

32

# **RX66T** Group

Renesas Starter Kit Smart Configurator Tutorial Manual For CS+

RENESAS 32-Bit MCU RX Family / RX600 Series

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics

Rev. 1.00 Sep 2018

#### Notice

- Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
- Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
- 3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
- 4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
- 5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

- 6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
- 7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
- 8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
- 9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
- 10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
- 11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
- 12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.
- (Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
- (Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

#### General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.

- <sup>3</sup>⁄<sub>4</sub> The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.
- 2. Processing at Power-on

The state of the product is undefined at the moment when power is supplied.

- <sup>3</sup>⁄<sub>4</sub> The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.
- 3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

- <sup>3</sup>⁄<sub>4</sub> The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
- 4. Clock Signals

After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

- When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.
- 5. Differences between Products

Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

<sup>3</sup>⁄<sub>4</sub> The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

### Disclaimer

By using this Renesas Starter Kit (RSK), the user accepts the following terms:

The RSK is not guaranteed to be error free, and the entire risk as to the results and performance of the RSK is assumed by the User. The RSK is provided by Renesas on an "as is" basis without warranty of any kind whether express or implied, including but not limited to the implied warranties of satisfactory quality, fitness for a particular purpose, title and non-infringement of intellectual property rights with regard to the RSK. Renesas expressly disclaims all such warranties. Renesas or its affiliates shall in no event be liable for any loss of profit, loss of data, loss of contract, loss of business, damage to reputation or goodwill, any economic loss, any reprogramming or recall costs (whether the foregoing losses are direct or indirect) nor shall Renesas or its affiliates be liable for any other direct or indirect special, incidental or consequential damages arising out of or in relation to the use of this RSK, even if Renesas or its affiliates have been advised of the possibility of such damages.

### Precautions

The following precautions should be observed when operating any RSK product:

This Renesas Starter Kit is only intended for use in a laboratory environment under ambient temperature and humidity conditions. A safe separation distance should be used between this and any sensitive equipment. Its use outside the laboratory, classroom, study area or similar such area invalidates conformity with the protection requirements of the Electromagnetic Compatibility Directive and could lead to prosecution.

The product generates, uses, and can radiate radio frequency energy and may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment causes harmful interference to radio or television reception, which can be determined by turning the equipment off or on, you are encouraged to try to correct the interference by one or more of the following measures;

- ensure attached cables do not lie across the equipment
- reorient the receiving antenna
- increase the distance between the equipment and the receiver
- connect the equipment into an outlet on a circuit different from that which the receiver is connected
- power down the equipment when not in use
- consult the dealer or an experienced radio/TV technician for help NOTE: It is recommended that wherever possible shielded interface cables are used.

The product is potentially susceptible to certain EMC phenomena. To mitigate against them it is recommended that the following measures be undertaken;

- The user is advised that mobile phones should not be used within 10m of the product when in use.
- The user is advised to take ESD precautions when handling the equipment.

The Renesas Starter Kit does not represent an ideal reference design for an end product and does not fulfil the regulatory standards for an end product.

## How to Use This Manual

#### 1. Purpose and Target Readers

This manual is designed to provide the user with an understanding of how to use Application Leading Tool (Smart Configurator) for RX together with the CS+ IDE to create a working project for the RSK platform. It is intended for users designing sample code on the RSK platform, using the many different incorporated peripheral devices.

The manual comprises of step-by-step instructions to generate code and import it into CS+, but does not intend to be a complete guide to software development on the RSK platform. Further details regarding operating the RX66T microcontroller may be found in the Hardware Manual and within the provided sample code.

Particular attention should be paid to the precautionary notes when using the manual. These notes occur within the body of the text, at the end of each section, and in the Usage Notes section.

In this manual, the display may differ slightly from screen shots. There is no problem in reading this manual.

The revision history summarizes the locations of revisions and additions. It does not list all revisions. Refer to the text of the manual for details.

The following documents apply to the RX66T Group. Make sure to refer to the latest versions of these documents. The newest versions of the documents listed may be obtained from the Renesas Electronics Web site.

Document Type	Description	Document Title	Document No.
User's Manual	Describes the technical details of the RSK hardware.	RSKRX66T User's Manual	R20UT4150EG
Tutorial Manual	Provides a guide to setting up RSK environment, running sample code and debugging programs.	RSKRX66T Tutorial Manual	R20UT4151EG
Quick Start Guide	Provides simple instructions to setup the RSK and run the first sample.	RSKRX66T Quick Start Guide	R20UT4152EG
Smart Configurator Tutorial Manual	Provides a guide to code generation and importing into the CS+ IDE.	RSKRX66T Smart Configurator Tutorial Manual	R20UT4153EG
Schematics Full detail circuit schematics of the RSK.		RSKRX66T Schematics	R20UT4149EG
Hardware Manual	Provides technical details of the RX66T microcontroller.	RX66T Group Hardware Manual	R01UH0749EJ

## 2. List of Abbreviations and Acronyms

Abbreviation	Full Form
ADC	Analog-to-Digital Converter
API	Application Programming Interface
bps	bits per second
CMT	Compare Match Timer
COM	COMmunications port referring to PC serial port
CPU	Central Processing Unit
E1/E2 Lite	Renesas On-chip Debugging Emulator
GUI	Graphical User Interface
IDE	Integrated Development Environment
IRQ	Interrupt Request
LCD	Liquid Crystal Display
LED	Light Emitting Diode
LSB	Least Significant Bit
LVD	Low Voltage Detect
MCU	Micro-controller Unit
MSB	Most Significant Bit
PC	Personal Computer
PLL	Phase-locked Loop
Pmod™	This is a Digilent Pmod <sup>™</sup> Compatible connector. Pmod <sup>™</sup> is registered to <u>Digilent Inc.</u> Digilent-Pmod_Interface_Specification
PSU	Power Supply Unit
RAM	Random Access Memory
ROM	Read Only Memory
RSK	Renesas Starter Kit
RTC	Real Time Clock
SAU	Serial Array Unit
SCI	Serial Communications Interface
SPI	Serial Peripheral Interface
TAU	Timer Array Unit
TPU	Timer Pulse Unit
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
WDT	Watchdog Timer

All trademarks and registered trademarks are the property of their respective owners.

## Table of Contents

1. Overview	
1.1 Purpose	
1.2 Features	
2. Introduction	9
3. Project Creation with CS+	
	10
3.2 Creating the Project	
4. Smart Configurator Using the CS+	
4.1 Introduction	
4.2 Project Configuration using Smart Configurator - Ov	erview page13
4.3 Board configuration page	
	14
4.4 Clocks configuration page	15
	15
4.5 Component page	16
4.5.1 Add a software component into the project	16
4.5.2 Compare Match Timer	17
4.5.3 Interrupt Controller	
4.5.4 Ports	
4.5.5 SCI/SCIF Asynchronous Mode	
4.5.6 SPI Clock Synchronous Mode	
4.6 Pins configuration page	
5 1 5	ent35
5. Completing the Tutorial Project	39
5.2 Additional Folders	
5.3 LCD Code Integration	
•	
5.3.2 CMT Code	
5.4 Switch Code Integration	
5.4.1 Interrupt Code	
5.4.2 De-bounce Timer Code	
	51
5.5 Debug Code Integration	
5.6 UART Code Integration	
-	
5.7 LED Code Integration	
6. Debugging the Project	63
7 Dunning the Smort Configurator Tytorial	<u></u>
<ul><li>7. Running the Smart Configurator Tutorial</li><li>7.1 Running the Tutorial</li></ul>	
8. Additional Information	65

# RENESAS

## RSKRX66T

RENESAS STARTER KIT

#### 1.1 Purpose

This RSK is an evaluation tool for Renesas microcontrollers. This manual describes how to use the CS+ IDE Smart Configurator to create a working project for the RSK platform.

#### 1.2 Features

This RSK provides an evaluation of the following features:

- Project Creation with CS+
- Code generation using the Smart Configurator.
- User circuitry such as switches, LEDs and a potentiometer

The RSK board contains all the circuitry required for microcontroller operation.



## 2. Introduction

This manual is designed to answer, in tutorial form, how to use the Smart Configurator for the RX family together with the CS+ IDE to create a working project for the RSK platform. The tutorials help explain the following:

- Project generation using the CS+
- Detailed use of the Smart Configurator for CS+
- Integration with custom code
- Building the project CS+

The project generator will create a tutorial project with three selectable build configurations:

- 'DefaultBuild' is a project with debug support and optimisation level set to two.
- 'Debug' is a project built with the debugger support included. Optimisation is set to zero.
- 'Release' is a project with optimised compile options (level two) and no 'Outputs debugging information' options not selected, producing code suitable for release in a product.

The tutorial examples in this manual assume that installation procedures described in the RSK Quick Start Guide have been completed. Please refer to the Quick Start Guide for details of preparing the configuration.

These tutorials are designed to show you how to use the RSK and are not intended as a comprehensive introduction to the CS+ debugger, compiler toolchains or the E2 emulator Lite. Please refer to the relevant user manuals for more indepth information.



## 3. Project Creation with CS+

#### 3.1 Introduction

In this section, the user will be guided through the steps required to create a new C project for the RX66T MCU, ready to generate peripheral driver code using Smart Configurator. This project generation step is necessary to create the MCU-specific project and debug files.

#### 3.2 Creating the Project

To use the program, start CS+:

Windows<sup>™</sup> 7: Start Menu > All Programs > Renesas Electronics CS+ > CS+ for CC (RL78,RX,RH850)

Windows<sup>™</sup> 8.1 & 8: From Apps View , click 'CS+ for CC (RL78,RX,RH850)' icon Windows<sup>™</sup> 10: Start Menu > All Apps > Renesas Electronics CS+ > CS+ for CC (RL78,RX,RH850)

• CS+ will show the Start Page. Use the 'GO' button to Create a New Project.	Create New Project A new project can be created. A new project can also be created by reusing the file configuration registered to an existing project.
<ul> <li>In the 'Create Project' dialog, select 'RX' from the 'Microcontroller' pull- down.</li> <li>In the 'Using Microcontroller' list control, scroll down to 'RX66T' and expand the tree control by clicking '+'. Select 'R5F566TEAxFP(100pin)'.</li> <li>Ensure that in the 'Kind of project' pull- down, 'Application(CC-RX)' is selected.</li> </ul>	Create Project × Microcontroller: Using microcontroller: Using microcontroller) Ugdate Product Name: RSF566TEAxFP(100pin) RSF566TEAxFP(100pin) RSF566TEAxFN(84pin) RSF566TEAxFN(84pin) RSF566TEAxFN(80pin) RSF566TEAxFN(80pin) RSF566TEAxFP(100pin) RSF566TEAxFP(100pin) RSF566TEAxFP(100pin) RSF566TEAxFP(100pin) RSF566TEAxFP(100pin) RSF566TEAxFP(100pin) RSF566TEAxFP(100pin)
<ul> <li>Choose an appropriate name and location for the project, then click 'Create'. Note: this tutorial assumes the project is named and located at the place shown opposite.</li> <li>If the folder entered cannot be found a 'Question' dialog will be displayed; click 'Yes'.</li> </ul>	Kind of project:       Application(CC-RX)         Wind of project:       Application(CC-RX)         Project game:       SC_Tutorial         Place:       C:Workspace         Make the project folder         C:Workspace\SC_Tutorial.mtpi         Pags the file composition of an existing project to the new project         Project to be passed:       (Input project file to be diverted.)         Project to be passed:       Cimput project folder to a new project folder.



