Displaying POT Level with LEDs

Libero SoC and µVision4 IDE Flow Tutorial for SmartFusion cSoC





Table of Contents

Introduction	3
Tutorial Requirements	
Working with Libero SoC and µVision	5
Step 1 - Creating a Libero SoC Project	5
Step 2 - Configuring MSS Peripherals	7
Step 3 - Generating the MSS Component	15
Step 4 - Generating the Program File	17
Step 5 - Programming SmartFusion Board Using FlashPro	18
Step 6 - Building the Software Application through Keil µVision [®] 4 IDE	
Step 7 - Configuring the Serial Terminal Emulation Program	26
Step 8 - Installing Drivers for the USB to RS232 Bridge	29
Step 9 - Connecting the ULINK-ME to the Board and PC	29
Step 10 - Debugging the Application Project Through μ Vision4	32
Appendix A – Libero SoC Catalog Settings	33
Appendix B – Firmware Catalog Settings	35
Appendix – C	37
List of Changes	43
Product Support	45
Customer Service	45
Customer Technical Support Center	45
Technical Support	45
Website	45
Contacting the Customer Technical Support Center	45



Introduction

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

- Creating and implementing a Libero® system-on-chip (SoC) v10.0 project using a SmartFusion cSoC
- Configuring the peripherals using SmartDesign
- Configuring the analog compute engine (ACE)
- · Generating the microcontroller subsystem (MSS) Component
- · Generating the programming file to program the SmartFusion cSoC device
- Opening the project in µVision[®]4 IDE and Debugger tool from Libero SoC and writing application code
- Compiling application code
- · Creating and launching a debug session
- Debugging and running the code using Keil µVision[®]4 IDE

Tutorial Requirements

Software Requirements

This tutorial requires the following software installed on your PC:

- Libero SoC v10.0 (or later) can be downloaded from www.microsemi.com/soc/download/software/libero/default.aspx.
- Keil µVision[®]4.11 version or later

Hardware Requirements

This tutorial requires the following hardware:

- SmartFusion Evaluation Kit Board or SmartFusion Development 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 the PC.
- Keil supplied ULINK2 or ULINK-ME debugger hardware (not supplied with the SmartFusion Kit Board)

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 Libero Keil POTlevel tut DF.

Note: Extract design files to root directory.

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



MSS Components Used

- ARM[®] Cortex[™]-M3 processor
- Clock conditioning circuitry (CCC)
- General purpose input/output (GPIO)
- UART_0
- Analog compute engine (ACE)

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 instruct how to configure SmartFusion analog channels and ACE that is used to monitor the voltage across the potentiometer (POT). The UART is used to send the analog-to-digital converter (ADC) results to a terminal program.

Design Steps

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

- Create a Libero SoC v10.0 project for SmartFusion cSoC.
- Configure the SmartFusion cSoC peripherals.
- Generate the SmartFusion cSoC MSS component.
- Perform synthesis and layout, and generate a programming file to program the SmartFusion cSoC device.
- Program the SmartFusion A2F200M3F or A2F500M3F cSoC device.
- Open the software project in µVision[®]4 IDE and write the application program.
- Run an application to monitor the voltage across the POT on the SmartFusion Evaluation Kit Board or Development Kit Board.

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 POT and four flags are included for the voltage monitoring. These flags are used to drive the four LEDs on the board.



Working with Libero SoC and µVision

This section describes how to create a Libero SoC project, configure the microcontroller subsystem (MSS), program the design on the SmartFusion board, and run an application program in the µVision4 IDE.

Step 1 - Creating a Libero SoC Project

- 1. Launch Libero SoC v10.0 (or later).
- 2. From the Project menu, select New Project. Enter the information as displayed in Figure 1 · .
 - Name: Voltage_Monitor
 - Location: <..> (For example, C:\Microsemiprj\ POT_LED_Libero_Keil)
 - Family: SmartFusion
 - 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 and leave others as default

Project Name: Voltage_Monitor Location: C:/Microsemipri/POT_LED_Libero_Kell Browse Prefered HDL type: Verliog VHDL Description: Edit Tool Profiles Device Family: SmartFusion AEF200MSF Package: 464 FBSA Yes Speed: STD I.S Operating Conditions: COM Speed: STD YCCA Voltage (in volts) 1.6 1.5 1.425 YCCI 1.5 Woltage (in volts) 1.6 1.7 YCCI 2.5 Woltage (in volts) 1.6 1.7 YCCI 3.3 Woltage (in volts) 3.6 3.3 Show only latest version Persign Template Core Version Yes Show only latest version Yes	New Project				?
Name: Volkage_Monitor Location: C:/Microsemipr//POT_LED_Libero_Keil Browse Prefered HDL type: O Verdag VHDL Description:	Project				
Location: C:/Microsemipr//POT_LED_Libero_Kell Browse Prefered HDL type: ● Verlog ● VHDL Description:		Voltage Monitor			
Prefered HDL type: ● Verilog ● VHDL Description: ●			Keil		Browse
Description: Edit Tool Profiles Device Family: SmartFusion Die: A2F200M3F Package: 494 FBGA Speed: 5TD Die Voltage: 1.5 Operating Conditions: COM Temperature (in degrees Celsius) VCC1 1.5 Voltage (in volts) VCC1 1.5 Voltage (in volts) VCC1 1.5 Voltage (in volts) VCC1 2.5 Voltage (in volts) VCC1 3.3 Voltage (in volts) VCC1 3.3 Voltage (in volts) VCC1 5.5 Voltage (in volts) VCC1 5.5 Voltage (in volts) VCC1 5.5 Voltage (in volts) VCC1 3.3 Voltage (in volts) VCC1 5.5 Volt					
Edit Tool Profiles Device Panily: Die: A2F200M3F Package: 484 FBGA Speed: Speed: Speed: 1.5 Operating Conditions: COM Temperature (in degrees Celsius) vCCA Voltage (in volts) vCCA 1.5 Voltage (in volts) vCCA 2.5 Voltage (in volts) vCCA 2.5 Voltage (in volts) vCCA 3.3 Voltage (in volts) vCCA 1.5 Voltage (in volts) vCCA 2.5 Voltage (in volts) vCCA 2.5 Voltage (in volts) vCCA 3.3 Voltage (in volts) 2.7 2.5 2.3 vCCA 3.3 Voltage (in volts) 2.6 wCCA 3.3 Voltage (in volts) 2.7 2.5 vCCA 2.5 Voltage (in volts) 2.6 3.6 3.3 3		Verlig V vrbc			
Device Family: SmartFusion Die: A2F200M3F Package: 484 FBGA Package: 484 FBGA Package: 1.5 Package: 1.5 Poerating Conditions: COM Die Voltage (in volts) VCCA Voltage (in vo	Description:				
Device Family: SmartFusion Die: A2F200M3F Package: 484 FBGA Package: 1.5 Package: 5TD Die Volkage: 1.5 Die Volkage: 1.5 Poperating Conditions: COM Temperature (in degrees Celsius) VCCA Volkage (in volts) VCCA Volkage (in					
Family: SmartFusion Die: A2F200M3F Package: 484 FBGA Speed: STD Die Voltage: 1.5 Die Voltage: 1.5 Operating Conditions: COM VCC1 I.5 Voltage (in volts) 1.6 VCC1 I.5 Voltage (in volts) 1.6 VCC1 I.5 Voltage (in volts) 1.6 VCC1 I.5 Voltage (in volts) 2.7 VCC1 I.5 Voltage (in volts) 2.7 VCC1 I.5 Voltage (in volts) 3.6 VCC1 I.5 Voltage (in volts) 2.7 VCC1 I.5 Voltage (in volts) 2.7 VCC1 I.5 Voltage (in volts) 2.7 VCC1 I.5 Voltage (in volts) 2.6 VCC1 I.5 Voltage (in volts) 2.7 VCC1 I.5 Voltage (in volts) 2.7 VCC1 I.5 Voltage (in volts) 2.6 VCC1 I.5 Voltage (in volts) 2.6 Stow only latest version Stow only latest version	L Edit Tool Profiles				
Die: A2F200M3F ♥ Package: 484 FBGA ♥ Speed: 5TD ♥ Die Voltage: 1.5 ♥ Operating Conditions: COM ♥ Temperature (in degrees Celsius) 0 25 85 VCCA Vibrage (in volts) 1.575 1.5 1.425 VCCA Vibrage (in volts) 1.6 1.5 1.4 VCCA Vibrage (in volts) 2.7 2.5 2.3 VCCI 3.3 Voltage (in volts) 3.6 3.3 3 Design Template € Use template Core Version SmartFlusion Microcontroller Subsystem (M55) 2.5.105 § Show only latest version	Device				
Package: 484 FBGA Speed: STD Die Volkage: 1.5 Operating Conditions: COM VCCA Volkage (n vols) 0 VCCA Volkage (n vols) 1.575 VCC1 1.5 Volkage (n vols) 1.6 VCC1 1.5 Volkage (n vols) 1.6 VCC1 2.5 Volkage (n vols) 1.6 VCC1 3.3 Volkage (n volts) 2.7 VCC1 3.3 Volkage (n volts) 3.6 VCC1 3.3 Volkage (n volts) 2.7 VCC1 3.3 Volkage (n volts) 2.5 VCC1 3.5 VCC1 3.5 VCC1 3.5 <td>Family:</td> <td>SmartFusion 💌</td> <td></td> <td></td> <td></td>	Family:	SmartFusion 💌			
Speed: STD V Die Voltage: 1.5 V Operating Conditions: COM V Temperature (in degrees Celsius) 0 25 85 VCCA Voltage (in volts) 1.575 1.5 1.425 VCCI 1.5 Voltage (in volts) 1.6 1.5 1.4 VCCI 2.5 Voltage (in volts) 2.7 2.5 2.3 VCCI 3.3 Voltage (in volts) 3.6 3.3 3 Design Template Core Version SmartFusion Microcontroller Subsystem (MSS) 2.5,106 SmartFusion Microcontroller Subsystem (MSS) 2.5,106	Die:	A2F200M3F			
Die Voltage: 1.5 ♥ Operating Conditions: OM ♥ Temperature (in degrees Celsius) 0 25 85 VCCA Voltage (in volts) 1.575 1.5 1.425 VCCI 1.5 Voltage (in volts) 1.6 1.5 1.4 VCCI 2.5 Voltage (in volts) 2.7 2.5 2.3 VCCI 3.3 Voltage (in volts) 3.6 3.3 3 Design Template © Use template Core Version SmartFusion Microcontroller Subsystem (MSS) 2.5105 SmartFusion Microcontroller Subsystem (MSS)	Package:	484 FBGA			
Operating Conditions: COM N Temperature (in degrees Celsius) 0 25 85 VCCA Voltage (in volts) 1.575 1.5 1.425 VCC1 1.8 Voltage (in volts) 1.6 1.5 1.4 VCC1 2.8 Voltage (in volts) 1.6 1.5 1.4 VCC1 2.8 Voltage (in volts) 2.7 2.5 2.3 VCC1 3.3 Voltage (in volts) 3.6 3.3 3	Speed:	STD 💌			
Temperature (in degrees Celsius) Best Typical Worst VCCA Voltage (in volts) 1.575 1.5 1.425 VCCI 1.5 Voltage (in volts) 1.6 1.5 1.4 VCCI 2.5 Voltage (in volts) 1.9 1.8 1.7 VCCI 3.5 Voltage (in volts) 2.7 2.5 2.3 VCCI 3.5 Voltage (in volts) 3.6 3.3 3	Die Voltage:	1.5 💌			
Temperature (in degrees Celsus) 0 25 85 VCCA Witkage (in volts) 1.575 1.5 1.425 VCC1 I.5 Voltage (in volts) 1.6 1.5 1.4 VCC1 Voltage (in volts) 1.9 1.8 1.7 VCC1 Voltage (in volts) 2.7 2.5 2.3 VCC1 3.3 Voltage (in volts) 3.6 3.3 3	Operating Conditions:	СОМ			
VCC1.1.5 Voltage (in volts) 1.6 1.5 1.4 VCC1.2.5 Voltage (in volts) 1.9 1.8 1.7 VCC1.2.5 Voltage (in volts) 2.7 2.5 2.3 VCC1.3.3 Voltage (in volts) 3.6 3.3 3 Design Template ✓ Version 2.5 2.5.106 SmartFusion Microcontroller Subsystem (M55) 2.5.106 ✓ Show only latest version	Temper	ature (in degrees Celsius)	0	25	85
VCCI 2.5 Voltage (in volts) VCCI 3.3 Voltage (in volts) 2.7 2.5 2.3 VCCI 3.3 Voltage (in volts) 3.6 3.3 3 Design Template V Use template Core Version SmartFusion Microcontroller Subsystem (MSS) VSCI 2.5,106 SmartFusion Microcontroller Subsystem (MSS)	VCCI 1	.5 Voltage (in volts)	1.6	1.5	1.4
Design Template	VCCI 2.	.5 Voltage (in volts)	2.7	2.5	2.3
Use template Core Version SmartEusion Mitrocontroller Subsystem (MSS) 2.5.105 Show only latest version	VCCI 3	.3 Voltage (in volts)	3.6	3.3	3
Use template Core Version SmartFusion Microcontroller Subsystem (MSS) 2.5.105 Show only latest version	Design Template				
SmartFusion Microcontroller Subsystem (MSS) [2.5.106 ✓ Show only latest version	Use template				
Show only latest version		Core			Version
	SmartFusion Microcontro	ler Subsystem (MSS)			2.5.106
Help OK Cancel				🗹 SH	now only latest version
Help OK Cancel					
	Help				OK Cancel

