 6 - Installing Drivers for the USB to RS232 Bridge

Note: To install the USB-RS232 drivers, you should have administrative privileges for your PC.

Use the following steps to install drivers for the USB to RS232 Bridge:

- 1. Unzip USB_Drivers.zip file and browse to the USB Drivers folder.
- 2. Double-click (run) the Preinstaller.exe.
- 3. Accept the default installation location and click Install.
- 4. Click Continue Anyway if prompted.
- 5. When the installation is complete, click **OK**. The ports (COM & LPT) section of the Device Manager lists SFE USB to RS232 Controller under the Ports section of the Device Manager.

Step 7 - Running the Application

Follow the steps given below to debug the application project using SoftConsole:

- 1. Select **Open Debug Dialog** or **Debug Configurations** from the **Run** menu of the SoftConsole. The Debug dialog is displayed.
- 2. Double click on **Microsemi Cortex-M3 RAM target**. The **Debug Configurations** window is displayed, as shown in Figure 29 · .

SC Debug Configurations							
Create, manage, and run confi	gurations	Ť.					
Constant Carter Service Constant Carter Service Constant Carter Service Service Carter Service Carter Service Carter Serv	Name: Vokage_Monitor_MSS_MSS_CM3_0_opp Debug Main % Debugger Commands Project (optional): Vokage_Monitor_MSS_MSS_CM3_0.opp C/C++ Application: ebugit Vokage_Monitor_MSS_MSS_CM3_opp C/C++ Application console Search Project	Browse					
?	Debug	Close					

Figure 29 · Debug Configurations Window

- 3. Confirm that the following appear on the Main tab in the Debug window:
 - Name: Voltage_Monitor_MSS_MSS_CM3_app Debug
 - Project: Voltage_Monitor_MSS_MSS_CM3_app
 - C/C++ application: Debug\ Voltage_Monitor_MSS_MSS_CM3_app
- 4. Select the **Commands** tab. Confirm that commands appear in the Initialize and Run command sections, as shown in Figure $30 \cdot .$

SC Debug Configurations						
Create, manage, and run config	gurations	Ť.				
Vorte Market Back State See State	Name: Voltage_Monitor_MSS_MSS_CM3_0_app Debug Image: Man Signature Commands Initialization commands for programs linked using: # [1] debug-in-actenia-mark usion-esram.ld # [2] debug-in-actenia-mark usion-esram.ld # [1] Targeting SmartFusion embedded SRAM (esram) @ 0x20000000. # (11] Targeting SmartFusion embedded SRAM (esram) @ 0x20000000. # (11] Targeting SmartFusion embedded SRAM (esram) @ 0x20000000. # (11] Targeting SmartFusion embedded SRAM (esram) @ 0x20000000. # (11] Targeting SmartFusion embedded SRAM (esram) @ 0x20000000. # (11] Targeting SmartFusion embedded SRAM (esram) @ 0x20000000. # (11] Targeting SmartFusion embedded SRAM (esram) @ 0x20000000. # (11] Targeting SmartFusion embedded SRAM (esram) @ 0x20000000. # (11] Targeting SmartFusion embedded SRAM (esram) @ 0x20000000. # (11] Targeting SmartFusion embedded SRAM (esram) @ 0x20000000. # (11] Targeting SmartFusion embedded SRAM (esram) @ 0x20000000. # (11] Targeting SmartFusion embedded SRAM (esram) @ 0x20000000. # (11] Targeting SmartFusion embedded SRAM (esram) @ 0x20000000. # (11] Targeting SmartFusion embedded SRAM (esram) @ 0x20000000. # (11] Targeting SmartFusion embedded SRAM (esram) @ 0x20000000. # (11] Targeting SmartFusion embedded SRAM (esram) @ 0x20000000. # (11] Targeting SmartFusion embedded SRAM (esram) @ 0x20000000. # (11] Targeting SmartFusion embedded SRAM (esram) @ 0x200000000. # (11] Target	Apply Revert				
?		Debug Close				

Figure 30 · Debugger Commands

- 5. Click Apply and Debug.
- 6. Click Yes when prompted for Confirm Perspective Switch. This displays the debug view mode.