#### RSKRX66T

#### 3. Project Creation with CS+





## 4. Smart Configurator Using the CS+

#### 4.1 Introduction

The Smart Configurator for the RX66T has been used to generate the sample code discussed in this document. Smart Configurator for CS+ is a tool for generating template 'C' source code and project settings for the RX66T. When using Smart Configurator, it supports user with a visual way of configuring the target device, clocks, software components, hardware resources and interrupts for the project. Thereby bypassing the need in most cases to refer to sections of the Hardware Manual.

By following the steps detailed in this tutorial, the user will generate a CS+ project called SC\_Tutorial. A fully completed Tutorial project is contained on the RSK Web Installer

(<u>https://www.renesas.com/rskrx66t/install/cs</u>) and may be imported into CS+ by following the steps in the Quick Start Guide. This tutorial is intended as a learning exercise for users who wish to use the Smart Configurator to generate their own custom projects for CS+.

Once the user has configured the project, the 'Generate Code' function is used to generate three code modules for each specific MCU feature selected. These code modules are name 'Config\_xxx.h', 'Config\_xxx.c', and 'Config\_xxx\_user.c', where 'xxx' is an acronym for the relevant MCU feature, for example 'S12AD'. Within these code modules, the user is then free to add custom code to meet their specific requirement. Custom code should be added, whenever possible, in between the following comment delimiters:

/\* Start user code for adding. Do not edit comment generated here \*/
/\* End user code. Do not edit comment generated here \*/

Smart Configurator will locate these comment delimiters, and preserve any custom code inside the delimiters on subsequent code generation operations. This is useful if, after adding custom code, the user needs to revisit Smart Configurator to change any MCU operating parameters.

The SC\_Tutorial project uses interrupts for switch inputs, the ADC module, the Compare Match Timer (CMT), the Serial Communications Interface (SCI) and uses these modules to perform A/D conversion and display the results via the Virtual COM port to a terminal program and also on the LCD display on the RSK.

Following a tour of the key user interface features of Smart Configurator and the reader is guided through each of the peripheral function configuration dialogs in §4.2. In §5, the reader is familiarised with the structure of the template code, as well as how to add their own code to the user code areas provided by the code generator.



## 4.2 Project Configuration using Smart Configurator - Overview page

In this section, a brief tour of Smart Configurator is presented. For further details of the Smart Configurator paradigm and reference, refer to the Smart Configurator Usage. You can download the latest document from: https://www.renesas.com/smart-configurator.

Smart Configurator will start up by double clicking on "Smart Configurator(Design Tool)" on the project tree. The Smart Configurator initial view is displayed as illustrated in **Figure 4-1**.

💰 Smart Configurator					-	o x	
File Window Help							
						🖻 물	<b>R</b>
Image: SC_Tutorial.scfg ⋈			- 0	🛃 MCU Package 🛛		- [	э
Overview information			🔁 🖆	🔚 🔺 🔎 👂 🗎 (	Type pin function		
← General Information			?				
This editor allows you to modify the settings stored in configuration file (.scfg)							
Board				2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	20 20 20 20 20 20 20 20 20 20 20 20 20 2		
Allow board and device selection					4 3 2 2 9 8 8 5 5 5 8 5 3 5 5 9 8 10 10 10 10 10 10 10 10 10 10 10 10 10		
	Application under	1			4 10 7 10		
Clocks	development			000 PJ 000 PJ 000 PJ	4 174 214		
Allow clock configuration		- Components		200 24 25 254 25	204 204 2010		
	Middleware			244 2	41 212 42 211		
	Device RTOS				NESAS		
Allow software component selection and configuration	unver 1			200 20 200 20 200 20	RX66T * ***		
		← Pins		<u></u>	IF566TEAxFP		
Pins							
Allow general pin configuration and pin configuration for selected software component					20 VCC 76 764		
Interrupt					2º 34 14 192		
Allow general interrupt configuration and interrupt configuration for selected software component					* * * * * * * * * * * * * * * * * * * *		
Allow general interrupt configuration and interrupt configuration for selected software component							
✓ Current Configuration							
Selected board/device: R5F566TEAxFP (ROM size: 512Kbytes, RAM size: 64Kbytes, Pin count: 100)			~				
Overview Board Clocks Components Pins Interrupts				▶ Legend			
🕒 Console 🛛 🗎 🔐 📑	📃 v 📑 v 🗖 🗖	🔝 Configuration Problems 🛛			-+	~ - [	a
Smart Configurator Output		0 items					
M05000001: Pin 13 is assigned to EXTAL	~	Description		Туре			
M05000001: Pin 11 is assigned to XTAL							
	~						
<	>						
		-					

Figure 4-1 Overview page

Smart Configurator provides GUI features for configuration of MCU sub systems. Once the user has configured all required MCU sub systems and peripherals, the user can click the 'Generate Code' button, resulting in a fully configured CS+ project that builds and runs without error.



#### 4.3 Board configuration page

On the board setting page, set the board type and device type.

Click the board tab and it will be displayed as shown in <b>Figure 4-2</b> .	
∰ SC_Tutorial.scfg ⊠	
Device selection	🖲 🖨
Device selection	Ъď
Board: Custom User Board	
Overview Board Clocks Components Pins Interrupts	

Figure 4-2 Board configuration page

#### 4.3.1 Board configuration

From the default 'Custom User Board', select 'RSKRX66T' from the pull down and change it.

虈 *SC_Tuto	rial.scfg 🔀				
Device selection					
Device se	lection				
Board:	Custom User Board 🗸 🗸 🗸				
Device:	RSKRX64M RSKRX65N-2MB RSKRX65N				
	RSKRX66T Custom User Board				

Figure 4-3 Select board

As shown in **Figure 4-4**, when the confirmation dialog of board change is displayed, please click the 'Continue' button and continue the subsequent procedure.



Figure 4-4 Confirm board change

#### 4.4 Clocks configuration page

Clocks configuration page configures clocks of the device selected. Clock source, frequency, PLL settings and clock divider settings can be configured for the output clocks. Clock configurations will be reflected to 'r\_bsp\_config.h' file in 'Smart Configurator\r\_config' of project tree.

#### 4.4.1 Clocks configuration

**Figure 4-5** shows a screenshot of Smart Configurator with the Clocks configurations. Click on the 'Clocks' tab. Configure the system clocks as shown in the figure. In this tutorial, we are using the on-board 24 MHz crystal resonator for our main clock oscillation source and the PLL circuit is in operation. The PLL output is used as the main system clock and the divisors should be set as shown in **Figure 4-5**.

Set VCC and AVCC to 3.3(V). Then, do not use Negative Voltage Input Settings of Analog Voltage Settings uncheck the check box.



Figure 4-5 Clocks Configuration page



#### 4.5 Component page

Drivers and middleware are handled as software components in Smart Configurator. Component page allows user to select and configure software components.

omponents	.ª₂ 🖻 ⊞ 📫
	😜 🖬
type filter text	
<ul> <li>✓</li></ul>	
Drivers	
Middleware Application	

Figure 4-6 Component page

#### 4.5.1 Add a software component into the project

Smart Configurator supports four types of software components: Startup, Drivers, Middleware and Application. In the following sub-sections, the reader is guided through the steps to configure the MCU for a simple project containing interrupts for switch inputs, timers, ADC and a SCI by component of Drivers.

lick 'Add component' ` icon.	∰ *SC_Tutorial.scfg ⊠	
	Software component	t configuration
	Components	
		<b>1</b>
	type filter text	
	🗸 🗁 Startup	
	V 🗁 Generic	
	iguro 4 7 Add a So	oftware component (1

In 'Software Component Selection' dialog -> Type, select 'Drivers'.

🔥 New Co		×			
Software	Component Selection				
Select component from those available in list					
Туре	Drivers			$\sim$	
Function	All Startup				
Filter	Drivers				
	Middleware Application				
Compor	ante Tuna	Vertion		-	

Figure 4-8 Add a Software component (2)



#### 4.5.2 **Compare Match Timer**

CMT0 will be used as an interval timer for generation of accurate delays. CMT1 and CMT2 will be used as timers in de-bouncing of switch interrupts. Select 'Compare Match Timer' as shown in **Figure 4-9** below then click 'Next'.

Type D	ivers			
Function Al				
Filter				
Componen	^ ts	Туре	Version	
Compar		Code Generator	13.0	
E Compar	e Match Timer	Code Generator	1.5.0	
Comple	mentary PWM Mode II	Code Generator	1.3.0	
	ous Scan Mode S12AD	Code Generator	1.3.0	
CRC Cal	culator	Code Generator	1.3.0	
<				>
Description This softwa	y latest version re component provides co V and can generate interru		32-bit timer with me	odule
Download m	ore software components	5		
	eneral settings			

Figure 4-9 Select Compare Match Timer

In 'Add new configuration for selected component' dialog -> Resource, select 'CMT0' as shown in Figure 4-10 below then click 'Finish'.

🐼 New Component – 🗆 🗙				
Add new configuration f	or selected component			
Compare Match Timer				
Configuration name:	Config_CMT0			
Resource:	СМТО ~			
	CMT0			
	CM11 CMT2			
	CMT3			
<	Back Next > Finish Cancel			
Figure	4-10 Select Resource - CMT0			



In the 'Config\_CMT0' configures CMT0 as shown in **Figure 4-11**. This timer is configured to generate a high priority interrupt every 1ms. We will use this interrupt later in the tutorial to provide an API for generating high accuracy delays required in our application.

Software component configuration       Image: Configure description         Components       Image: Configure description         type filter text       Count clock setting         Image: Compare match setting       O PCLK/32       O PCLK/128       O PCLK/512	ster "SC_Tutorial.scrg 🗠		
type filter text     Count clock setting       • PCLK/8     PCLK/128       • Compare match setting	Software component configurati	on	🔞 🕒
	type filter text	Count clock setting	

Figure 4-11 Config\_CMT0 setting

Click 'Add component' **t** icon. In 'Software Component Selection' dialog -> Type, select 'Drivers'. Select 'Compare Match Timer' then click 'Next'. In 'Add new configuration for selected component' dialog -> Resource, select 'CMT1' as shown in **Figure 4-12** below then click 'Finish'.