Figure 1 · New Project Dialog Box



3. Click Edit Tool Profiles and add Keil by clicking on Software IDE as shown in Error! Reference source not found.

🗢 Tool Profiles				? 🗙
Tools Software IDE Synthesis Simulation Programming	🗢 Edit Profile	Software IDE profiles	?	
	Name: Tool integration: Version: Location:	Keil <mark>Keil</mark> Unknown C:\Keil\UV4\Uv4.exe	Browse	tConsole Embedded Workbench 6.0 m.exe
	Help		OK Cancel	<u></u>
Help			Export Profiles	OK Cancel

Figure 2 · Selecting Kiel as Software IDE

- After adding the Profile, click OK to close the Tool Profiles Dialog Window. Repeat the steps (3 and 4) above for Synthesis, Simulation, and Programming and then click OK to close the Tool Profile dailog window.
- 5. Select the MSS core in New Project Dialog Box and click OK.
 - Note: If SmartFusion cSoC MSS does not appear in the list, refer to the Appendix A Libero SoC Catalog Settings to find out how to set your repositories. If your vault does not have the MSS core, download the core by double clicking on the core name in Design template in New Project Dialog Box.

6. The project is created and the Libero SoC window is displayed as shown in Figure 3 · . The SmartDesign "Voltage_Monitor" is created with the instantiation of the MSS component.

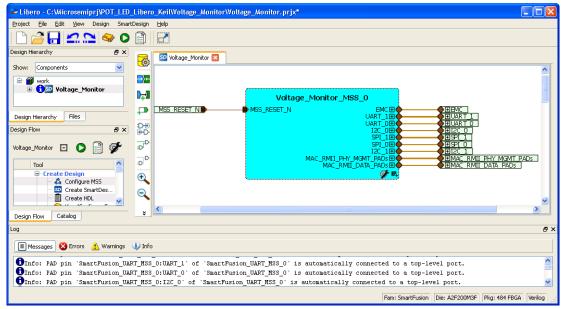


Figure 3 · The Libero Window After Completing New Project Wizard

Step 2 - Configuring MSS Peripherals

1. Double-click on **Voltage_Monitor_MSS_0** to configure the MSS. The MSS is displayed in the SmartDesign Canvas in a new tab as shown in Figure 4 · .

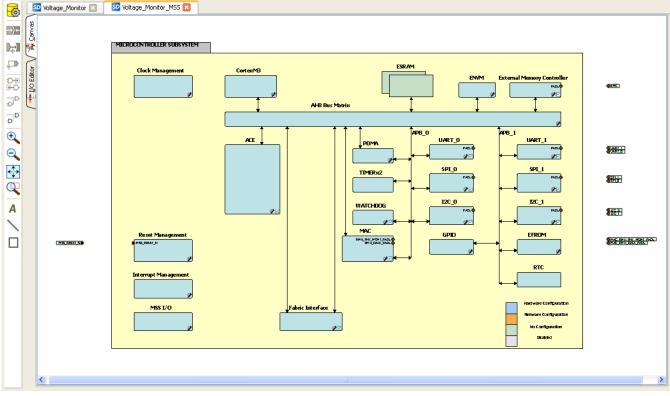


Figure 4 · MSS in the SmartDesign Canvas

The enabled MSS are highlighted in blue, and can be configured in the 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, or clear the 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 is shown in Figure $5 \cdot$.

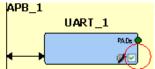
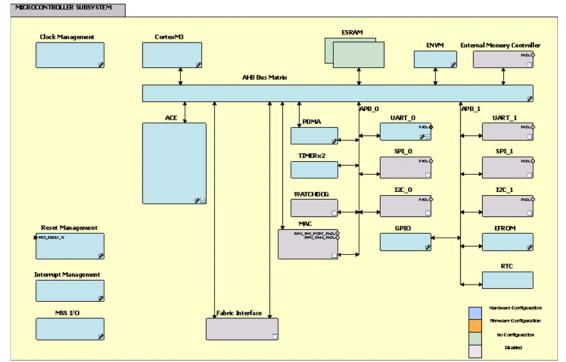
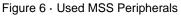


Figure 5 · Enabling the Peripheral

This example uses only the clock management, analog compute engine (ACE), GPIO, and UART_0 peripherals.

2. Disable the following peripherals: MAC, WATCHDOG, Fabric Interface, SPI0, SPI1, I2C0, I2C1, UART1, and EMC.

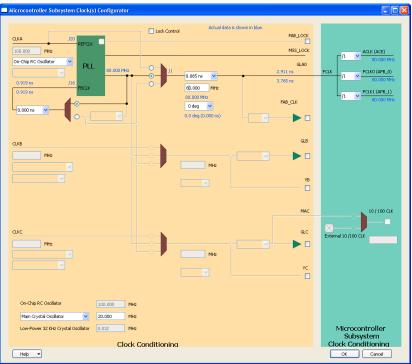




- 3. Double-click the **Clock Management** block and configure as shown below:
 - CLKA: On-chip RC Oscillator
 - MSS clock source: PLL output
 - MSS clock frequency: 80 MHz

Use default settings for all other fields.