Figure 31 · Confirm Perspective Switch

7. The Debug Prospective window is displayed, as shown in Figure 32 · .

🕿 Debug - Voltage_Moniter_MSS_MSS_CM3_0_app/main.c - Microsemi SoftConsole IDE v3.3							
<u>Eile Edit S</u> ource Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp							
i 📫 • 🗟 🖆 📾 i 🏇 • 🔕 • 隆 🖋 • i 🌽 🕫 🐓 • 🖓 • 💝	<- + → -				😭 🐝 Debug	₽ C/C++	
🏇 Debug 🛛 🦳 🗖	🛛 🕪= Variable	s 🛛 💊 Bre	eakpoints 🐰 Reg	isters 🚬 M	odules	- 0)	
🎉 🖉 🔶 🗈 🔳 🔣 🔜 🖉 🙀 🎽	7			6	🏷 🍕 📄 🏟 💕	× ¾ ▽	
SC Voltage_Moniter_MS5_MS5_CM3_0_app Debug [Microsemi Cortex-M3 RAM Target]	Name					Value	
🚍 🔗 Embedded GDB (9/15/11 8:03 PM) (Suspended)	🗉 🥭 gre	eting				0x2000ffaC	
E Thread [1] (Suspended)	🗄 🌩 cha	nnel_name				0×0000000	
I main() c:\microsemipr)\pot_led_libero_softconsole\voltage_moniter\softcon							
C:\Program Files\Microsemi\SoftConsole v3.3\Sourcery-G++\bin\arm-none-eabi-gdb	.e						
	<					>	
						~	
						~	
						>	
🕒 Welcome 🔀 🖸 -1 <symbol available="" is="" not=""></symbol>				E Outline	22		
25					1ª, 😿 🖋	● ₩ ▽	
26 #define Microsemi_logo \					\drivers\mss uart\mss (
27"\n\r \					\drivers\mss_ace\mss_a		
28** ** ******* ****** *****	***** **	** *	***** 📃 🔤		\drivers\mss_dco\mss (
29** ** * * * * * * * * *	* *	* * *	*		tdio.h	,,	
30* * * * * * * **** * * ****	***** *	* * *	*	• m	nain() : int		
31* * * * * * * * * * *	* *	* *	*		1icrosemi logo		
32* * ****** ***** * * **** *****	***** *	* *	*****				
33							
34 35 const uint8 t greeting[] = "\n\rWelcome to Microse							
	emi's Smar	trusion Vo	Itage M				
36 const uint8 t * channel name;			>				
Console 🛛 🤕 Tasks 🖹 Problems 🕐 Executables 🟮 Memory			< 💥 🖹 🚮		🖻 🖉 🖻 🔹 🗎		
Voltage_Moniter_MS5_MS5_CM3_0_app Debug [Microsemi Cortex-M3 RAM Target] C:\Program Fi Loading section .data, size 0x5b8 lma 0x2000ab90	iles\Microsemi\S	oftConsole v3.3\	Sourcery-G++\bin\	arm-none-ea	bi-gdb.exe (9/15/11 8:0		
Start address 0x20000298, load size 45384						<u>^</u>	
Transfer rate: 31 KB/sec, 5673 bytes/write.						_	
set \$sp = *0x60080000						~	
						>	
: □ ◆ w	/ritable	Smart Insert	35:1				

Figure 32 · Debug Perspective

- Run the application by clicking Run > Resume or by clicking the Run icon on the SoftConsole toolbar. The voltage measurement along with the greeting message is displayed in the terminal program window.
- Turn the potentiometer (POT) on the SmartFusion Evaluation Kit Board or the SmartFusion Development Kit Board. The voltage measurement is displayed on HyperTerminal and the LEDs on the SmartFusion Evaluation Kit Board or the SmartFusion Development Kit Board will illuminate when one of the voltage monitor flags is asserted.
- 10. Adjust the POT and observe that the voltage measurement is continuously updated.

\sub Microsemi.

Displaying POT Values over UART - SoftConsole Standalone Flow Tutorial for SmartFusion cSoC