孩 New Component		—	
Add new configuration	for selected component		
Compare Match Timer			
Configuration name:	Config_CMT1		
Resource:	CMT1		~
	СМТО		
	CMT1		
	CM12		
	CMT3		
?	< Back Next > Finish		Cancel

Figure 4-12 Select Resource – CMT1

Navigate to the 'Config\_CMT1' and configure CMT1 as shown in **Figure 4-13**. This timer is configured to generate a high priority interrupt after 20ms. This timer is used as our short switch de-bounce timer later in this tutorial.

🕸 *SC_Tutorial.scfg 🛛		
Software component configuration	on	🐻 🖆
$\begin{array}{c} \text{Components} \\ \downarrow^a_Z \ \boxdot \ \textcircled{=} \ \textcircled{=} \ \overleftrightarrow{=} \ \checkmark \end{array}$	Configure	
type filter text	Count clock setting PCLK/8 PCLK/32 PCLK/128 PCLK/512 Compare match setting Interval value 20 mms (Actual value: 2) Register value (CMCOR) 24999 Compare match interrupt (CMI1) Priority Level 10 v	))

Figure 4-13 Config\_CMT1 setting

Click 'Add component' icon. In 'Software Component Selection' dialog -> Type, select 'Drivers'. Select 'Compare Match Timer' then click 'Next'. In 'Add new configuration for selected component' dialog -> Resource, select 'CMT2' as shown in **Figure 4-14** below then click 'Finish'.

孩 New Component			
Add new configuration	for selected component		
Compare Match Timer			
Configuration name:	Config_CMT2		
Resource:	CMT2		~
	CMT0 CMT1		
	CMT2 CMT3		
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> ini	sh	Cancel

Figure 4-14 Select Resource – CMT2

Navigate to the 'Config\_CMT2' and configure CMT2 as shown in **Figure 4-15**. This timer is configured to generate a high priority interrupt after 200ms. This timer is used as our long switch de-bounce timer later in this tutorial.

Interview SC_Tutorial.scfg ⋈			
Software component configuration	n		🖲 🖨
type filter text	Configure Count clock setting PCLK/8 PCLK/32 Compare match setting Interval value Register value (CMCOR) Compare match interrupt (CMI1) Priority	○ PCLK/128  ● PCLK/512       200     ms       15624     ✓	✓ (Actual value: 200)
Config_CMT2			

Figure 4-15 Config\_CMT2 setting



#### 4.5.3 Interrupt Controller

Referring to the RSK schematic, SW1 is connected to IRQ0(P10) and SW2 is connected to IRQ9 (PB3). SW3 is connected IRQ7(P20) and the ADTRG0n. Tutorial used ADTRG0n and will be configured later in §4.5.7.

Click 'Add component' 눟 icon.

In 'Software Component Selection' dialog -> Type, select 'Drivers'. Select 'Interrupt Controller' as shown in **Figure 4-16** then click 'Next'.

诸 New C	Component		— C	x נ
	Component Selection mponent from those availab	le in list		#
Туре	Drivers			~
Function	All			$\sim$
Filter				
Compor	nents Nave Mode	Type Code Generator	Version	^
2000	rupt Controller		1.5.0	
H Norr	Power Consumption nal Mode Timer :e Counting Mode Timer	Code Generator Code Generator Code Generator	1.5.0 1.3.0 1.5.0	
Show Description	only latest version			>
Interrup	t Controller configures the in t, NMI pin interrupt and IRQ		ed by ICU: Software	
Downloa	d more software component	<u>is</u>		
<u>Configure</u>	e general settings			
?	< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish C	ancel

Figure 4-16 Select Interrupt Controller

In 'Add new configuration for selected component' dialog -> Resource, select 'ICU' as shown in **Figure 4-17** below then click 'Finish'.

孩 New Component		_		×
Add new configuration	for selected component			Ļ.
Interrupt Controller				
Configuration name:	Config_ICU			
Resource:	ICU			~
?	Back <u>N</u> ext >	<u>F</u> inish	Cancel	

Figure 4-17 Select resource – ICU



Navigate to the 'Config\_ICU', configure these two interrupts as falling edge triggered as shown in **Figure 4-18** below.