4. After completing the configuration, click OK

Figure 7 · MSS Clock Configuration

Configuring ACE

- 1. To configure ACE, double-click the ACE peripheral block and configure as follows:
 - Connect TM0 to the POT on the SmartFusion Evaluation Kit Board or Development Kit Board. Configure a voltage monitor to measure the voltage across the POT and also to 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 are used to illuminate the LEDs on the SmartFusion Evaluation Kit Board or Development Kit Board.

nfigure ACE					
Configure ACE	Controller	Flags			
ACL <u>K</u> : 80 MHz	ADC Clock: 10 MHz	Regolution: 12 vits		Advanced Opti	ons
Add service:					X
Active Bipolar Prescaler Type AbC Presc Type Current Input Differential Input Temperature Input U/TT Linput Analog Comparator Signa Deka DAC	Service	Signal	Sample time [us]	Package Pin	SCB
	ADC Block 0 ADC Block 1				
				ОК	Cancel

Figure 8 · MSS ACE Configuration



- 2. Select ADC Direct Input > Add (or, double-click ADC Direct Input) and enter the parameters as shown in Figure 9 · :
 - Signal name: TM0_Voltage
 - Send raw results to DMA: Cleared check box
 - Acquisition time: 10 µs
 - Filtering factor: None

Configure ADC Direc	t Input:							×
_								
тмо	<u>S</u> ignal na	me:						
	TM0_i/ol	tage		Send raw Al	DC result to D	MA		
Acguisition time:	10.	000 us						
Digital filtering —				Linear trans	formation			
Filtering factor:	: Nor	ie 💌		Scale factor:				
Ini <u>t</u> ial value;	0.0	00 V		Offset:				
Send filtere	d result to	DMA		Send tran	nsformed resu	ilt to DMA		
		Thre	eshold Detectio	n		*	\times	
Flag	Name	Flag Type	Threshold (∀)	Hysteresis (m¥)	Assert Samples	De-asse Sample		
over_1p		OVER	1	1				
over_1p over_2p		OVER OVER	1.5	1			_	
over_2p		OVER	2.5	1			-	
		ocessing completed						
		conplote	•					
					ОК		ancel	

Figure 9 · MSS ADC Direct Input Configuration

3. Next, add the flags as shown in Table 1:

Table 1 · Flag Definitions

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

4. Click OK.



5. Assign the ADC Direct Input Signal to package pin W8 in the Configure ACE dialog box. The Configure ACE tab is displayed as shown in Error! Reference source not found.

Configure ACE	Controller	Flags				
ADC Configuration						
ACL <u>K</u> : 80 MHz	ADC Clock: 1	0 MHz Regolution:	12 v bits		Advanced Ogti	ons
Add service:						_/ ×
Active Bipolar Prescaler Inpu ADC Direct Input	Service		Signal	Sample time (us)	Package Pin	SCB
Current Input Differential Input	ADC Direct Input	TM0_Voltage		10.000 W8 (T)	401	✓ 0(4)
Temperature Input LVTTL Input						
Analog Comparator Sigma Delta DAC						
Add >>						
	ADC Block 0					
	ADC Block 0					

Figure 10 · MSS ACE Configuration With ADC Direct Input

on dialog is

		ne Controller tab (n erating sequence e		
Configure ACE	Controller	Flags		
Procedures		• = / ×		
ADCO_MAIN ADC1_MAIN	Name		Block 0 Block 1	
) Operating sequence entry:	O Auto 💿 Manual			
Available signals:	Sampling rate		Actual Rate (ksps) erating sequence slot SE Execution Time (us)	

Figure 11 · MSS ACE Configuration to Enable Sampling Sequence



- 8. Click Insert operating sequence slot as shown in Figure 11 · .
- 9. Select SAMPLE.

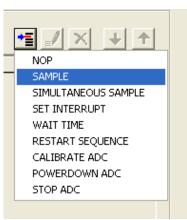


Figure 12 · Select SAMPLE

10. The Configure SAMPLE window is displayed. Select TM0_voltage and click OK.

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

Figure 13 · Configure SAMPLE

11. Click Insert operating sequence slot again and select RESTART SEQUENCE.

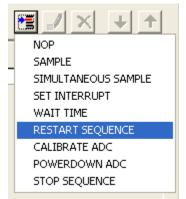


Figure 14 · Select Restart Sequence



12.	Click	Calc	culate	Actual	Rate.
-----	-------	------	--------	--------	-------

Configure ACE Controller Procedures Name ADC: MANN ADC: MANN ADC: MANN Configure ACE Name ADC: Block.0 ADC: Block.0 Configure ACE Name ADC: MANN Configure ACE Name Configure ACE Name Configure ACE Configure ACE <th>Configure ACE</th> <th>X</th>	Configure ACE	X
Name ADC_MAIN ADC_MAIN Operating sequence entry: Auto Operating sequence Calculate a round robin operating sequence of the specified signals. Calculate a crund robin operating sequence of the specified signals. Calculate a crund robin operating sequence for this timeslot	Configure ACE Controller Flags	
ADCC MAIN ADC Block 0 ADC Block 1 Calculate structure Calculate a round robin operating sequence of the specified signals. Calculate Actual Rate Calculate A	Procedures	
ACCI_MAIN Persiting sequence entry: Auto Manual Petails of procedure: ACC0_MAIN Available signals: Sampling rate Calculate a round robin operating sequence of the specified signals. Calculate Actual Rate Calculate Actual Rate Instruction SSE E Execution Sample TM0_Voltage Instruction SSE E Execution Sample TM0_Voltage Instruction Sample TM0_Voltage Instruction Sequence Actual Rate Actual Rate Actual Rate Actual Rate Actual R		
Details of procedure: ADC0_MAIN Available signals: Signal Actual Rate Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Sig	ADC1_MAIN ADC Block 1	
Details of procedure: ADC0_MAIN Available signals: Signal Actual Rate Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Signal Image: Sig		
Available signals: Sampling rate Image: Signal i		
Signal [ksps] IM0_Voltage 0.000 Calculate a round robin operating sequence of the specified signals. Calculate Actual Rate Calculate a round robin operating sequence of the specified signals. Operating sequence SSE Execution fine [us] Sample TM0_Voltage 11.725 Restarts the execution sequence for this timestot 0.100	Available signals: Sampling rate	
Calculate a round robin operating sequence of the specified signals. Calculate Actual Rate Operating sequence Instruction Sample TM0_Voltage Restarts the execution sequence for this timeslot		
Calculate a round robin operating sequence of the specified signals.		
Calculate Actual Rate Operating sequence Instruction Sample TM0_Voltage Restarts the execution sequence for this timeslot		
Instruction SSE Execution Time (us) Sample TM0_Voltage 11.725 Restarts the execution sequence for this timeslot 0.100		
Instruction Time (us) Sample TM0_Voltage 11.725 Restarts the execution sequence for this timeslot 0.100	Operating sequence	
Restarts the execution sequence for this timeslot	Time (us)	
OK Cancel		
	OK Cancel	

Figure 15 · MSS ACE Configuration: Final Controller Tab

13. The **Controller** tab window is displayed as shown in Figure $16 \cdot :$

jure ACE								
Configure ACE	Controlle	r	Flags					
Procedures			• = / ×					
	Name							
ADC0_MAIN				ADC B				
ADC1_MAIN								
	0.11	O •• •						
Operating sequence entry:	🔾 Auto	 Manual 						
	ADC0_MAIN							
Available signals:	S	ampling rate						
	->		Signal		Actual Rate (ksps)			
	->>	TM0_Voltage			83.200			
	<-							
	<<-							
	Calculate Actua	al Rate	<u>⊺</u> otal sar	npling rate:	83.200 ksps			
Operating sequence				*	X¥↑			
		nstruction		\$5	E Execution Time (us)			
Sample TM0_Volta Restarts the execut	je	his the state		_	11.900			
n estarts the execut	on sequence for t	nis unesiot			0.080			
						ОК	Cancel	

Figure 16 · MSS ACE Configuration: Controller Tab

- 14. Click the Flags tab in the Configure ACE window. This tab lists the flags set from PPE registers.
- 15. Click the + sign to expand the Flag registers group. The PPE_FLAGSn registers contain the userdefined flags.