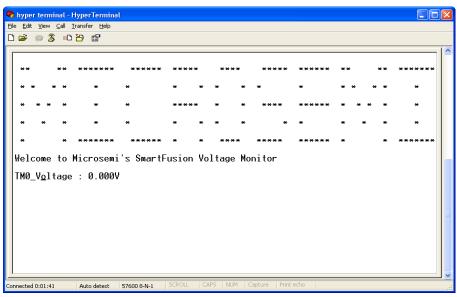


Figure 33 · Voltage Monitor

8 hyper terminal - HyperTerminal <u>File Edit Vi</u> ew <u>C</u> all <u>I</u> ransfer <u>H</u> elp	ļ						
□☞ ◎ Ѯ ▫ " " "							
** ** ******	*****	****	****	****	*****	** **	*****
** ** *	×	* *	* *	*	*	* * * *	*
* * * * *	*	*****	* *	****	*****	* * * *	*
* * * *	*	* *	* *	*	*	* * *	*
* * ******	*****	* *	****	****	*****	* *	******
Welcome to Microsemi	's SmartF	usion Vo	oltage Mo	nitor			
TM0_Voltage : 2.560V							
							=
Connected 0:02:11 Auto detect	57600 8-N-1	SCROLL C	APS NUM (apture Print	acha I		V

Figure 34 · Voltage Measurement Continuous Update

- 11. Observe the state of the LEDs as the POT is adjusted. Confirm that the flags work as specified in the ACE configurator.
- 12. Suspend the software application by clicking **Run > Suspend** from the SoftConsole menu.



Step 8 - Debugging the Application

Use the following steps to further debug the application:

1. Select the **Registers** tab on the upper right window pane to view the value of the Cortex-M3 processor internal registers.

🗱 Variables 🤷 Breakpoints 🛛	🚻 Registers 🕴 🛋 Modules
Name	Value
🖃 👬 Main	
8889 rO	0x2000b4f4
0000 r1	0x2000ff80
1010 r2	0x60
888 r3	0x2000b4f4
1010 r4	0x0
1818 r5	0xe0042020
8889 r6	0xc
8889 r7	0x2000ff58
888 r8	0xe004201c
888 r9	0x33f236b5
000 r10	0×1da60880
0101 r11	0×18001666
1010 40	

Figure 35 · The Registers Tab

2. Select the Variables tab in the upper left window pane to view the value of variables in the source code.

🎋 Debug 🗄 Outline 🖾 Variables 🖾 💊 Breakpoints 🔐 Registers 🛋 Mod	lules
Name	Value
(x)= rx_size	0
🖃 🥭 rx_buff	0x2000ffeb
(X)= rx_buff[0]	0
🖃 🥭 greeting	0x2000ffba
(X)= greeting[0]	'W'
(X)= greeting[1]	'e'
(X)= greeting[2]	Ψ
(X)= greeting[3]	'C'
	1
<	

Figure 36 · The Variables Tab

3. In the **Debug** window, select **Window > Show View > Disassembly** to display the assembly level instructions. The Assembly window opens on the right-side of the Debug perspective.

🏷 Microsemi.

Displaying POT Values over UART - SoftConsole Standalone Flow Tutorial for SmartFusion cSoC

SC Debug - Voltage_Moniter_MSS_MSS_CM3_0_app/main.c	
📸 - 🔚 📄 📸 - 🔕 - 🍳 - 🍅 🔗 - 🌛	
₩ Debug 🛛 🖓	🛋 Modules 🗣 Breakpoints 👫 Registers 🔎 Variables 励 Disassembly 🖾 🗸 🖓 🗖
C:\Program Files\Microsemi\SoftConsole v3.3\Sourcery-G++\bin	<pre>(</pre>
Imain.c III Imain.c III Imain.c IIII Imain.c IIIII Imain.c IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	<pre>0x200004d4 <main+36>: ldmia.w lr!, (r0, r1, r2, r3) 0x200004d8 <main+40>: stmia.w r12!, (r0, r1, r2, r3) 0x200004dc <main+40>: stmia.w r12!, (r0, r1) 0x200004ed <main+40>: stmia.w r12, (r0, r1) MSS_GPIO_init(); 0x200004e4 <main+52>: b1 0x20001880 <mss_gpio_init> MSS_GPIO_config(MSS_GPIO_27 , MSS_GPIO_0UTPUT_MODE); 0x200004e8 <main+56>: mov.w r1, #S 0x200004ec <main+60>: mov.w r1, #S 0x200004e0 <main+64>: b1 0x200018ec <mss_gpio_config> MSS_GPIO_config(MSS_GPIO_26 , MSS_GPIO_0UTPUT_MODE);</mss_gpio_config></main+64></main+60></main+56></mss_gpio_init></main+52></main+40></main+40></main+40></main+36></pre>
	0x200004f4 <main+68>: mov.w r0, #26</main+68>
Console Araks Problems Executables Immory Voltage_Moniter_MSS_MSS_CM3_0_app Debug [Microsem Cortex-M3 RAM Ta Transfer rate: 33 KB/sec, 5673 bytes/write. set \$sp = *0x60080000 set \$pc = *0x60080004 - 1 cont Cont O	rget] C:\Program Files\Microsemi\SoftConsole v3.3\Sourcery-G++\bin\arm-none-eabi-gdb.exe (9/15/11 8:11 PM)