Spin-transporting         Spin-transporting           Spin-transportransporting         Spin-transporting	Components 🎝 🖓 🖃 🕀 🎝	- Configure						
image:			ing					
yryentered      Statuy     Statu	16 1							
Printly       Werk 13 Singlet()         Printly       Werk 13 Singlet()         Printly       Werk 13 Singlet()         Printly       Detector type          Print								
Charge     Ch				Level 15 (highest)	$\sim$			
IMM piniterry Detection by Talling edge Department of the file 0 MH2 IMD setting Department of the file 0 MH2 <		NIMI all intermediated						
Consigned of the second system       Defection system<			-	17 UK 1			0	
MOS setting       Detection type       Exting edge       Deptifier       Deptifier       Motified         MCI setting       Detection type       Exting edge       Deptifier       Deptifier       Motified       Deptifier         MCI setting       Detection type       Exting edge       Deptifier       Motified       Deptifier       Motifier       Motifier       Motifier       Motifier       Motifier       Motifier       Motifier       Motifier       Motifier <td></td> <td></td> <td>Detection type</td> <td>Failing edge</td> <td>V Digi</td> <td>tai filter</td> <td>(IVIHz)</td> <td></td>			Detection type	Failing edge	V Digi	tai filter	(IVIHz)	
Config.CAT1       Priori       Level 13 Stightett       Dignal file:       Mailtaine         MO1 setting       Detection type       Son tend       Dignal file:       Mailtaine       Dignal file:       Ma		IRQ0 setting						
Cereig_CMI2 Priority Level 35 highered Priority		RQ0	Detection type	Falling edge	∼ Digi	tal filter No filter 🗸	• 0 (MHz)	
Import       Detection type       Designed there       Deplot there       Defection         Import       Exect 15 Singlenet       Deplot there       Deplot there<			Priority	Level 15 (highest)	$\sim$			
Impo         Detection type         Deplatifier         0         0.0440           Printy         Exection System         Digital filter         0         0.0440           Impo         Exection System         Digital filter         0		IPO1 setting						
Incy setting       Incy setting       Detection type       Setting       Digital filter       No filter       N			Detection type	Low level	Diai	tal filter No filter	(MHz)	
R02 setting       Detection type       R04 well setting       Digital filter       Mod filter <t< td=""><td></td><td></td><td></td><td></td><td>Digi</td><td>No miler</td><td>(11112)</td><td></td></t<>					Digi	No miler	(11112)	
Image:       Detection type       Developed affice       Notifier       0       0.4400         Image:       Detection type       Developed       Digital filter       Notifier       0       0.4400         Image:       Detection type       Developed       Digital filter       Notifier       0       0.4400         Image:       Detection type       Developed       Digital filter       Notifier       0       0.4400         Image:       Detection type       Detection type       Detection type       Digital filter       Notifier       0       0.4400         Image:       Detection type       Detection type       Detection type       Digital filter       Notifier       0       0.4400         Image:       Detection type       Detection type       Digital filter       Notifier       0       0.4400         Image:       Detection type       Detection type       Digital filter       Notifier       0       0.4400         Image:       Detection type       Detection type       Digital filter       Notifier       0       0.4400         Image:       Detection type       Detection type       Digital filter       Notifier       0       0.4400         Image:       Detection type       Develo			Priority	Level 15 (highest)	$\sim$			
Image       Protony       Level 15 (highpect)         Image       Detection type       Level 16 (highpect)         Image       Detection type       Level 15 (highpect)       Digital filter       Maritime       0       (M440)         Image       Detection type       Level 15 (highpect)       Digital filter       Maritime       0       (M440)         Image       Detection type       Level 15 (highpect)       Digital filter       Maritime       0       (M440)         Image       Detection type       Level 16 (highpect)       Digital filter       Maritime       0       (M440)         Image       Detection type       Level 16 (highpect)       Digital filter       Maritime       0       (M440)         Image       Detection type       Level 16 (highpect)       Digital filter       Maritime       0       (M440)         Image       Detection type       Level 16 (highpect)       Digital filter       Maritime       0       (M440)         Image       Detection type       Level 16 (highpect)       Digital filter		IRQ2 setting						
RQ3 setting       Detection type (sew/level)       Digital filter       Notifier       0.0450         RQ4 setting       Detection type (sew level)       Digital filter       Notifier       0       0.450         RQ5 setting       Detection type (sew level)       Digital filter       Notifier       0       0.450         RQ5 setting       Detection type (sew level)       Digital filter       Notifier       0       0.450         RQ6 setting       Detection type (sew level)       Digital filter       Notifier       0       0.450         RQ6 setting       Detection type (sew level)       Digital filter       Notifier       0       0.450         RQ6 setting       Detection type (sew level)       Digital filter       Notifier       0       0.450         RQ7 setting       Detection type (sew level)       Digital filter       Notifier       0       0.450         RQ8 setting       Detection type (sew level)       Digital filter       Notifier       0       0.450         RQ8 setting       Detection type (sew level)       Digital filter       Notifier       0       0.450         RQ1 setting       Detection type (sew level)       Digital filter       Notifier       0       0.450         RQ1 setting       Detection type (sew leve		IRQ2	Detection type	Low level	∼ Digi	tal filter No filter 🗸	0 (MHz)	
RG3 setting       Detection type (seviced)       Optical filter       Notifier       0.0450         RG4 setting       Detection type (seviced)       Optical filter       Notifier       0.0450         RG4 setting       Detection type (seviced)       Optical filter       Notifier       0.0450         RG5 setting       Detection type (seviced)       Optical filter       Notifier       0.0450         RG6 setting       Detection type (seviced)       Optical filter       Notifier       0.0450         RG6 setting       Detection type (seviced)       Optical filter       Notifier       0.0450         RG7 setting       Detection type (seviced)       Optical filter       Notifier       0.0450         RG7 setting       Detection type (seviced)       Optical filter       Notifier       0.0450         RG8 setting       Detection type (seviced)       Optical filter       Notifier       0.0450         RG8 setting       Detection type (seviced)       Optical filter       Notifier       0.0450         RG8 setting       Detection type (seviced)       Optical filter       Notifier       0.0450         RG8 setting       Detection type (seviced)       Optical filter       Notifier       0.0450         RG9 setting       Detection type (seviced)       Opti			Priority	Level 15 (highest)	~			
ImC3       Detection type       Doylet filter       No filter       0       (AH4)         ImC4 setting       Detection type       Exel 13 (highest)       Digital filter       No filter       0       (AH4)         ImC3 setting       Detection type       Exel 13 (highest)       Digital filter       No filter       0       (AH4)         ImC3 setting       Detection type       Exel 13 (highest)       Digital filter       No filter       0       (AH4)         ImC3 setting       Detection type       Exel 13 (highest)       Digital filter       No filter       0       (AH4)         ImC3 setting       Detection type       Exel 13 (highest)       Digital filter       No filter       0       (AH4)         ImC3 setting       Detection type       Exel 13 (highest)       Digital filter       No filter       0       (AH4)         ImC3 setting       Detection type       Exel 13 (highest)       Digital filter       No filter       0       (AH4)         ImC3 setting       Detection type       Exel 13 (highest)       Digital filter       No filter       0       (AH4)         ImC3 setting       Detection type       Exel 13 (highest)       Digital filter       No filter       0       (AH4)         ImC41 setting <t< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></t<>								
Provedy       Level 13 (highest)       Updat filter       No filter       0       (AH4)         IRQ4 setting       Detection type       Level 13 (highest)       Digital filter       No filter       0       (AH4)         IRQ5 setting       Detection type       Level 13 (highest)       Digital filter       No filter       0       (AH4)         IRQ5 setting       Detection type       Level 13 (highest)       Digital filter       No filter       0       (AH4)         IRQ5 setting       Detection type       Level 13 (highest)       Digital filter       No filter       0       (AH4)         IRQ5 setting       Detection type       Level 13 (highest)       Digital filter       No filter       0       (AH4)         IRQ6 setting       Detection type       Level 13 (highest)       Digital filter       No filter       0       (AH4)         IRQ6 setting       Detection type       Level 13 (highest)       Digital filter       No filter       0       (AH4)         IRQ6 setting       Detection type       Level 13 (highest)       Digital filter       No filter       0       (AH4)         IRQ1 setting       Detection type       Level 13 (highest)       Digital filter       No filter       0       (AH4)         IRQ1 setting			Detection to	Level and		tal filtar No filtar	0 0.000	
IRQ4 setting       Detection type       Deptat filter       No filter       0       0.0442         IRQ5 setting       Detection type       Low level       Digital filter       No filter       0       0.0442         IRQ5 setting       Detection type       Low level       Digital filter       No filter       0       0.0442         IRQ5 setting       Detection type       Low level       Digital filter       No filter       0       0.0442         IRQ5 setting       Detection type       Low level       Digital filter       No filter       0       0.0442         IRQ5 setting       Detection type       Low level       Digital filter       No filter       0       0.0442         IRQ5 setting       Detection type       Low level       Digital filter       No filter       0       0.0442         IRQ5 setting       Detection type       Low level       Digital filter       No filter       0       0.0442         IRQ5 setting       Detection type       Low level       Digital filter       No filter       0       0.0442         IRQ1 setting       Detection type       Low level       Digital filter       No filter       0       0.0442         IRQ1 setting       Detection type       Low level <t< td=""><td></td><td></td><td></td><td></td><td>Digr</td><td>tai filter</td><td>(IVIHZ)</td><td></td></t<>					Digr	tai filter	(IVIHZ)	
IRCA       Detection type       Low level       Digital filter       No filter       0       (AH2)         IRC3 setting       Detection type       Low level       Digital filter       No filter       0       (AH2)         IRC3 setting       Detection type       Low level       Digital filter       No filter       0       (AH2)         IRC3 setting       Detection type       Low level       Digital filter       No filter       0       (AH2)         IRC4       Detection type       Low level       Digital filter       No filter       0       (AH2)         IRC3 setting       Detection type       Low level       Digital filter       No filter       0       (AH2)         IRC3 setting       Detection type       Low level       Digital filter       No filter       0       (AH2)         IRC3 setting       Detection type       Low level       Digital filter       No filter       0       (AH2)         IRC3 setting       Detection type       Low level       Digital filter       No filter       0       (AH2)         IRC3 setting       Detection type       Low level       Digital filter       No filter       0       (AH2)         IRC1 setting       Detection type       Low level <t< td=""><td></td><td></td><td>Priority</td><td>Level 15 (highest)</td><td>~</td><td></td><td></td><td></td></t<>			Priority	Level 15 (highest)	~			
RO3 setting       Impact Single		IRQ4 setting						
IRQ3 setting       Detection type       Low level       Digital filter       No filter       0       (AH4)         IRQ3 cetting       Detection type       Low level       Digital filter       No filter       0       (AH4)         IRQ3 cetting       Detection type       Low level       Digital filter       No filter       0       (AH4)         IRQ5 setting       Detection type       Low level       Digital filter       No filter       0       (AH4)         IRQ5 setting       Detection type       Low level       Digital filter       No filter       0       (AH4)         IRQ5 setting       Detection type       Low level       Digital filter       No filter       0       (AH4)         IRQ5 setting       Detection type       Low level       Digital filter       No filter       0       (AH4)         IRQ6 setting       Detection type       Low level       Digital filter       No filter       0       (AH4)         IRQ6 setting       Detection type       Low level       Digital filter       No filter       0       (AH4)         IRQ1 setting       Detection type       Low level       Digital filter       No filter       0       (AH4)         IRQ1 setting       Detection type       Low lev		IRQ4	Detection type	Low level	∼ Digi	tal filter No filter 🗸	0 (MHz)	
IRQ3 setting       Detection type       Low level       Digital filter       No filter       0       (AH4)         IRQ3 cetting       Detection type       Low level       Digital filter       No filter       0       (AH4)         IRQ3 cetting       Detection type       Low level       Digital filter       No filter       0       (AH4)         IRQ5 setting       Detection type       Low level       Digital filter       No filter       0       (AH4)         IRQ5 setting       Detection type       Low level       Digital filter       No filter       0       (AH4)         IRQ5 setting       Detection type       Low level       Digital filter       No filter       0       (AH4)         IRQ5 setting       Detection type       Low level       Digital filter       No filter       0       (AH4)         IRQ6 setting       Detection type       Low level       Digital filter       No filter       0       (AH4)         IRQ6 setting       Detection type       Low level       Digital filter       No filter       0       (AH4)         IRQ1 setting       Detection type       Low level       Digital filter       No filter       0       (AH4)         IRQ1 setting       Detection type       Low lev			Priority	Level 15 (highest)	~			
ImC3       Detection type       Low level       Digital filter       No filter       0       (M+2)         ImC6       Detection type       Low level       Digital filter       No filter       0       (M+2)         ImC6       Detection type       Low level       Digital filter       No filter       0       (M+2)         ImC6       Detection type       Low level       Digital filter       No filter       0       (M+2)         ImC6       Detection type       Low level       Digital filter       No filter       0       (M+2)         ImC6       Detection type       Low level       Digital filter       No filter       0       (M+2)         ImC6       Detection type       Low level       Digital filter       No filter       0       (M+2)         ImC60       Detection type       Low level       Digital filter       No filter       0       (M+2)         ImC61       Detection type       Low level       Digital filter       No filter       0       (M+2)         ImC61       Detection type       Low level       Digital filter       No filter       0       (M+2)         ImC61       Detection type       Low level       Digital filter       No filter       0								
Inclusion       Priority       Level 13 (highest)       Digital filter       No filter       0       (M+d)         Inclusion       Priority       Level 13 (highest)       Digital filter       No filter       0       (M+d)         Inclusion       Priority       Level 13 (highest)       Digital filter       No filter       0       (M+d)         Inclusion       Priority       Level 15 (highest)       Digital filter       No filter       0       (M+d)         Inclusion       Priority       Level 15 (highest)       Digital filter       No filter       0       (M+d)         Inclusion       Priority       Level 15 (highest)       Digital filter       No filter       0       (M+d)         Inclusion       Detection type       Level 15 (highest)       Digital filter       No filter       0       (M+d)         Inclusion       Detection type       Level 15 (highest)       Digital filter       No filter       0       (M+d)         Inclusion       Detection type       Level 15 (highest)       Digital filter       No filter       0       (M+d)         Inclusion       Detection type       Level 15 (highest)       Digital filter       No filter       0       (M+d)         Inclusion       Detection type </td <td></td> <td></td> <td>D. L. M. L</td> <td></td> <td></td> <td></td> <td>0</td> <td></td>			D. L. M. L				0	
IRQ5 setting       Detection type       Low level       Digital filter       No filter       0       (M+4c)         IRQ7 setting       Detection type       Low level       Digital filter       No filter       0       (M+4c)         IRQ7 setting       Detection type       Low level       Digital filter       No filter       0       (M+4c)         IRQ7 setting       Detection type       Low level       Digital filter       No filter       0       (M+4c)         IRQ8 setting       Detection type       Low level       Digital filter       No filter       0       (M+4c)         IRQ9 setting       Detection type       Low level       Digital filter       No filter       0       (M+4c)         IRQ10       Detection type       Low level       Digital filter       No filter       0       (M+4c)         IRQ10       Detection type       Low level       Digital filter       No filter       0       (M+4c)         IRQ11       Detection type       Low level       Digital filter       No filter       0       (M+4c)         IRQ12       Detection type       Low level       Digital filter       No filter       0       (M+4c)         IRQ13       Detection type       Low level       Digital		LIKQ5			V Digr	tal filter No filter ~	(MHz)	
IRQ8       Detection type       Low level       Digital filter       No filter       0       (M+b)         IRQ7       Detection type       Low level       Digital filter       No filter       0       (M+b)         IRQ8       Etting       Detection type       Low level       Digital filter       No filter       0       (M+b)         IRQ8       Detection type       Low level       Digital filter       No filter       0       (M+b)         IRQ8       Detection type       Low level       Digital filter       No filter       0       (M+b)         IRQ9       Setting       Detection type       Low level       Digital filter       No filter       0       (M+b)         IRQ10       Detection type       Low level       Digital filter       No filter       0       (M+b)         IRQ11       Detection type       Low level       Digital filter       No filter       0       (M+b)         IRQ12       Detection type       Low level       Digital filter       No filter       0       (M+b)         IRQ13       Detection type       Low level       Digital filter       No filter       0       (M+b)         IRQ13       Detection type       Low level       Digital filter			Priority	Level 15 (highest)	$\sim$			