16. Select **PPE_FLAGS0** (FLAGBANK0). 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.

elect a register to view ACE flag mapping:	Available	bits of PPE_FLAG50 register:			
Fabric Flag Registers	Bit	Source	ACE	Cortex-M3	<u>^</u>
PPE_FLAG50 (0x40021450)	0	TM0_Voltage:over_1p0v	54	118	
PPE_FLAG51 (0x40021454)		TM0_Voltage:over_1p5v	55	119	
- PPE_FLAG52 (0x40021458)		TM0_Voltage:over_2p0v	56	120	
		TM0 Voltage:over 2p5v	57	121	
PPE_SFFLAGS (0x40021450)	4	<none></none>	58	122	
PPE_SHFLAGS (0X40021460)	5	<none></none>	59	123	
	6	<none></none>	60	124	
		<none></none>	61	125	
	8	<none></none>	62	126	
		<none></none>	63	127	
		<none></none>	64	128	
		<none></none>	65	129	
		<none></none>	66	130	
		<none></none>	67	131	
		<none></none>	68	132	
		<none></none>	69	133	
		<none></none>	70	134	
		<none></none>	71	135	
		<none></none>	72	136	
		<none></none>	73	137	
		<none></none>	74	138	
		<none></none>	75	139	
	22	<none></none>	76	140	
		<none></none>	77	141	
		<none></none>	78	142	
		<none></none>	79	143	
	29	<none></none>	83	147	~
	26 27 28 29	<none> </none>	80 81 82 83	145 145 146 147	

Figure 17 · ACE Flag Mapping - PPE Flag Registers

17. Click **OK** to close the ACE configuration window.

Configuring the GPIO Peripheral

- Note: If you are not using the SmartFusion Evaluation Kit Board Revision 2 or later, or using the SmartFusion Development Kit Board, follow Appendix C, Skip Step 3 Generating the MSS Component and Step 4 Generating the Program File.
 - 1. Double-click the **GPIO** block in the **MSS component**, configure as shown in Figure $18 \cdot$, and click **OK**.

55	Configuring MSS_GPI0_0 (MSS_GPI0 - 1.0.101)	
	Configuration	
	Multiplexed With I2C1 Peripheral Dedicated I/Os	
	GPIO_31: Use as MSS I/O Pad Output V U20 or connect to Fabric Not Used V	
	GPIO_30: Use as MSS I/O Pad Output V22 or connect to Fabric Not Used V	
	Multiplexed With UART1 Peripheral Dedicated 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 V	
	Multiplexed With SPI1 Peripheral Dedicated 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 Lised V AA22 or connect to Fabric Not Lised V	~
	Нер 🗸	ancel

Figure 18 · Configure MSS_GPIO_0

- 2. This example requires GPIO_31, GPIO_30, GPIO_29, and GPIO_28 to be connected to LED_8 to LED_5 on the SmartFusion Evaluation Kit Board.
- 3. Click **File > Save** to save the Voltage_Monitor_MSS.

Step 3 - Generating the MSS Component

1. Right-click on Voltage_Monitor_MSS_0 on the Voltage_Monitor_MSS tab and select Update Instance(s) with Latest Component... as shown in Figure 19 · .

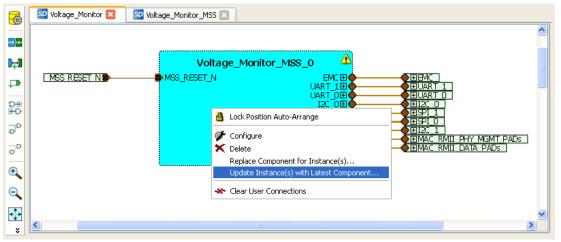


Figure 19 · Updating the MSS

2. Click **Design > Configure Firmware** as shown in Figure 20 · .

Project	File	Edit	View	Design	SmartDesign	Help
1 P o	3		20	🈔 Coni	figure Firmware	
				🜔 Gen	erate Programm	ing Data
Design Hi	erarchy	<u> </u>		-		
				🗐 Rep	orts	
-1	-	•				

Figure 20 · Opening Design_Firmware

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 **Keil-MDK** as the configuration.

 ✓ ✓	0 ACE_Driver_0	HAL		Voltage_Monitor_MSS
MSS_4	ACE_Driver_0			
		MSS_ACE_Driver	2.3.105	Voltage_Monitor_MS5:MS5_ACE_0
MSS_C	GPIO_Driver_0	MSS_GPIO_Driver	2.0.105	Voltage_Monitor_MSS:MSS_GPIO_0
MSS_I	IAP_Driver_0	MSS_IAP_Driver	2.2.101	Voltage_Monitor_MSS
MSS_)	NVM_Driver_0	MSS_NVM_Driver	2.2.102	Voltage_Monitor_MSS:MSS_ENVM_0
MSS_F	PDMA_Driver_0	MSS_PDMA_Driver	2.0.102	Voltage_Monitor_MSS:MSS_DMA_0
MSS_F	RTC_Driver_0	MSS_RTC_Driver	2.0.100	Voltage_Monitor_MSS:MSS_RTC_0
MS5_1	Timer_Driver_0	MSS_Timer_Driver	2.1.101	Voltage_Monitor_MSS:MSS_TIMER_0
MSS_L	UART_Driver_0	MSS_UART_Driver	2.3.101	Voltage_Monitor_MSS:MSS_UART_0
🖂 🐼 Smarti	Fusion_CMSIS_PAL_0	SmartFusion_CMSIS_PAL	2.3.103	Voltage_Monitor_MSS
		SmartFusion_CMSIS_PAL		

Figure 21 · Configuring SmartFusion_CMSIS_PAL_0

- 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. Appendix B – Firmware Catalog Settings
- 4. Click File > Save to save the Design_Firmware.



5. Save the design and generate the component by clicking Generate Component or by selecting SmartDesign > Generate Component from menu.

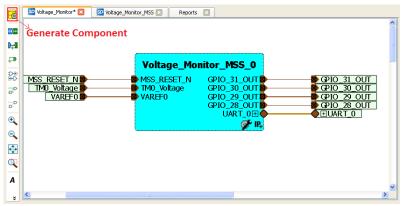


Figure 22 · Generating the MSS Component

- After successful generation of MSS component the log window displays the message "Info: 'Voltage_Monitor' was successfully generated. Open datasheet for details". The datasheet has the Project information like Generated files, used IO's and Memory map etc.
- 7. Confirm that the Keil folder is created with the subfolders and files shown in Figure 23 · .

Files	Β×
designer	
Georginal	
drivers	
drivers_config	
main.c	
Voltage_Monitor_hw_platform.h	
Voltage_Monitor_MS5_MS5_CM3_0.uvmp	w
Voltage Monitor MSS MSS CM3 0 app.	
Voltage_Monitor_MSS_MSS_CM3_0_app.	
Voltage_Monitor_MS5_MS5_CM3_0_hw	
phy_synthesis	
💷 🦳 simulation	
smartgen	
stimulus	
📄 🔁 synthesis	
🖮 🔁 viewdraw	
—	

Figure 23 · Files Window

Step 4 - Generating the Program File

Libero SoC provides the push button flow for Generating programming data of the project in a single step. By clicking **Generate Programming Data**, you can complete the synthesis, place and route, verify timing, and generate the programming file. You can also complete the flow by running the synthesis and place and route tools in interactive mode (step-by-step), for more information refer *Libero SoC Quick Start Guide*.

Push-Button Design Flow

 Click Generate Programming Data as shown in Error! Reference source not found. to complete the place and route, verify timing, and generate the programming file. This completes the fdb file generation.

Project	File	Edit	View	Design	Tools	Help	
- C	3		20	≥ 0			
Design Hier	rarchy	/			Generat	e Programming Data	₽×

Figure 24 · Build the Project

2. The **Design Flow** window looks as shown in Figure $25 \cdot .$

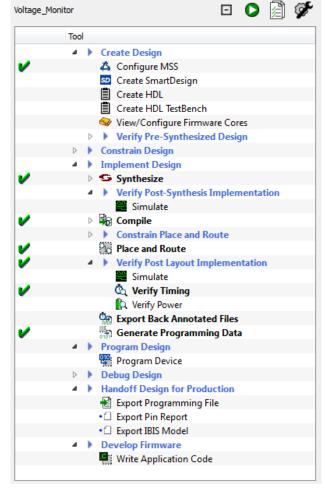


Figure 25 · Design Flow Window After Building the Project



Step 5 - Programming SmartFusion Board Using FlashPro

Before you proceed with programming the device, ensure that the low cost programming stick (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 *SmartFusion Development Kit User's Guide* for additional information.

Jumper Settings for SmartFusion Evaluation Kit Board

- 1. JP10: Short pin 1 and 2 using a jumper
- 2. JP7: Short pin 1 and 2 using a jumper for LCPS mode
- 3. J6: Connect pin 1 and 2 using the jumper
- 4. JP6: Connect pin 2 and 3 using the jumper
- 5. J13: Connect the USB cable to J13 connector. Install the FlashPro drivers if they are not already installed
- 6. J14: Connect second USB cable for power
- 7. JP11, JP12, JP13, and Jp14: Short pin 2 and 3 using a jumper (in A2F EVAL REV 2)

Jumper Settings for SmartFusion Development Kit

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 Table 2:

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	_

Table 2 · Jumper Settings for Development Kit Board



Programming the Device

- 1. Double click **Program Device** under **Program Design** in the **Design Flow** window to program the SmartFusion cSoC device.
- 2. Click Yes when it prompts that the I/O and timing constraints are not yet set.
 - Note: Do not interrupt the programming sequence; it may damage the device or the programmer. If you face any problems, contact Microsemi SoC Products Group Tech Support at soc_tech@microsemi.com.

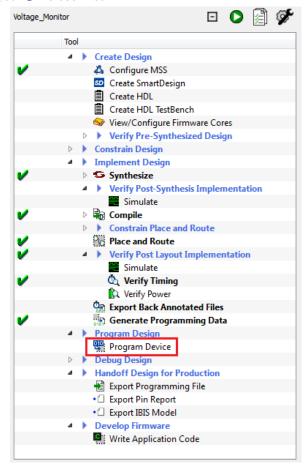


Figure 26 · Design Flow Window

You can also run FlashPro interactively by right clicking on **Program Device** in **Design Flow** window and selecting **Open Interactively**. For more information on FlashPro refer to the *FlashPro User's Guide*.