Figure 37 · Assembly Window

- 4. You can single-step through the source code by choosing **Run > Step Into** or **Run > Step Over** or by clicking the **Step Into** → or **Step Over** → icons. Observe the changes in the source code window and Disassembly view. Performing a Step Over allows for stepping over functions. The entire function is executed but there is no need to single step through each instruction contained in the function.
- 5. Click the Instruction Stepping incompared in the perform Step Into operations. Observe that Step Into now executes a single line of assembly code.
- 6. Click the **Instruction Stepping** icon to exit the instruction stepping mode. Single-step through the application and observe the instruction sequence in the source code window in the middle of the Debug perspective, and the values of the variables and registers.
- 7. Resume execution of the code by choosing **Run > Resume** or by clicking the **Resume** icon.
- 8. You can even add breakpoints in the application for further debugging.
- 9. Once you are done, terminate the debugger by selecting Voltage_Monitor_MSS_MSS_CM3_0_app Debug in the Debug view, then right-clicking and selecting **Terminate and Remove**.



SC Debug - Voltage	_Moniter_MSS_MSS_CM3_0_app/main.c - Microsen
File Edit Source Re	factor Navigate Search Project Run Window Help
i 📬 • 🔝 🖆 💼) 🏇 • 🔘 • 🏊 • 🅭 🛷 • 🌛 🖢 * 🖗
🏇 Debug 🛛	🗖 🗖 🛋 Modu
	📕 🖓 🧟 🚑 🦛 👼 👘 🧖 🦷
	er MSS MSS CM3 0, ann Dehun (Microsemi Cortex-M
🖃 🔗 Embedder 🖻 🛷 Threa	Find C
= 1	
🚽 C:\Progra	Drop To Frame
	6 Restart
	🚴 Step Into 🛛 🖓
	Step Over F€
	_in≥Step Return Fi
-	i⇒ Instruction Stepping Mode
	😿 Use Step Filters
<	😽 Resume Without Signal
🚺 main.c 🛛 🚺	III Resume F8
25	00 Suspend
26#define	📕 Terminate 🛛 🔤
27"\n\r \	🖐 Terminate and Relaunch
28** **	∆ ¶ Disconnect
30* * * *	Remove All Terminated
31* * *	Relaunch
	SCEdit Voltage_Moniter_MSS_MSS_CM3_0_app Debug
	Wedit Source Lookup
🖳 Console 🛛 🔪	Terminate and Remove
Voltage_Moniter_MSS_ Transfer rate	Terminate/Disconnect All
set \$sp = *0x(Properties

Figure 38 · Terminating the Program

- 10. Close the Debug perspective by selecting Close Perspective from the Window menu.
- 11. Close the voltage monitor project by selecting the project name in the SoftConsole Project Explorer view, right–clicking, and selecting **Close Project**.
- 12. Close SoftConsole using File > Exit.
- 13. Close HyperTerminal using File > Exit. Click Yes, when prompted for closing immediately.

Step 9 - Building an Executable Image in Release mode

You can build an application executable image in "release mode" and load it into eNVM for executing code in eNVM of SmartFusion cSoC device. You can load the application executable image into eNVM with the help of eNVM data storage client from SmartDesign MSS Configurator and IAP or FlashPro programming software. In the release mode, you cannot use the SoftConsole debugger to load the executable image into eNVM.

For steps to build an executable image for our application refer the tutorial *SmartFusion: Building Executable Image in Release Mode and Loading into eNVM*.

This concludes the tutorial.



Appendix A – Libero SoC Vault/Repository Settings

Listed below are the steps to show how to configure your vault location and set up the repositories in Libero SoC.

- 1. Click Project > Vault/Repositories Settings.
- The Vault/Repositories Settings window is displayed. Click Repositories and add the following in the address field:
 - www.actel-ip.com/repositories/SgCore
 - www.actel-ip.com/repositories/DirectCore
 - www.actel-ip.com/repositories/Firmware

Note: Click Add after entering each path.