RQ7 setting       ImQ7       Detection type       Digital filter       No filter       0       (MHz)         ImQ8 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         ImQ8 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         ImQ8 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         ImQ10 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         ImQ10 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         ImQ11 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         ImQ12 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         ImQ13 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         ImQ13 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         ImQ13 setting       Detection type       Low le		IRQ6 setting						
IRQ7 setting       Interview       Digital filter       No filter       0       (MHz)         IRQ8 setting       IRQ8 setting       Digital filter       No filter       0       (MHz)         IRQ8 setting       IRQ8 setting       Digital filter       No filter       0       (MHz)         IRQ8 setting       Detection type       Iswiesel       Digital filter       No filter       0       (MHz)         IRQ9 setting       Detection type       Falling edge       Digital filter       No filter       0       (MHz)         IRQ9 setting       Detection type       Falling edge       Digital filter       No filter       0       (MHz)         IRQ1 setting       Detection type       Iswiesel       Stipital filter       No filter       0       (MHz)         IRQ1 setting       Detection type       Iswiesel       Digital filter       No filter       0       (MHz)         IRQ1 setting       IRQ1 setting       Detection type       Iswiesel       Digital filter       No filter       0       (MHz)         IRQ1 setting       IRQ13 setting       Detection type       Iswiesel       Digital filter       No filter       0       (MHz)         IRQ13 setting       Detection type       Iswiesel       Digita		IRQ6	Detection type	Low level	<ul> <li>✓ Digi</li> </ul>	tal filter No filter ~	0 (MHz)	
IRQ7 setting       Interview       Digital filter       No filter       0       (MHz)         IRQ8 setting       IRQ8 contents       Digital filter       No filter       0       (MHz)         IRQ8 setting       Interview       Digital filter       No filter       0       (MHz)         IRQ8 setting       Interview       Digital filter       No filter       0       (MHz)         IRQ9 setting       Interview       Digital filter       No filter       0       (MHz)         IRQ10 setting       Interview       Digital filter       No filter       0       (MHz)         IRQ10 setting       Interview       Digital filter       No filter       0       (MHz)         IRQ11 setting       Interview       Digital filter       No filter       0       (MHz)         IRQ12 setting       Interview       Interview       Digital filter       No filter       0       (MHz)         IRQ12 setting       Interview       Interview       Digital filter       No filter       0       (MHz)         IRQ13 setting       Interview       Interview       Digital filter       No filter       0       (MHz)         IRQ13 setting       Interview       Interview       Interview       Interview </td <td></td> <td></td> <td>Priority</td> <td>Level 15 (highest)</td> <td><math>\sim</math></td> <td></td> <td></td> <td></td>			Priority	Level 15 (highest)	$\sim$			
Image: RQ7       Detection type       Low level       Digital filter       No filter       0       (MHz)         Image: RQ8       Detection type       Low level       Digital filter       No filter       0       (MHz)         Image: RQ3       Detection type       Low level       Digital filter       No filter       0       (MHz)         Image: RQ3       Detection type       Low level       Digital filter       No filter       0       (MHz)         Image: RQ3       Detection type       Falling edge       Digital filter       No filter       0       (MHz)         Image: RQ3       Detection type       Falling edge       Digital filter       No filter       0       (MHz)         Image: RQ10       Detection type       Low level       Digital filter       No filter       0       (MHz)         Image: RQ11       Detection type       Low level       Digital filter       No filter       0       (MHz)         Image: RQ12       Detection type       Low level       Digital filter       No filter       0       (MHz)         Image: RQ12       Detection type       Low level       Digital filter       No filter       0       (MHz)         Image: RQ12       Detection type       Low level								
RQ3 setting         IRQ9         Detection type       Low level         Digital filter       No filter         IRQ9 setting         IRQ9 setting         IRQ10       Detection type         IRQ11       Detection type         IRQ12       Detection type         IRQ13       Detection type         IRQ14       Detection type         IRQ12       Detection type         IRQ13       Detection type         IRQ14       Detection type         IRQ15       Detection type         IRQ13       Detection type         IRQ14       Detection type         IRQ15       Detection type         IRQ13       Detection type         IRQ13       Detection type         IRQ14       Detection type         IRQ15       Detection type         IRQ14       Detection type         IRQ14       Detection type         IRQ14       Detection type         IRQ14       Detection type         IRQ14 <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>								
IRQ8       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ9       Priority       Level 15 (highest)       Digital filter       No filter       0       (MHz)         IRQ10       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ10       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ11       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ11       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ12       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ12       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14       Detection type       Low level       Digital filter       No filter       0 </td <td></td> <td>L] IRQ7</td> <td>Detection type</td> <td>Low level</td> <td>✓ Digi</td> <td>tal filter No filter ~</td> <td>0 (MHz)</td> <td></td>		L] IRQ7	Detection type	Low level	✓ Digi	tal filter No filter ~	0 (MHz)	
IRQ3       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ3 setting       IRQ3 setting       Digital filter       No filter       0       (MHz)         IRQ3 setting       IRQ10       Detection type       Digital filter       No filter       0       (MHz)         IRQ10       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ11 setting       IRQ12       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ12 setting       IRQ12 setting       IRQ12 setting       Digital filter       No filter       0       (MHz)         IRQ12 setting       IRQ13       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       IRQ14 setting       Digital filter       No filter       0       (MHz)         IRQ14 setting       IRQ14 setting       Digital filter       No filter       0       (MHz)         IRQ14 setting       Priority       Low level       Digital filter       No filter       0       (MHz)         IRQ14 setting       IRQ15 setting       Digital filter       No filter       0 <td></td> <td></td> <td>Priority</td> <td>Level 15 (highest)</td> <td><math>\sim</math></td> <td></td> <td></td> <td></td>			Priority	Level 15 (highest)	$\sim$			
IRQ3       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ3 setting       IRQ3 setting       Digital filter       No filter       0       (MHz)         IRQ3 setting       IRQ10       Detection type       Digital filter       No filter       0       (MHz)         IRQ10       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ11 setting       IRQ12       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ12 setting       IRQ12 setting       IRQ12 setting       Digital filter       No filter       0       (MHz)         IRQ12 setting       IRQ13       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       IRQ14 setting       Digital filter       No filter       0       (MHz)         IRQ14 setting       IRQ14 setting       Digital filter       No filter       0       (MHz)         IRQ15       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14 setting       IRQ15       Detection type       Low level       Digital filter<		IRQ8 setting						
IRQ3 setting       Priority       Level 15 (highest)       Digital filter       No filter       0       (MH2)         IRQ10       Detection type       Low level       Digital filter       No filter       0       (MH2)         IRQ10       Detection type       Low level       Digital filter       No filter       0       (MH2)         IRQ11       Detection type       Low level       Digital filter       No filter       0       (MH2)         IRQ12       Detection type       Low level       Digital filter       No filter       0       (MH2)         IRQ13       Detection type       Low level       Digital filter       No filter       0       (MH2)         IRQ13       Detection type       Low level       Digital filter       No filter       0       (MH2)         IRQ13 setting       IRQ14       Detection type       Low level       Digital filter       No filter       0       (MH2)         IRQ14 setting       IRQ14       Detection type       Low level       Digital filter       No filter       0       (MH2)         IRQ15 setting       IRQ15 setting       Digital filter       No filter       0       (MH2)			Detection type	Low level	<ul> <li>✓ Diqi</li> </ul>	tal filter No filter 🗸	0 (MHz)	
IRQ9 setting       Detection type       Falling edge       Digital filter       No filter       0       (MHz)         IRQ10 setting       IRQ10       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ11 setting       IRQ11       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ11 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ11 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ12 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       IRQ14       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14 setting       IRQ14       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ15 setting       IRQ15       Detection type       Low level       Digital filter       No filter       0 <td></td> <td></td> <td></td> <td></td> <td>~</td> <td></td> <td></td> <td></td>					~			
Image: RQ1       Detection type       Falling edge       Digital filter       No filter       0       (MHz)         ImQ10 setting       ImQ10       Detection type       Low level       Digital filter       No filter       0       (MHz)         ImQ11 setting       ImQ11       Detection type       Low level       Digital filter       No filter       0       (MHz)         ImQ12 setting       ImQ12 setting       Digital filter       Digital filter       No filter       0       (MHz)         ImQ12 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         ImQ12 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         ImQ13 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         ImQ14 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         ImQ14 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         ImQ14 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         ImQ15 <td></td> <td></td> <td></td> <td>Level 15 (linghest)</td> <td></td> <td></td> <td></td> <td></td>				Level 15 (linghest)				
IRQ10 setting       IRQ10       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ11 setting       IRQ11       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ12 setting       IRQ12 setting       Digital filter       Digital filter       No filter       0       (MHz)         IRQ12 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       IRQ13       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       IRQ14 setting       Digital filter       Digital filter       No filter       0       (MHz)         IRQ14 setting       IRQ14 setting       Digital filter       No filter       0       (MHz)         IRQ14 setting       IRQ14 setting       Digital filter       No filter       0       (MHz)         IRQ15       Detection type       Low level       Digital filter       No filter       0       (MHz)								
IRQ10 setting         IRQ10       Detection type       Low level       Digital filter       No filter       0 (MHz)         RQ11 setting       Detection type       Low level       Digital filter       No filter       0 (MHz)         IRQ12 setting       Detection type       Low level       Digital filter       No filter       0 (MHz)         IRQ12 setting       IRQ12       Detection type       Low level       Digital filter       No filter       0 (MHz)         IRQ13 setting       IRQ13       Detection type       Low level       Digital filter       No filter       0 (MHz)         IRQ13 setting       IRQ13       Detection type       Low level       Digital filter       No filter       0 (MHz)         IRQ14 setting       IRQ14       Detection type       Low level       Digital filter       No filter       0 (MHz)         IRQ14 setting       IRQ14       Detection type       Low level       Digital filter       No filter       0 (MHz)         IRQ15 setting       IRQ15       Detection type       Low level       Digital filter       No filter       0 (MHz)		RQ9	Detection type	Falling edge	✓ Digi	tal filter No filter V	0 (MHz)	
IRQ10       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ11 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ12 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ12 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ15       Detection type       Low level <td></td> <td></td> <td>Priority</td> <td>Level 15 (highest)</td> <td><math>\sim</math></td> <td></td> <td></td> <td></td>			Priority	Level 15 (highest)	$\sim$			
IRQ10       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ11 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ12 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ12 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ15       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ15       Detection type       Low level		IRQ10 setting						
IRQ11 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ12 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ12 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ15 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)			Detection type	Low level	<ul> <li>✓ Diai</li> </ul>	tal filter No filter ~	0 (MHz)	
IRQ11 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ12 setting       IRQ12       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       IRQ13 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14 setting       Priority       Level 15 (highest)       Digital filter       No filter       0       (MHz)         IRQ15 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)								
IRQ11       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ12 setting       IRQ12       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       IRQ13 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       IRQ14       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14 setting       IRQ14       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ15 setting       IRQ15 setting       Inforty Level 15 (highest)       Digital filter       No filter       0       (MHz)         IRQ15 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)			Priority	Level 19 (nighest)				
IRQ12 setting         IRQ12       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       IRQ13       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14 setting       IRQ14       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14 setting       IRQ14       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ15 setting       IRQ15       Detection type       Low level       Digital filter       No filter       0       (MHz)								
IRQ12 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       IRQ13       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14 setting       IRQ14       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14 setting       IRQ14       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ15 setting       IRQ15       Detection type       Low level       Digital filter       No filter       0       (MHz)		[] IRQ11	Detection type	Low level	∼ Digi	tal filter No filter 🗸	0 (MHz)	
IRQ12       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       IRQ13       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ15       Detection type       Low level       Digital filter       No filter       0       (MHz)			Priority	Level 15 (highest)	$\sim$			
IRQ12       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13 setting       IRQ13       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ13       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ15       Detection type       Low level       Digital filter       No filter       0       (MHz)		IRO12 setting						
Priority Level 15 (highest)         IRQ13 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14 setting       Priority       Level 15 (highest)        Digital filter       No filter       0       (MHz)         IRQ14 setting       Priority       Level 15 (highest)        Digital filter       No filter       0       (MHz)         IRQ15 setting         Detection type       Low level        Digital filter       No filter       0       (MHz)         IRQ15 setting        Detection type       Low level        Digital filter       No filter       0       (MHz)			Detection type	Low level	<ul> <li>→ Diai</li> </ul>	tal filter No filter	(MHz)	
IRQ13 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14 setting       Priority       Level 15 (highest)        Digital filter       No filter       0       (MHz)         IRQ14 setting       Priority       Level 15 (highest)        Digital filter       No filter       0       (MHz)         IRQ15 setting       IRQ15       Detection type       Low level       Digital filter       No filter       0       (MHz)						in the first of	(IVIII 14/	
IRQ13       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14 setting       IRQ14 setting       Detection type       Low level       Digital filter       No filter       0       (MHz)         IRQ14 setting       Priority       Level 15 (highest)       Digital filter       No filter       0       (MHz)         IRQ15       Detection type       Low level       Digital filter       No filter       0       (MHz)			Priority	Level 15 (highest)				
Priority       Level 15 (highest)         IRQ14 setting       Digital filter       No filter       0       (MHz)         Priority       Level 15 (highest)              IRQ15       Detection type       Low level       Digital filter       No filter       0       (MHz)		IRQ13 setting						
IRQ14 setting       IRQ14       Detection type       Low level       Digital filter       No filter       0       (MHz)         Priority       Level 15 (highest)		[] IRQ13	Detection type	Low level	<ul> <li>✓ Digi</li> </ul>	tal filter No filter 🗸	0 (MHz)	
IRQ14 setting □ IRQ14 Detection type Low level			Priority	Level 15 (highest)	$\sim$			
IRQ14       Detection type       Low level       Digital filter       No filter       0       (MHz)         Priority       Level 15 (highest)								
Priority     Level 15 (highest)       IRQ15 setting     IRQ15     Detection type     Low level     Digital filter     No filter     0     (MHz)								
IRQ15 setting ☐ IRQ15 Detection type Low level Digital filter No filter ○ 0 (MHz)		L] IRQ14	Detection type	Low level	✓ Digi	tal filter No filter ~	0 (MHz)	
IRQ15     Detection type     Low level     Digital filter     No filter     0     (MHz)			Priority	Level 15 (highest)	$\sim$			
IRQ15     Detection type     Low level     Digital filter     No filter     0     (MHz)		IRQ15 setting						
			Detection type	Low level	<ul> <li>✓ Diai</li> </ul>	tal filter No filter 🗸	0 (MHz)	
Priority Level 15 (highest)							(····/	
		~	Priority	Level 15 (highest)				