Step 6 - Building the Software Application through Keil μ Vision[®]4 IDE

1. From the Libero SoC, open the Keil project by double clicking on **Write Application Code** under **Develop Firmware** in **Design Flow** window.

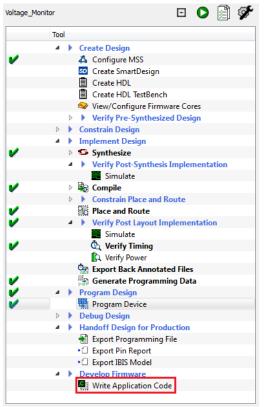


Figure 27 · Invoking µvision4 Project from Libero SoC

2. The µVision perspective looks like Figure 28 ·

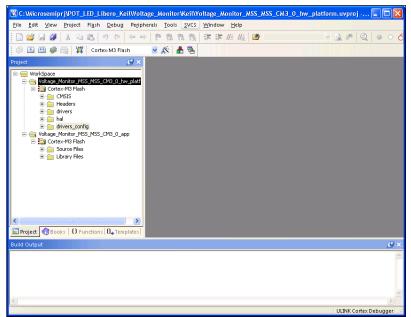


Figure 28 · µVision4 Project



```
3. Copy the code provided below and paste it in main.c file under the
   Voltage_Monitor_MSS_MSS_CM3_0_app project in the µVision editor and delete the existing code.
  * (c) Copyright 2011 Microsemi Corporation. All rights reserved.
   * Sample test program for the SmartFusion ACE. TMO is used to monitor the
   * voltage across the potentiometer. The UART is used to send the ADC results
   * to a terminal program. The hardware configuration has four flags:
      - over 1.0v
     - over 1.5v
      - over 2.0v
     - over 2.5v
   * The flag values are displayed on the SmartFusion eval board LEDs.
   #include "mss uart.h"
  #include "mss ace.h"
  #include "mss gpio.h"
  #include "ace_handles.h"
  #include <stdio.h>
  int main()
  {
     size t rx size;
     uint8 t rx buff[1];
      const uint8_t greeting[] = "Welcome to Microsemi's SmartFusion Voltage
  Monitor\n\r";
      const uint8 t instruction[] = "\n\rPress Any Key\n\r";
  int32 t flag status 2p5v, flag status 2p0v, flag status 1p5v, flag status 1p0v;
      uint32 t gpio output;
  /*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( &g_mss_uart0, greeting, sizeof(greeting) );
  MSS UART polled tx( &g mss uart0, instruction, sizeof(instruction) );
```

```
for (;;)
{
rx_size = MSS_UART_get_rx( &g_mss_uart0, rx_buff, sizeof(rx_buff));
if (rx size > 0)
{
uint8 t display buffer[32];
uint16_t adc_result;
int32 t adc value mv;
adc_result = ACE_get_ppe_sample( TM0_Voltage );
adc_value_mv = ACE_convert_to_mV( TM0_Voltage, adc_result );
if ( adc value mv < 0 )
snprintf( display buffer, sizeof(display buffer),
"-%d.%d V\n\r", -adc value mv / 1000, -adc value mv % 1000);
}
else
{
snprintf( display buffer, sizeof(display buffer),
"%d.%d V\n\r", adc value mv / 1000, adc value mv % 1000);
MSS UART polled tx string( &g mss uart0, display buffer );
           /* Checking the status of Voltage flags */
           flag status 2p5v = ACE get flag status(TMO Voltage over 2p5v);
           flag status 2p0v = ACE get flag status (TMO Voltage over 2p0v);
           flag status 1p5v = ACE get flag status(TMO Voltage over 1p5v);
           flag_status_1p0v = ACE_get_flag_status(TM0_Voltage_over_1p0v);
           /* Voltage flags are displayed on the LEDs through GPIO */
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 )
```



4. Right-click on Cortex – M3 Flash under Voltage_Monitor_MSS_MSS_CM3_hw_platform and select Build.

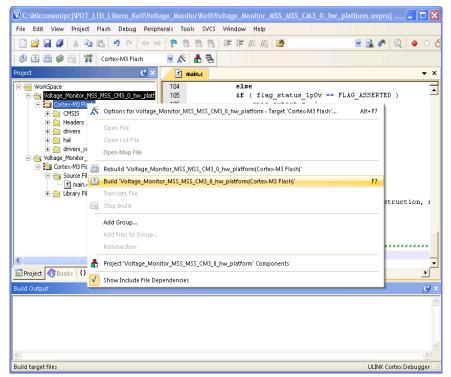


Figure 29 · Building the Hardware Platform



5. Right-click on Voltage_Monitor_MSS_MSS_CM3_app and select Set as Active Project.

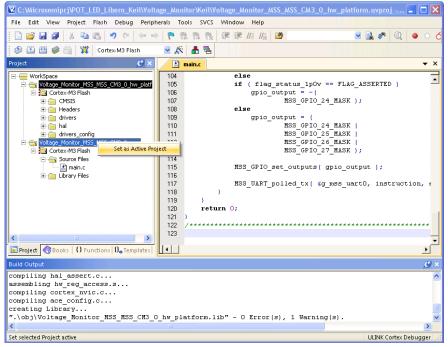


Figure 30 · Setting the Application Project as Active

6. Right-click on Cortex – M3 Flash under Voltage_Monitor_MSS_MSS_CM3_app and click Options for project.

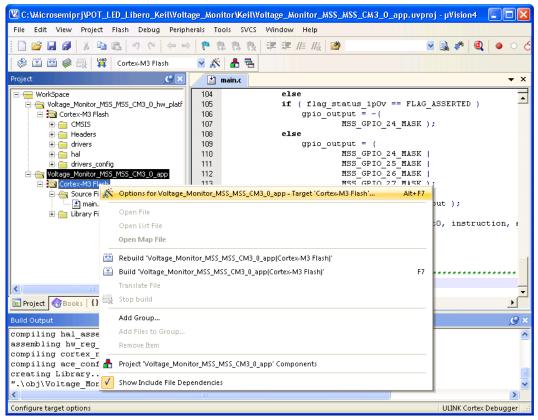


Figure 31 · Target Options



7. Go to the Target tab and change XTAL (MHZ) clock to 80 as shown in Figure 32 · .

🛚 Option	s for Ta	rget 'Cortex-	M3 Flash'							
Device	Target 0	utput Listing	User C/C++	Asm	Linker I	Debug l	Jtilities			
Actel A2F200M3F Code Generation Code Generation Use Cross-Module Optimization Use MicroLIB Big Endian										
	Read/Only Memory Areas default off-chip Start Size Startup default off-chip Start Size Nolnit									
	ROM1:	0x0	0x1000000	۲		BAM1:	0x10000000	0x100000		
	ROM2:			0		RAM2:				
	ROM3:			С		RAM3:				
Г	on-chip IROM1: IROM2:			c c	▼ 	on-chip IRAM1: IRAM2:	0×20000000	0x10000		
		[ОК	Car	ncel	Defa	ults		Help	

Figure 32 · Target Option for Target Cortex-M3 Flash

- 8. Close the options dialog by clicking **OK**.
- Select rebuild all target files from the Project drop-down menu. This action compiles all of the source files and links the object files into an AXF file for debug. Correct any syntax errors and re-build if necessary.

The following messages are displayed in the console:

Build Output		
C:\Microsemiprj\POT_LED_Libero_Keil\Voltage_Monitor\Keil\main.c(65): warning: #167-D: argument of type "uint8_t *" is	incompatible wit	h paramet 木
C:\Microsemiprj\POT_LED_Libero_Keil\Voltage_Monitor\Keil\main.c(70): warning: #167-D: argument of type "uint8_t *" is	incompatible wit	h paramet
C:\Microsemiprj\POT_LED_Libero_Keil\Voltage_Monitor\Keil\main.c(120): warning: #111-D: statement is unreachable		
linking		
Program Size: Code=8110 RO-data=1522 RW-data=164 ZI-data=4988		
".\obj\Voltage_Monitor_MSS_MSS_CM3_0_app.axf" - 0 Error(s), 3 Warning(s).		~
)	>
ULINK Cortex Debugger	CA	p num scril 🤞

Figure 33 · Build Output

- 10. Open the **Target Options** by right-clicking **Cortex M3 Flash** under Voltage_Monitor_MSS_MSS_CM3_app and select Options.
- 11. Click the **Utilities** tab on the **Options for Target** dialog box and clear the "**Update Target before Debugging**" checkbox.