🔜 Vault/Repositorie	s Settings	? 🔀
Repositories Vault location	www.actel-ip.com/repositories/SgCore www.actel-ip.com/repositories/DirectCore www.actel-ip.com/repositories/Firmware	Add Remove Defaults
Help	ОК	Cancel

Figure 39 · Setting Repositories

3. Click on **Vault location** in the **Vault/Repositories Settings** window. Browse to a location on your computer to set the vault location where the IPs can be downloaded from the repositories.

🔜 Vault/Repositorie:	s Settings 🔹 🤶 🔀
 Repositories Vault location 	Current vault location: C:\Documents and Settings\donthus\Application I
	Select new <u>vault location</u> : C:\Documents and Settings\donthus\Application Data\Actel\tools
Help	OK Cancel

Figure 40 · Setting the Vault Location

4. Click OK.



Appendix B – Firmware Catalog Settings

- 1. Open the <Libero Installation directory>\Designer\bin\catalog.exe.
- 2. Select Tools > Vault/Repositories Settings, from the Firmware Catalog widow.

😋 Reload Catalog				
Vault/Repositories Settings			<u>Search</u> by all fields (<u>30/30</u>):	
	ones			Į -
display only the latest version of a core				•
Vame	 Version 	Size (MB)	Status	^
CoreAI Driver	3.0.101	0.43		
IoreAhbNvm Driver	2.1.102	0.23		
oreGPIO Driver	3.0.101	0.55		
IoreI2C Driver	2.0.103	0.43		
CoreInterrupt Driver	2.1.102	0.2		
IorePWM Driver	2.0.102	0.26		
oreSPI Driver	2.3.103	0.31		
CoreTimer Driver	2.1.101	0.3		
CoreUARTapb Driver	2.1.102	0.39		
CoreWatchdog Driver	2.1.101	0.26		
usion-II IAR EWARM flash loader	1.0.101	0.049		
usion-II IAR Embedded Workbench device deso		0.022		
usion-II SPI Driver	1.0.3	0.11		
usion2 In-Application-Programming	1.0.2	0.56		
Hardware Abstraction Layer (HAL)	2.1.102	0.21		
lardware Abstraction Layer (HAL)	2.1.1 (*)	0.13	2.1.102 is available for download	
martFusion CM515-PAL	2.1.100	0.36		
imartFusion CMSIS-PAL	2.0.100 (*)	0.34	2.1.100 is available for download	
SmartFusion Chip Boot	1.3.1	0.091		
imartFusion MSS ACE Driver	2.0.102	2.3		
SmartFusion MSS Ethernet MAC Driver	2.0.103	0.28		
imartFusion MSS GPIO Driver	2.0.103	0.75		
imart usion MSS I2C Driver	2.0.103	0.55		
imart usion MSS Peripheral DMA Driver	2.0.102	0.33		
				~
SmartFusion M55 SPI Driver No core selected	2.0.101	0.77		

Figure 41 · Firmware Catalog Settings

- 3. Select Repositories in the Vault/Repositories Settings dialog box.
- 4. Confirm that the following repositories are displayed (add them if needed):
 - www.actel-ip.com/repositories/SgCore
 - www.actel-ip.com/repositories/DirectCore
 - www.actel-ip.com/repositories/Firmware
- 5. Add the above mentioned paths in the address field if required by selecting the repository and clicking **Add**.
- 6. If new cores are available for download, click **Download them now!** to download the new cores to the vault.



List of Changes

Revision	Changes	Page
Revision 4	Modified Tutorial Requirements section in Introduction (SAR 38347)	3
(April 2012)	Modified Step 2 - Configuring MSS Peripherals (SAR 38347)	5
	Replaced Figure 2 · (SAR 38347)	6
	Modified Step 4 - Building the Project (SAR 38347)	17
Revision 3 (November 2011)	Updated the document for Libero SoC v10.0 (SAR 35046).	NA

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.



Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call **800.262.1060** From the rest of the world, call **650.318.4460** Fax, from anywhere in the world **408.643.6913**

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

Visit the Microsemi SoC Products Group Customer Support website for more information and support (http://www.microsemi.com/soc/support/search/default.aspx). Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on website.

Website

You can browse a variety of technical and non-technical information on the Microsemi SoC Products Group home page, at http://www.microsemi.com/soc/.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Sales office listings can be found at www.microsemi.com/soc/company/contact/default.aspx.

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech_itar@microsemi.com. Alternatively, within My Cases, select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.



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**.

© 2012 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.