Figure 4-18 Config\_ICU setting

#### 4.5.4 Ports

Referring to the RSK schematic, LED0 is connected to P95, LED1 is connected to P94, LED2 is connected to P93 and LED3 is connected to PE0. PA2 is used as one of the LCD control lines, together with P61, P62 and P63.

Click 'Add component' <sup>to</sup> icon. In 'Software Component Selection' dialog -> Type, select 'Drivers'. Select 'Ports' as shown in Figure 4-19 then click 'Next'

🔥 New Co	omponent				$\times$
Software Component Selection Select component from those available in list					
Type Function Filter	Drivers All				~
	Output Enable	Type Code Generator	Version 1.3.0		^
E SCI/S	s 1 Mode Timer GCIF Asynchronous Mode GCIF Clock Synchronous M	Code Generator Code Generator Code Generator Code Generator	1.5.0 1.3.0 1.3.0	3	×
Descriptio This soft Common	only latest version on ware component provides co n features such as reading, wi onfigured. Enabling features	riting, and setting the direct	ion of ports	and pins	
ups are a	d more software components e general settings		and interna	ii puii-	~
?	< <u>B</u> ack	<u>N</u> ext > <u>F</u> inis	h	Cance	4

Figure 4-19 Select Ports

In 'Add new configuration for selected component' dialog -> Resource, select 'PORT' as shown in Figure 4-20 below then click 'Finish'.

~

Figure 4-20 Select resource – PORT

## 'PORT6', 'PORT9', 'PORTA', 'PORTE' tick box is checked as shown in **Figure 4-21** below.

Components 🛛 🖓 🖻 🕀 🖨	• C	onfigure				
10 T		Port selection	PORT6	PORT9	PORTA	PORTE
type filter text						
<ul> <li>✓ ➢ Startup</li> <li>∧ ➢ Generic</li> </ul>		PORT0		PO	RT1	
<ul> <li>interview of the second second</li></ul>		PORT2		PO	RT3	
✓		DORT4		PO	RT5	
✓ → I/O Ports Config_PORT		PORT6		PO	RT7	
<ul> <li>E Timers</li> <li>Config_CMT0</li> </ul>		PORT8		<mark>∕ P</mark> O	RT9	
Config_CMT1				PO	RTB	
				<mark>∠</mark> PO	RTE	

Figure 4-21 Select Port selection

Navigate to the 'Ports' configure these four I/O lines and LCD control lines as shown in **Figure 4-22**, **Figure 4-23**, **Figure 4-24** and **Figure 4-25** below. Ensure that the 'Output 1' tick box is checked, except P63. Select 'PORT6' tab.

🔅 *SC_Tutorial.scfg 🛛		
Software component configurat	ion	🔞 🔁
$\begin{array}{c} \text{Components} \\ \downarrow^a_Z \ \boxdot \ \boxplus \ \clubsuit \\ \checkmark \end{array}$	Configure	^
10 To	Port selection PORT6 PORT9 PORTA PORTE	
type filter text	Apply to all Unused GPIO O In O Ut Pull-up CMOS output Output 1	
<ul> <li>interrupt</li> <li>interrupt</li> <li>interrupt</li> </ul>	P60 Unused GPIO O In O Out Pull-up CMOS output Output 1	
✓	P61 ○ Unused GPIO ○ In	
Config_CMT1	P62 ○ Unused GPIO ○ In	
	P63 ○ Unused GPIO ○ In  Out □ Pull-up CMOS output ✓ □Output 1	
	P64            Unused GPIO             Output             Output 1	
¥	< P65	~ ~ ~

Figure 4-22 Select PORT6 tab

### Select 'PORT9' tab.

聽 *SC_Tutorial.scfg 🛛		
Software component configuration	n	🔞 🖹
Components $\downarrow^{a}_{Z} \Box \Box \Rightarrow$	Configure	^
10 T	Port selection PORT6 PORT9 PORTA PORTE	
type filter text		
🗸 🗁 Startup	Apply to all	
✓   Generic	© Unused GPIO ○ In ○ Out □ Pull-up CMOS output ∨ □ Output 1 Normal drive output ∨	
V 🗁 Drivers	P90	
✓	● Unused GPIO ○ In ○ Out □ Pull-up CMOS output ∨ □ Output 1 Normal drive output ∨	
Config_PORT	P91	
✓ ⇒ Timers Config_CMT0	● Unused GPIO ○ In ○ Out □ Pull-up CMOS output ∨ □ Output 1 Normal drive output ∨	
Config_CMT1	P92	
Config_CMT2	● Unused GPIO ○ In ○ Out □ Pull-up CMOS output ∨ □ Output 1 Normal drive output ∨	
	P93	
	O Unused GPIO O In Out □ Pull-up CMOS output ✓ Output 1 Normal drive output ✓	
	P94	
	O Unused GPIO O In  O Out □ Pull-up CMOS output ✓ Output 1 Normal drive output ✓	
	P95	
	◯ Unused GPIO ◯ In	
	P96	~
Overview Board Clocks Components Pins Int	tamunte	>
Overview board clocks components Pins int	icinality and a second s	

Figure 4-23 Select PORT9 tab

#### Select 'PORTA' tab.

$ \downarrow_Z^a \Box \blacksquare \ddagger \ddagger $		ıre								'
type filter text	Port	selection PORT6	i PORT	9 PORTA	PORTE					
✓ (⇒) Startup ✓ (⇒) Startup ✓ (⇒) Generic ✓ (≥) Generic ✓ (≥) F_DSp		Apply to all Unused GPIO	◯In	Out	Pull-up	CMOS output	~ [](	Dutput 1	Normal drive output	~
<ul> <li>✓ Config_ICU</li> <li>✓ Config_ICU</li> <li>✓ Config_ICU</li> <li>✓ Config_ICU</li> </ul>		PA0 • Unused GPIO	◯In	Out	Pull-up	CMOS output	~ [](	Dutput 1	Normal drive output	~
Config_CONT Config_CONT Config_CMT0 Config_CMT1 Config_CMT2		PA1 Unused GPIO	◯In	() Out	Pull-up	CMOS output	~ [](	Output 1	Normal drive output	~
		PA2 O Unused GPIO	Oln	) Out	Pull-up	CMOS output	~	Dutput 1	Normal drive output	~
		PA3 Unused GPIO	◯In	() Out	Pull-up	CMOS output	~ [](	Output 1	Normal drive output	~
		₽A4 ● Unused GPIO	◯In	() Out	Pull-up	CMOS output	~ [](	Output 1	Normal drive output	~
~	<	DAS								`

Figure 4-24 Select PORTA tab

#### Select 'PORTE' tab.



Figure 4-25 Select PORTE tab



#### 4.5.5 SCI/SCIF Asynchronous Mode

In the RSKRX66T SCI11 is connected via a Renesas RL78/G1C to provide a USB virtual COM port as shown in the schematic.

Click 'Add component' じ icon.

In 'Software Component Selection' dialog -> Type, select 'Drivers'. Select 'SCI/SCIF Asynchronous Mode' as shown in **Figure 4-26** then click 'Next'.

Kew Component		×				
Software Component Selection Select component from those available in list						
Type Drivers 🗸						
Function All					$\sim$	
Filter						
Components		Туре	Version		^	
PWM Mode Timer		Code Generator				
SCI/SCIF Asynchro		Code Generator	1.3.0			
SCI/SCIF Clock Syn		Code Generator	1.3.0			
Single Scan Mode		Code Generator	1.5.0			
Smart Card Interfac	te Mode	Code Generator	1.3.0		~	
<					>	
Show only latest ver	sion					
Description						
This software component provides configurations for SCI(SCIF) single(multi- processor) asynchronous mode.						
Download more software components						
Configure general settings						
?	< <u>B</u> ack	<u>N</u> ext > <u>F</u> inis	h	Cance	el	

Figure 4-26 Select SCI/SCIF Asynchronous Mode

In 'Add new configuration for selected component' dialog -> Work mode, select 'Transmission/Reception' as shown in **Figure 4-27** below.