🕱 Options for Target 'Cortex-M3 Flash'
Device Target Output Listing User C/C++ Asm Linker Debug Utilities
Configure Flash Menu Command
C Use Target Driver for Flash Programming
ULINK Cortex Debugger 💌 Settings 🔽 Update Target before Debugging
Init File: Edit
C Use External Tool for Flash Programming
Command:
Arguments:
🗖 Run Independent
OK Cancel Defaults Help

Figure 34 · Clear Update Target Before Debugging

12. Click Settings.



bug Trace Flash Download				
Download Function C Erase Full Chip C Erase Sectors C Do not Erase Programming Algorithm	Verify	BAM for A	Algorithm 0x20000000	Size: 0x0800
Description	Device Type	Device Size	Addre	ss Range
	Add	Start:		Size:

Figure 35 · Cortex – M Target Driver Setup

13. Click Add. The Add Flash Programming algorithm is displayed as shown in Figure 36 · .

A	Add Flash Programming Algorithm 🛛 🔀									
	Description	Deuise Ture	Device Size							
		Device Type								
	A2FxxxM3 256kB Flash	On-chip Flash	256k							
	AM29x128 Flash	Ext. Flash 16-bit	16M							
	ATSAM3N Flash	On-chip Flash	256k	_						
	ATSAM3N GPNVM bits	On-chip Flash	16							
	ATSAM3S Flash	On-chip Flash	256k							
	ATSAM3S GPNVM bits	On-chip Flash	16							
	ATSAM3U Flash	On-chip Flash	128k							
	ATSAM3U Bank 1 Flash	On-chip Flash	128k							
	ATSAM3U GPNVM bits	On-chip Flash	16							
	ATSAM3X 128kB Flash	On-chip Flash	128k							
	ATSAM3X 256kB Flash	On-chip Flash	256k							
	ATSAM3X 512kB Flash	On-chip Flash	512k							
	ATSAM3X GPNVM bits	On-chip Flash	16							
	EFM32 Gecko/Tiny Gecko	On-chip Flash	128k							
	HT32 Series Flash	On-chip Flash	128k							
	HT32 Series Flash Options	On-chip Flash	4k	~						
	Add	Cancel								

Figure 36 · Add Flash Programming Algorithm

- 14. After adding the required algorithm, click **OK** in the Target Driver Setup window.
- 15. Click **OK** to close the Options for Target dialog box (Figure $34 \cdot$).

Step 7 - 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 PC. Perform the following steps to use the SmartFusion Evaluation Kit Board or the SmartFusion Development Kit Board:

- 1. 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 PC. If Windows prompts you to connect to Windows Update, select No, not at this time and click Next.
- 2. If the Silicon Labs CP210x USB to UART Bridge drivers are automatically detected (this can be verified in Device Manager), as shown in Figure 37 · . Proceed to next step, otherwise follow
- 3. Step 8 Installing Drivers for the USB to RS232 Bridge.



A Device Manager
<u>File</u> <u>Action</u> <u>View</u> <u>H</u> elp
⊿ 📇 w7-Donthus
🔈 📲 Computer
Disk drives
Display adapters
DVD/CD-ROM drives
Human Interface Devices
IDE ATA/ATAPI controllers
⊳ · ─── Keyboards
Mice and other pointing devices
D Monitors
Network adapters
Ports (COM & LPT)
Silicon Labs CP210x USB to UART Bridge (COM3)
Processors
Sound, video and game controllers
System devices
Universal Serial Bus controllers

Figure 37 · Device Manager Listing Silicon Labs CP210x USB to UART Bridge Drivers

- 4. From the Windows Start menu, select Programs > Accessories > Communications > HyperTerminal. This opens HyperTerminal. If your PC 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.
- 5. Enter Hyperterminal in the Name field in the Connection Description dialog box and click OK.

Connection Description
New Connection
Enter a name and choose an icon for the connection:
Name:
Hyperterminal
lcon:
OK Cancel

Figure 38 · New Connection

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



Connect To	? 🛛
Nyperterr	ninal
Enter details for t	he phone number that you want to dial:
<u>C</u> ountry/region:	India (91) 💽
Ar <u>e</u> a code:	91
<u>P</u> hone number:	
Co <u>n</u> nect using:	СОМЗ 🗸 🗸
	OK Cancel

Figure 39 · Selecting the COM Port

- 7. 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 Defaults
	K Cancel Apply

Figure 40 · 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 8 - 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:

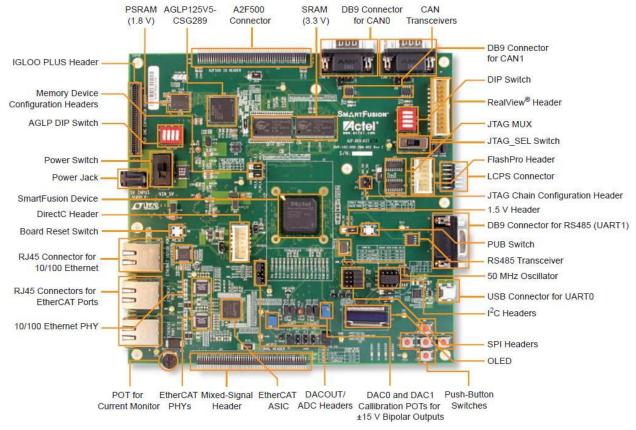
1. Download the USB to RS232 bridge drivers from www.microsemi.com/soc/documents/CP2102_driver.zip.

- 2. Unzip the CP2102_driver.zip file.
- 3. Double-click (Run) the CP210x_VCP_Win_XP_S2K3_Vista_7.exe file.
- 4. Accept the default installation location and click Install.
- 5. Click Continue Anyway if prompted.

6. When the installation is complete, click OK. The Ports (COM & LPT) section of the Device Manager lists Silicon Labs CP210x USB to UART Bridge under the Ports section of Device Manager.

Step 9 - Connecting the ULINK-ME to the Board and PC

This section describes the connection between the SmartFusion Evaluation Kit Board or SmartFusion Development Kit Board, ULINK-ME, and PC. Make sure to use the appropriate setting for the board that is being used.







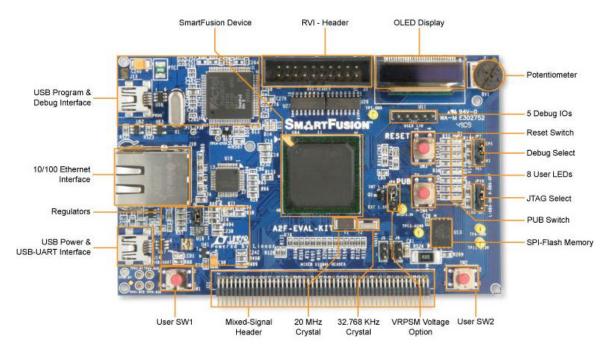


Figure 42 · SmartFusion Evaluation Kit Board





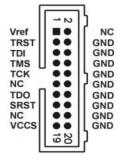


Figure 44 · ULINK-ME Header

Using the SmartFusion Development Kit Board

- 1. Connect a USB A-Mini B cable between your PC and the SmartFusion Development Kit Board (J9). This is used to display the HyperTerminal communications.
- Verify that the ULINK-ME debugger is connected to the SmartFusion Development Kit Board (J3) and also to your PC via USB A-Mini B cable. The ULINK-ME adapter has one LED that indicates connection status in the following ways:
 - Slow flashing indicates that ULINK-ME is ready to communicate with the debugger.
 - Fast flashing indicates that the target board is executing the program under debugger control.



- ON during debugging indicates that the debugger has halted the target board.
- ON during download indicates that target download/verification is in progress.
- 3. Change the switch SW9 to ON position.
- 4. Connect pin 2 and 3 on the jumper JP7 on the SmartFusion Development Kit Board.

Using the SmartFusion Evaluation Kit Board

- 1. Connect a USB A-Mini B cable between your PC and the SmartFusion Evaluation Kit Board (RVI-Header). This is used to display the HyperTerminal communications.
- 2. Verify that the ULINK-ME debugger is connected to the SmartFusion Evaluation Kit Board (J3) and also to your PC via USB A-Mini B cable. The ULINK-ME adapter has one LED that indicates connection status in the following ways:
 - Slow flashing indicates that ULINK-ME is ready to communicate with the debugger.
 - Fast flashing indicates that the target board is executing the program under debugger control.
 - ON during debugging indicates that the debugger has halted the target board.
 - ON during download indicates that target download/verification is in progress.
- 3. Connect pin 2 and 3 on the jumper JP10 on the SmartFusion Evaluation Kit Board.
- 4. Connect pin 1 and 2 on the jumper JP6 on the SmartFusion Evaluation Kit Board.
- 5. Connect pin 2 and 3 on the jumper JP7 for Keil debugging mode.

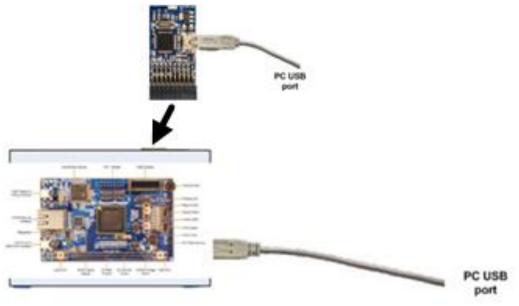


Figure 45 · SmartFusion Evaluation Kit Board, ULINK Debugger, and USB Connections



Step 10 - Debugging the Application Project Through µVision4

 Download the processor code to the SmartFusion eSRAM and execute it via the debug hardware by selecting Start/Stop Debug Session from the Debug drop-down menu. The code will automatically 'run to main' and then stop.

	<u>P</u> roject Fl <u>a</u> sh ※ 학교 대립 / 북 위 (아 (아 국) /		<u>Window</u> <u>H</u> elp	※ ・ 昆	-	•
egisters	<u>@</u> ×	Disassembly	<u>@</u> ×	Symbols		9
Register Core	Value	23: uint8_t	rx_size; rx_buff[1];	Mask: ×	Addr	Type
R1 R2 R3 R4 R5 R6 R7 R8 R9 R10 R11 R12 R13(SP) R14 (LR) R14 (LR) R14 (LR) R12 R13(SP) R14 (LR) R12 R13(SP) R14 (LR) R14 (LR) R15 R16 R17 R17 R18 R17 R18 R18 R19 R19 R19 R19 R19 R19 R19 R19	0x20001 0x00002 0x00002 0x00002 0x00002 0x00000 0x00000 0x00000 0x00000 0x00000 0x00000 0x00000 0x00000 0x00000 0x00000	0x000002&E F04F0234 No main.c 010 * - over 2.5v	int8_t greeting[] = ' V r2,#0x34		nu Ira C d d d d d d	Application Module Module Module Module Module Module Module
Command		,9×	Locals			9
Load "D:\\Deep	thi\\satish	D\\Voltage_Monitor\\Vo.	Name		Value	-
>		2	rx_size t rx_buff		Kout of scope 0x2000141C	[]

Figure 46 · Debug Session

The Registers Window is displayed by default in the Project Window. To modify a register value, select the value and press F2, or double-click the value.

2. Click the **Run** icon from the debug toolbar, or select **Run** from the **Debug** drop-down menu, it should display the following message in HyperTerminal:

```
Welcome to Microsemi's SmartFusion Voltage Monitor
```

Press any Key

- Move your cursor into the HyperTerminal window and press any key on your PC keyboard. The voltage measurement is displayed. In addition, also observe the LEDs on the SmartFusion Evaluation Kit Board. They are illuminated when one of the voltage monitor flags is asserted.
- 4. Adjust the POT while pressing a key and observe that the voltage measurement is continuously updated

🎨 hyperterminal - Hyper Terminal	
Ele Edt Yiew Call Iransfer Help	
Welcome to Microsemi's SmartFusion Voltage Monitor	
Press Any Key 1.608 V	
Press Any Key 2.164 V	
Press Any Key 2.349 V	
Press Any Key 2.486 V	
Press Any Key 2.560 V	
Press Any Key 1.234 V	
Press Any Key	
Connected 0:00:34 Auto detect 57600 8-N-1 SCROLL CAPS NUM Capture Print echo	¥

Figure 47 · HyperTerminal Window

Observe the state of the LEDs as the POT is adjusted. Confirm that the flags work as specified in the ACE configurator. To stop the program, click on the 'Stop' icon (²) on the debug toolbar or **Stop** from **Debug** menu.



Appendix A – Libero SoC Catalog Settings

The following steps show how to configure your vault location and set up the repositories in Libero SoC.

1. On the **Catalog** window, click **Options**.

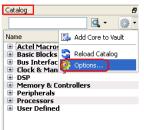
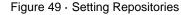


Figure 48 · Catalog – Options

- 2. The **Options** window is displayed. Click **Repositories** under **Vault/Repositories Settings** 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.

Options		? 🔀
Veult/Repositories Settings Repositories View Settings View Settings Filters	www.actel-ip.com/repositories/SgCore www.actel-ip.com/repositories/DirectCore www.actel-ip.com/repositories/Firmware	Add Remove Defaults
Help	ОК	Cancel



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

Options	2 🔀
Vault/Repositories Settings Vault/location View Settings View Settings Display Filters	Current vault location: C:\Documents and Settings\donthus\Application I Browse to location Select new vault location: C:\Documents and Settings\donthus\Application Data\Actel\tools Default Default
Help	OK Cancel

Figure 50 · 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 window.

🔄 Reload Catalog			
🥁 Reload Catalog			
Vault/Repositories Settings	_		Search by all fields (30/30):
🔏 All 🔤 vacin 💽 web repositories	;		
display only the latest version of a core	- -		
Name ^	Version	Size (MB)	Status
CoreAI Driver	3.0.101	0.43	
CoreAhbNvm Driver	2.1.102	0.23	
CoreGPIO Driver	3.0.101	0.55	
CoreI2C Driver	2.0.103	0.43	
CoreInterrupt Driver	2.1.102	0.2	
CorePWM Driver	2.0.102	0.26	
CoreSPI 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	
Fusion-II IAR EWARM flash loader	1.0.101	0.049	
Fusion-II IAR Embedded Workbench device descri	1.0.2	0.022	
Fusion-II SPI Driver	1.0.3	0.11	
Fusion2 In-Application-Programming	1.0.2	0.56	
Hardware Abstraction Layer (HAL)	2.1.102	0.21	
Hardware Abstraction Layer (HAL)	2.1.1 (*)	0.13	2.1.102 is available for download
SmartFusion CMS15-PAL	2.1.100	0.35	
SmartFusion CMSIS-PAL	2.0.100 (*)	0.34	2.1.100 is available for download
5martFusion Chip Boot	1.3.1	0.091	
5martFusion MSS ACE Driver	2.0.102	2.3	
SmartFusion MSS Ethernet MAC Driver	2.0.103	0.28	
5martFusion MSS GPIO Driver	2.0.103	0.75	
5martFusion MSS I2C Driver	2.0.101	0.55	
5martFusion MSS Peripheral DMA Driver	2.0.102	0.7	
SmartFusion MSS SPI Driver	2.0.101	0.77	~

Figure 51 · Firmware Catalog Settings

- 3. Select Repositories under Vault/Repositories Settings in the Options 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**.

If new cores are available for download, click **Download them now!** to download the new cores to the vault.



Appendix – C

Configuring the GPIO Peripheral

1. Double-click the **GPIO** block in the **MSS component**, configure as shown in Figure $52 \cdot$, and click **OK**.

Sconfiguring MSS_GPIO_0 (MSS_GPIO - 1.0.101)	
Configuration	<u> </u>
Multiplexed With I2C1 Peripheral Dedicated I/Os	
GPIO_31: Use as MSS I/O Pad Not Used 🔍 U20 or connect to Fabric Output 💌	
GPIO_30: Use as MSS I/O Pad Not Used 💟 V22 or connect to Fabric Output 💌	
Multiplexed With UART1 Peripheral Dedicated I/Os	
GPIO_29: Use as MSS I/O Pad Not Used 😪 W22 or connect to Fabric Output	
GPIO_28: Use as MSS I/O Pad Not Used 🕑 V20 or connect to Fabric Output 💌	
Multiplexed With SPI1 Peripheral Dedicated 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 Lised 👽 🛛 AA22 or connect to Fabric Not Lised 🔍	×
Help	Cancel

Figure 52 · Configure MSS_GPIO_0

This example requires GPIO_31, GPIO_30, GPIO_29, and GPIO_28 to be connected to LED_4 to LED_1 on the SmartFusion Evaluation Kit Board (D4 to D1 on the SmartFusion Development Kit Board). These signals will be routed through the fabric to I/O pins H17, C19, B20, and B19, respectively.

2. Click File > Save to save the Voltage_Monitor_MSS.

Generating the MSS Component

1. Right-click on Voltage_Monitor_MSS_0 component on the Voltage_Monitor tab and select Update Instance(s) with Latest Component... as shown in Figure 53 · .

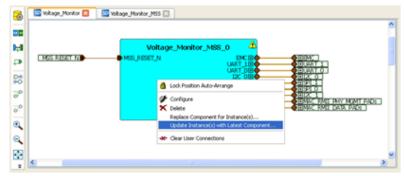


Figure 53 · Updating the MSS



Displaying POT Level with LEDs - Libero SoC and µVision4 IDE Flow Tutorial for SmartFusion cSoC

2. Promote the M2F_GPIO [31:28] pins to top level.

6	💴 Voltage_Monitor* 🔀	SO Voltage_Monitor_MSS 🔀
-)		<u>^</u>
₽₽		Voltage_Monitor_MSS_0
₽	MSS_RESET_N TMD_Voltage	MSS_RESET_N M2F GPO_07
D⊕ ⊕D	VAREFO 🔊	💁 VAREFO 🔰 V12F 🖓 🦛 Disconnect
0 ⁰ 0		
D D		Promote to Top Level Clear Attribute
Ð		Invert
Θ		Tie Low Tie High
<		Mark Unused
×	<	Add Pins to Group

Figure 54 · GPIO Pins Promoted to Top Level

3. Click **Design > Configure Firmware** as shown in Figure $55 \cdot .$

Project File Edit View	Design SmartDesign Help
i 🗅 🚄 🔲 💋 🕤	😔 Configure Firmware
	Generate Programming Data
Design Hierarchy	The Description
-	🗐 Reports

Figure 55 · Opening Design_Firmware

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.