🔇 New Component	— 🗆 X
Add new configuration	for selected component
SCI/SCIF Asynchronous	Mode
Configuration name:	Config_SCI1
Work mode:	Transmission/Reception ~
Resource:	Transmission Reception
	Transmission/Reception Multi-processor Transmission Multi-processor Reception Multi-processor Transmission/Reception
?	< Back Next > Einish Cancel

Figure 4-27 Select Work mode – Transmission/Reception



In 'Resource', select 'SCI11' as shown in **Figure 4-28** below.

孩 New Component	~	_		×
Add new configuration	n for selected component			
SCI/SCIF Asynchronous	Mode			
Configuration name:	Config_SCI1			
Work mode:	Transmission/Reception			$\sim$
Resource:	SCI1			$\sim$
	SCI1 SCI5 SCI6 SCI8 SCI9 SCI11 SCI12			
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> ini	sh	Cano	el

Figure 4-28 Select Resource – SCI11

Ensure that the 'Configuration name' is set to 'Config\_SCI11' as shown in Figure 4-29 below then click 'Finish'

🕼 New Component – 🗆 🗙							
Add new configuration for selected component							
SCI/SCIF Asynchronous Mo	de						
Configuration name:	Config_SCI11						
Work mode:	Transmission/Reception	~					
Resource:	SCI11	$\sim$					
? < <u>F</u>	ack <u>N</u> ext > <u>F</u> inish	Cancel					

Figure 4-29 Ensure Configuration name - Config\_SCI11



Configure SCI11 as shown in **Figure 4-30**. Ensure the 'Start bit edge detection' is set as 'Falling edge on RXD11 pin' and the 'Bit rate' is set to 19200 bps. All other settings remain at their defaults.

omponents Jª	- + ⇒ -	Configure		
type filter text	65	FIFO mode setting Non-FIFO mode	○ FIFO mode	
✓ ➢ Startup ✓ ➢ Generic	^	Start bit edge detection setting O Low level on RXD11 pin	Falling edge on RXD11 pin	
<ul> <li>✓ r_bsp</li> <li>✓ ➢ Drivers</li> <li>✓ ➢ Interrupt</li> </ul>		Data length setting O 9 bits	• 8 bits	○ 7 bits
<ul> <li>✓ Config_ICU</li> <li>✓ (⇒ I/O Ports</li> </ul>		Parity setting None	◯ Even	Odd
<ul> <li>Config_PORT</li> <li>Communications</li> <li>Config_SCI11</li> </ul>	_	Stop bit length setting	◯ 2 bits	
✓		Transfer direction setting	◯ MSB-first	
Config_CMT2		Transfer rate setting		
		Transfer clock	Internal clock	~
		Base clock	16 cycles for 1-bit period	$\sim$
		Bit rate	19200	<ul> <li>(bps) (Actual value: 19230.769, Error: 0.16%)</li> </ul>
		Enable modulation duty correction		
		SCK11 pin function	SCK11 is not used	$\checkmark$

Figure 4-30 Config\_SCI11 setting



#### 4.5.6 SPI Clock Synchronous Mode

In the RSKRX66T SCI6 is used as an SPI master for the Pmod LCD on the PMOD1 connector as shown in

the schematic. Click 'Add component' 👈 icon.

In 'Software Component Selection' dialog -> Type, select 'Drivers'. Select 'SPI Clock Synchronous Mode' as shown in **Figure 4-31** then click 'Next'.

💰 New C	New Component						
	Component Selection mponent from those availab	le in list					
Туре	Drivers			~			
Function	All			~			
Filter							
Compor		Туре	Version	^			
	t Card Interface Mode	Code Generator	1.3.0				
· · · · · · · · · · · · · · · · · · ·	lock Synchronous Mode Operation Mode	Code Generator Code Generator	1.3.0				
	ge Detection Circuit	Code Generator	1.3.0				
	hdog Timer	Code Generator	1.3.0				
<	,			× *			
Descriptio				-			
bus). It i	nponent provides clock synd ncludes 4 transfer modes: SI /receive and Master transmi	ave transmit/receive, Sla					
Download	d more software component	ts					
<u>Configure</u>	e general settings						
?	< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish	Cancel			

Figure 4-31 Select SPI Clock Synchronous Mode

Ensure Operation, is set to 'Master transmit only' as shown in Figure 4-32 below.

🚯 New Component - 🗆 X						
Add new configura						
SPI Clock Synchron	ous Mode					
Configuration name	:: Config_RSPI0					
Operation:	Slave transmit/receive			$\sim$		
Resource:	Slave transmit/receive Slave transmit only Master transmit/receive					
	Master transmit only					
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish		Can	cel		

Figure 4-32 Select Operation – Master Transmit



In 'Resource', select 'SCI6' as shown in Figure 4-33 below.

	孩 New Component – 🗆 🗙							
Add new configuration for selected component								
	- SPI Clock Synchron	ous Mode						
	Configuration nam	e: Config_RSPI0						
	Operation:	Master transmit only		$\sim$				
	Resource:	SCI6		$\sim$				
		RSPI0 SCI1 SCI5						
		SCI6						
		SCI8 SCI9 SCI11 SCI12	_					
	?	< <u>B</u> ack <u>N</u> ext > <u>Finish</u>	Can	cel				

Figure 4-33 Select Resource – SCI6

Ensure that the 'Configuration name' is set to 'Config\_SCI6' as shown in Figure 4-34 below then click 'Finish'

Kew Component -						
Add new configuration for selected component						
SPI Clock Synchronous N	lode					
Configuration name:	Config_SCI6					
Operation:	Master transmit only			$\sim$		
Resource:	SCI6			$\sim$		
?	K Back Next > Finish		Cance	٤l		

Figure 4-34 Ensure Configuration name - Config\_SCI6

#### RSKRX66T

Configure SCI6 as shown in **Figure 4-35**. Ensure the 'Transfer direction' is set as 'MSB-first' and the 'Bit rate' is set to 8000 kbps. All other settings remain at their defaults.

Components 👌 📲 🗉	- 🖶 🕀 -	Configure	
	ت ت	Transfer direction setting	
type filter text		○ LSB-first	
🗸 🗁 Startup	^	Data inversion setting	
✓ ⇐ Generic		Normal	○ Inverted
V 🗁 Drivers		Transfer speed setting	
Interrupt     Config_ICU     Config_ICU		Transfer clock	Internal clock (SCK6 pin functions as clock output pin) $\qquad \qquad \qquad$
<ul> <li>I/O Ports</li> <li>Config_PORT</li> </ul>		Bit rate	8000 (kbps) (Actual value: 10000, Error: 25%)
<ul> <li>Communications</li> <li>Config_SCI11</li> </ul>		Enable modulation duty correction	
Config_SCl6		Clock setting	
Fimers Config_CMT0		Enable clock delay	Enable clock polarity inversion
Config_CMT1		Data handling setting	
Config_CMT2		Transmit data handling	Data handled in interrupt service routine $\sim$
		Interrupt setting	
		TXI6 priority	Level 15 (highest) $\checkmark$
	•	TEI6 priority (Group BL0)	Level 15 (highest) $\checkmark$

Figure 4-35 Config\_SCI6 setting



#### 4.5.7 Single Scan Mode S12AD

We will be using the S12AD on Single Scan Mode on the AN000 input, which is connected to the RV1 potentiometer output on the RSK. The conversion start trigger will be via the pin connected to SW3. Click

'Add component' **t** icon. In 'Software Component Selection' dialog -> Type, select 'Drivers'. Select 'Single Scan Mode S12AD' as shown in **Figure 4-36** then click 'Next'.

🔇 New Component		— [	⊐ ×
Software Component Selection Select component from those availabl	e in list		
Type Drivers			~
Function All			~
Filter			
Components	Туре	Version	^
BCI/SCIF Clock Synchronous M	Code Generator	1.3.0	
🖶 Single Scan Mode S12AD	Code Generator	1.5.0	
H Smart Card Interface Mode	Code Generator	1.3.0	
H SPI Clock Synchronous Mode	Code Generator	1.3.0	
H SPI Operation Mode	Code Generator	1.3.0	~
<	0 · 0 ·		>
Show only latest version Description This software component provides si Converter which the analog inputs an ascending channel order.			
Download more software components	2		
Configure general settings			
? < <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish (	Cancel

Figure 4-36 Select Single Scan Mode S12AD

In 'Add new configuration for selected component' dialog -> Resource, select 'S12AD0' as shown in **Figure 4-37** below then click 'Finish'.

孩 New Component		_	
Add new configuration	for selected component		
Single Scan Mode S12AD			
Configuration name:	Config_S12AD0		
Resource:	S12AD0		~
?	Back Next > Finis	sh	Cancel

Figure 4-37 Select resource – S12AD0



Configure S12AD0 as shown in **Figure 4-38** and **Figure 4-39**. Ensure the 'Analog input channel' tick box for AN000 is checked and the 'Start trigger source' is set to 'A/D conversion start trigger pin'. All other settings remain at their defaults.



Figure 4-38 Config\_S12AD0 setting (1)



Data placement	Right-alignment $\checkmark$
Automatic clearing	Disable automatic clearing $\sim$
Addition/Average mode select	Addition mode $\sim$
Addition count	1-time $\checkmark$
Window function setting	
Disable	○ Enable
Window A/B operation setting	
Enable comparison window A	Enable comparison window B
Window A/B complex condition	Window A comparison condition matched OR window B comparison condition matched
A/D comparison A setting	
Reference data 0 for comparison	0
Reference data 1 for comparison	0
Use comparator for AN000	Reference data 0 > A/D-converted value <
Use comparator for AN001	Reference data 0 > A/D-converted value <
Use comparator for AN002	Reference data 0 > A/D-converted value <
Use comparator for AN003	Reference data 0 > A/D-converted value <
Use comparator for AN007	Reference data 0 > A/D-converted value ~
A/D comparison B setting	
Reference data 0 for comparison	0
Reference data 1 for comparison	0
Comparison B channel	Unused $\checkmark$
	Reference data 0 > A/D-converted value $\qquad \qquad \lor$
Input sampling time setting	
Dedicated sample and hold circuit	0.675 (µs) (Actual value: 0.675)
AN000/Self-diagnosis	0.675 (µs) (Actual value: 0.675)
AN001	0.675 (µs) (Actual value: 0.675)
AN002	0.675 (µs) (Actual value: 0.675)
AN003	0.675 (µs) (Actual value: 0.675)
AN007	0.675 (µs) (Actual value: 0.675)
	(Total conversion time: 1.575µs)
ADST0 output setting	
Enable ADST0 pin output	
Event link control setting	
ELC scan end event generation condition	On completion of all scans $\qquad \lor$
Interrupt setting	
Enable AD conversion compare interrupt (S1	2CMPAI) Enable AD conversion compare interrupt B (S12CMPBI)
Group BL1 priority	Level 15 (highest) 💛

Figure 4-39 Config\_S12AD0 setting (2)



### 4.6 Pins configuration page

Smart Configurator assigns pins to the software components that are added to the project. Assignment of the pins can be changed at Pins page. \_\_\_\_\_

∰ *SC_Tutorial.scfg ⊠	
Pin configuration	
Hardware Resource	⊟ ↓ª₂ ♣
Type filter text	
All Clock generator Clock frequency accuracy measurement circu Buses EXDMA controller Interrupt controller unit Multi-function timer pulse unit 3 MTU0 MTU0 MTU1 MTU2 MTU2 MTU3 MTU4 MTU5	it
MTU6	
Pin Function Pin Number	
Overview Board Clocks Components Pins Interrupts	

Figure 4-40 Pin configuration page

#### 4.6.1 Change pin assignment of a software component

To change the pin assignment of a software component in Pin Function list. Click to change view to show by Software Components.