S	tartPag	SD	Voltage_Monitor	IGN_FIRMWARE		
	Generate		Instance Name	Core Type	Version	Compatible Hardware Instance
1		ø 🖣	HAL_0	HAL	2.2.102	Voltage_Monitor_MSS
2		-	MSS_ACE_Driver_0	MSS_ACE_Driver	2.3.105	Voltage_Monitor_MSS:MSS_ACE_0
3		-	MSS_GPIO_Driver_0	MS5_GPIO_Driver	2.0.105	Voltage_Monitor_MSS:MS5_GPIO_0
4		-	MSS_IAP_Driver_0	MSS_IAP_Driver	2.2.101	Voltage_Monitor_MSS
5		-	MSS_NVM_Driver_0	MSS_NVM_Driver	2.2.102	Voltage_Monitor_MSS:MSS_ENVM_0
6		-	MSS_PDMA_Driver_0	MSS_PDMA_Driver	2.0.102	Voltage_Monitor_MSS:MSS_DMA_0
7		-	MSS_RTC_Driver_0	MSS_RTC_Driver	2.0.100	Voltage_Monitor_MSS:MSS_RTC_0
8		-	MSS_Timer_Driver_0	MSS_Timer_Driver	2.1.101	Voltage_Monitor_MSS:MSS_TIMER_0
9		-	MSS_UART_Driver_0	MSS_UART_Driver	2.3.101	Voltage_Monitor_MSS:MSS_UART_0
10		6	SmartFusion_CMSIS_PAL_0	SmartFusion_CMSIS_PAL	2.3.103	Voltage_Monitor_MSS
			Configuring SmartFusion_CMSIS_PAL Configuration Software Tool Chain Kell-MDK Help V OK	Cancel		

Figure 56 · Firmware Configuration Settings – CMSIS Peripheral

- 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.
- 6. Click File > Save to Save the Design_Firmware.

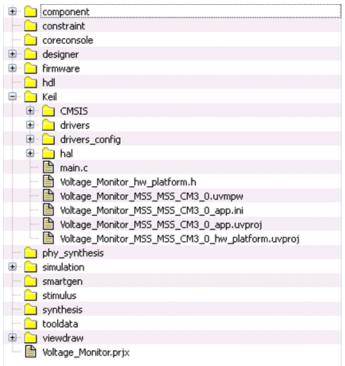


7. Save the design and generate the component by clicking Generate Component or by selecting SmartDesign > Generate Component.

Gen	Voltage_Monitor* 🗙 💿 Voltage_Mon erate Component	itor_MSS 🛛		
₽₽		ſ		
₽		Voltage_Mon	itor_MSS_0	
D® BD	MSS_RESET_N	MSS_RESET_N TM0_Voltage VARE=0	M2F_GPO_31 M2F_GPO_30 M2F_GPO_29	M2F_GPO_31 M2F_GPO_30 M2F_GPO_29
0 ⁰		VARETU	M2F_GPO_29 M2F_GPO_28 UART_0	M2F_GPO_29 M2F_GPO_28 €∎UART_0
Ð			🜮 IP+	
Θ				~
*	<)	>

Figure 57 · Generating the MSS Component

- After successful generation of project, the log window displays the message "Info: 'Voltage_Monitor' was successfully generated. <u>Open datasheet for details</u>". The datasheet has the Project information such as the generated files, used IO's, memory map, etc.
- 9. Confirm that the Keil folder is created with the folders and files as shown in Figure 58 · .





Generating the Program File

Libero SoC provides the push button flow for generating programming data of the project in a single step. By clicking the **Generate Programming Data**, you can complete the synthesis, place and route, verify timing, and generate the programming file. You can also complete the flow by running the synthesis and place and route tools in interactive mode (step-by-step), for more details; refer to the *Libero SoC Quick Start Guide*.

Push-button Design Flow

1. Click Edit I/O Attributes under Constrain Place and Route in the Design Flow window.



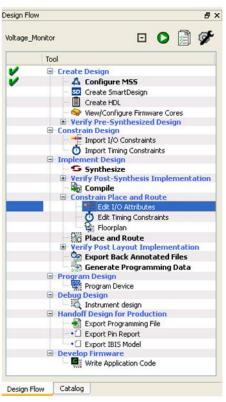


Figure 59 · Configuring SmartFusion_CMSIS_PAL_0

- 2. Make the following pin assignments in **MultiView Navigator** window as shown in Figure $60 \cdot :$
 - GPO_28 to B19
 - GPO_29 to B20
 - GPO_30 to C19
 - GPO_31 to H17

📌 MultiView Navigator [Voltage_	Monitor *] - [I/O Attr	ibute Editor]						
₩E Elle Edit Yew Logic Format]	ools <u>W</u> indow <u>H</u> elp						-	Ξ×
∎© ∰ ⊇ ⊆ ₩ ? ,	0 P P R (?)	-0 •0 🕚	8 to to [₽ @ ₩	• ≻	🕺 📲 🤺	- 1	. ⁺n
	Port Name/	Group	Macro Cell	Pin Number	Locked	Bank Name	1/0 Standard	Out; Drive
⊕ ▲ M2F_GP0_29_pad	1 M2F_GP0_28	1	ADLIB:OUTB	B19	V	Bank0	LVTTL	12
⊕ ▲ ■: M2F_GP0_30_pad ⊕ ▲ ■: M2F_GP0_31_pad	2 M2F_GP0_29	1	ADLIB:OUTB	B20		Bank0	LVTTL	12
⊕- A+: M2F_GP0_31_pad ⊕- A+: Voltage_Monitor_MS	3 M2F_GP0_30	1	ADLIB:OUTB	C19		Bank0	LVTTL	12
	4 M2F_GP0_31		ADLIB:OUTB	H17 ⊻		Bank1	LVTTL	12
	5 MSS_RESET.		ADLIB:INBUF	R1	V	Bank2	LVTTL	
	< ≪ ▼ Ports &	Package Pins	1					2
×	× → Output / R	esults 👌 Find 1	1					
Ready			row 4, col 4	FAM: SmartFusio	on DIE: A2	F200M3F PAC	KAGE: 484 FB	IGA

Figure 60 · MultiView Navigator GUI



- 3. Commit and check the edits using **File > Commit and Check**. Correct any errors that are reported in the MVN log window.
- 4. Close the MultiView Navigator using File > Exit.
- 5. Close the Designer window and select Yes when it prompts to save changes.

🏕 Designer - [Voltage_Monitor*]	_ 🗆 🗙
B File View Tools Options Help	_ 8 ×
□ ☞ ■ ? 荷荷 ♀ ━ ※ ☆ ኵ 静 橋 ♀ ≫ り /	
	^
Design Flow	
Compile	
Designer 🔀 hPro Data File	
MultiView Ni Save changes to Volkage_Monitor? TTime	
	mart ower
	~
Version: 10.0.9.22	^
Release: 10.1	
Info: The design Voltage_Monitor.adb was last modified by software ve	rsion
Opened an existing Libero design Voltage_Monitor.adb.	
The Execute Script command succeeded (00:00:45)	
	~
<	>
Ready FAM: SmartFusion DIE: A2F200M3F PKG: 484 FBG	A

Figure 61 · Designer Window

6. Click **Generate Programming Data** to complete the place and route, verify timing and generate the programming file. This completes the fdb file generation.

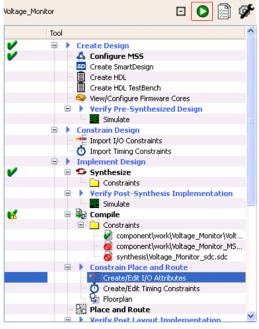


Figure 62 · Generating Programming Data

The **Design Flow** window looks as shown in Figure $63 \cdot .$



Displaying POT Level with LEDs - Libero SoC and μ Vision4 IDE Flow Tutorial for SmartFusion cSoC

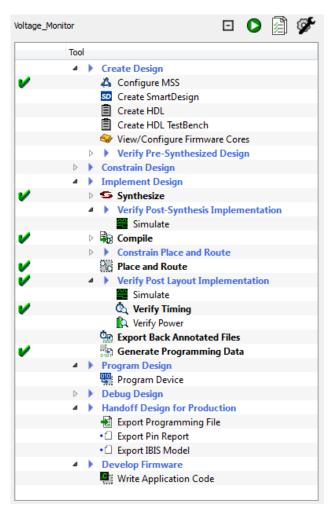


Figure 63 \cdot Design Flow Window After Building the Project

7. Follow Step 5 - Programming SmartFusion Board Using FlashPro.



List of Changes

Revision	Changes	Page
Revision 3	Replaced Figure 6 · (SAR 38348)	8
(May 2012)	Replaced Figure 21 · (SAR 38348)	15
	Replaced Figure 25 · (SAR 38348)	17
	Replaced Figure 26 · (SAR 38348)	19
	Replaced Figure 27 · (SAR 38348)	20
	Modified Step 6 - Building the Software Application through Keil $\mu\text{Vision}^{\$}4$ IDE (SAR 38348)	20
	Modified Appendix – C (SAR 38348)	37
	Replaced Figure 56 · (SAR 38348)	38
	Replaced Figure 58 · (SAR 38348)	39
	Replaced Figure 62 · (SAR 38348)	41
	Replaced Figure 63 · (SAR 38348)	42
Revision 2 (February 2012)	Modified Associated Project Files section (SAR 36902).	3
	Modified Step 2 - Configuring MSS Peripherals section (SAR 36902).	8
	Modified Step 7 - Configuring the Serial Terminal Emulation Program section (SAR 36902).	28
	Updated Figure 38 (SAR 36902).	29
Revision 1 (November 2011)	Updated the document for Libero SoC v10.0 (SAR 35043).	NA
	Corrected the signal name from TM0_Voltage to TM0_voltage in point 10 listed below Figure 12 · (SAR 30307)	12

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.



Displaying POT Level with LEDs - Libero SoC and µVision4 IDE Flow Tutorial for SmartFusion cSoC

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.