∰ *SC_Tutorial.scfg ⊠	
Pin configuration	
Software Components	🕀 🕞 🖧 👪
Type filter text	

Figure 4-41 Change view to show by Software Components



Select the Config\_ICU of software component. In the Pin Function list -> Assignment column, change the pin assignment IRQ0 to P10, IRQ9 to PB3. Ensure the 'Enable' tick box of IRQ0 and IRQ9 are checked, as shown in **Figure 4-42**.



Figure 4-42 Configure pin assignment - Config\_ICU

Select the Config\_SCI11 of software component. In the Pin Function list -> Assignment column, Ensure the 'Enable' tick box of RXD11 and TXD11 are checked and Assignment column of RXD11 is PB6 and TXD11 is PB5 as shown in **Figure 4-43**.

oftware Components 🕀 🕞 🛱 d						2 🖬 🗠 u
Type filter text	Type pin function					
🗸 💑 Compare Match Timer	Enabled	Function	Assignment	Pin Number	Direction	Remarks
Config_CMT0		CTS11#	Not assigned	Not assigned	None	
Config_CMT1		RTS11#	Not assigned	Not assigned	None	
<ul> <li>Config_CMT2</li> <li>Interrupt Controller</li> </ul>		RXD11	PB6/TRDATA0/A3/GTIOC2A/GTIOC2A#/RXD5/SMISO	27	1	
Config_ICU		SCK11	Not assigned	Not assigned	None	
✓ ▲ Ports		TXD11	PB5/TRSYNC/A2/GTIOC2B/GTIOC2B#/TXD5/SMOSI5	28	0	
Config_PORT						
<ul> <li><u>SCI/SCIF Asynchronous Mode</u></li> </ul>						
Config_SCI11						
✓ ♣ SPI Clock Synchronous Mode						
Config_SCl6						
🗸 🚣 Single Scan Mode S12AD						
Config_S12AD0						
🗸 💑 r_bsp						
📦 r_bsp						
	<					

Figure 4-43 Configure pin assignment - Config\_SCI11


Select the Config\_SCI6 of software component. In the Pin Function list -> Assignment column, Ensure the 'Enable' tick box of SCK6 and SMOSI6 are checked and Assignment column of SCK6 is PA4, SMOSI6 is PB0 as shown in **Figure 4-44**.

Type filter text	Type pin	function				
<ul> <li>Compare Match Timer</li> <li>Config_CMT0</li> <li>Config_CMT1</li> <li>Config_CMT2</li> <li>Interrupt Controller</li> <li>Config_ICU</li> <li>Config_PORT</li> <li>SCI/SCIF Asynchronous Mode</li> <li>Config_SCI6</li> <li>Single Scan Mode S12AD</li> <li>Config_S12AD0</li> <li>Interrupt</li> </ul>	Type pin	function SCK6 SMISO6 SS6#	Assignment PA4/MTIOC1B/MTIOC1B#/TMCI7/SCK6/TXD8/SMOS Not assigned Not assigned	Pin Number 37 Not assigned 35 Not assigned	Direction IO None O None O O O O O O O O O O O O O O O O O O O	Remarks           - </th

Figure 4-44 Configure pin assignment - Config\_SCI6

Select the Config\_S12AD0 of software component. In the Pin Function list -> Assignment column, Ensure the 'Enable' tick box of AN000, AVCC0, AVSS0 and ADTRG0# are checked and Assignment column of AN000 is P40, ADTRG0# is P20 as shown in **Figure 4-45**.

ftware Components	J <sup>a</sup> z → Pin Function	n				것 🖬 🗠 🛆
Type filter text	Type pin fi	unction				
<ul> <li>         Compare Match Timer         <ul> <li>Config_CMT0</li> <li>Config_CMT1</li> <li>Config_CMT2</li> <li>Interrupt Controller</li> <li>Config_ICU</li> <li>Config_PORT</li> <li>Config_SCI11</li> <li>SPI Clock Synchronous Mode</li> <li>Config_SCI6</li> <li>Single Scan Mode S12AD</li> <li>Config_S12AD0</li> <li>T_bsp</li> <li>r_bsp</li> </ul> </li> </ul>	Enabled	Function ADST0 ADTRG0# AN000 AN001 AN002 AN003 AN007 AVCC0 AVS0 PGAVSS0	Assignment Not assigned P20/D15/MTIOC9C/MTCLKB/MTIOC9C#/MTCLKB#/ P40/AN00/CMPC00/CMPC01 Not assigned Not assigned Not assigned AVCC0 AVSS0 Not assigned	Pin Number Not assigned 90 Not assigned Not assigned Not assigned 93 94 Not assigned	Direction None I None None None None None	Remarks
	<					>

Figure 4-45 Configure pin assignment - Config\_S12AD0

Peripheral function configuration is now complete. Save the project using the File -> Save, then click Generate Code' at location of **Figure 4-46**.

Pin configuration			<b>1</b>

Figure 4-46 Generate Code Button

If the Section Setting Dialog is displayed as shown in the Figure 4-47, Please check the box and click "Yes".



Figure 4-47 Section Setting Dialog

The Console pane should report 'Code generation is successful', as shown Figure 4-48 below.



Figure 4-48 Smart Configurator console

When code generation is executed, the startup files generated at the time of CS+ project creation are replaced with those generated by Smart Configurator. **Figure 4-49** the project tree after code generation. In the next chapter, user code is added to these files, and SC\_Tutorial is completed by adding a new source file to the project.



Figure 4-49 Smart Configurator folder structure



# 5. Completing the Tutorial Project

## 5.1 Project Settings

•	In the 'Project Tree' pane, select 'CC-RX (Build Tool)'. The build properties will appear in the main window. CS+ creates a single build configuration called 'Default Build' for the project. This has standard code optimisation turned on by default.	CC-RX Property           Build Mode         DefaultBuild           Charge property value for all build modes at once         No           CPU         No           Instruction set architecture         RxV3 architecture(sa=rxv3)           Uses floating-point operation instructions         Test (sa = rxv3)           Endia type for data         Little endian dataferidam-little)           Rounding method for floating-point constant operations         round to enseret(cround-nearett)           Handling of denormalized numbers in floating-point constants         Handles as areos(denomalize-off)           Replaces the intrype with the short type         Handles as unsigned char(unsigned_char)           Sign of the char type         Handles as unsigned (unsigned_char(unsigned_char))           Sign of the bit-field type         Handles as unsigned (unsigned_char(unsigned_char))           Sign of the bit-field type         Handles as unsigned (unsigned_char(unsigned_char))           Charles the curve structure members is 1         No(unspack)           Charles the boundary only in fast interrupt functions         None(firt_register           Base register for RUM         None           Address value of base register that se
•	Select the 'Compile Options' tab at the bottom of the properties window pane. Under 'Language of the C source file' select 'C99(-lang=c99)' as shown opposite.	CC-RX Property     Source     Language of the C++ source file     Comparison of the C++ source file
•	Select the 'Link Options' tab at the bottom of the properties window pane. Under 'Section -> ROM to RAM mapped section', add the three mappings as shown opposite.	Section Section start address SU.SI.B_1.R_1.B_2.R_2.B.R/4.C_1.C_2.C.C\$*.D*.W*.L.P*/FF The specified section that outputs externally defined symbols to the file Section alignment Section alignment(0) ROM to RAM mapped section ROM to RAM mapped section[3]
	These settings are easily added by clicking the button '' and pasting the following text into the dialog: R 1=R_1 2=R_2 This ensures that the linker assigns RAM rather than ROM addresses to C variables. Click 'OK'	Text Edit     X       Iext:     D=R       D_1=R_1     D_2=R_2       0_2=R_2     0K       OK     Help



#### RSKRX66T





#### RSKRX66T

### 5. Completing the Tutorial Project

•	All of the sample code projects contained in this RSK are configured with three Build Modes; 'DefaultBuild', 'Debug' and 'Release'. 'Release' is created in the same way as above; by duplicating 'Default Build'. 'Release' Build Mode leaves code optimisation turned on and removes debug information from the output file. To remove debug information from the 'Release' Build Mode, in the 'CC- RX Property' window, select the 'Common Options' tab at the bottom of the window pane. For the 'Outputs debugging information' option, select 'No(-nodebug). Reset the Build Mode back to 'Debug' using the 'Build Mode' pull- down control. From the menus, select 'File -> Save All' to save all project settings.	System include paths     System include     Macro definition     Outputs debugging information     Optimization level     Outputs additional information for inter-module optimization     Optimization level     Outputs additional information for inter-module optimization     No     Optimization type     Optimizes     Outputs a source list file     No(nolistfile)     Frequently Used Options(for Assemble)     Frequently Used Options(for Link)     Using libraries     Outputs debugging information     Yes (Outputs     Optimization type     Yes (Outputs	on[0] ) n emphasis on code size(size) (0) to the output file)(-DEBug) to coutput file)(-DEBug) to <output file="" name="">.dbg file)(-SDebug)</output>
•	Additional Folders         Before new source files are added to the project, we will create two additional folders in the CS+ Project Tree.         In the Project Tree pane, right-click the SC_Tutorial project and select 'Add -> Add New Category'.	Project Tree       4 ×         2       3       2         3       SC Tutorial (Project)       1         4       RSF566TEAxFP (M)       Rebuild SC_Tutorial         5       Smart Configuration       1         4       CC-RX (Build Tool       1         5       RX Simulator (Deb)       1         6       SC_Tutorial       1         7       File       1         6       Sc_Tutorial       1         7       CC-RX (Build Tool       1         10       Open Folder with Explorer       1         11       Windows Explorer Menu       1         12       Save Project Menu       1         13       Save Project and Development Tools as Package       1         14       Paste       Ctrl+         15       Property       1	
•	Rename the newly-created 'New Category' folder to 'C Source Files'. Repeat these steps to create a new category folder for 'Dependencies'.	Project Tree       Image: Constraint (Project)         Image: Constraint (Project)       Image: Constraint (Project)         Image: Project (Project)       Image: Project (Pro	



## 5.3 LCD Code Integration

API functions for the Okaya LCD display are provided with the RSK. Refer to the Tutorial project folder created according to the Quick Start Guide procedure. Locate the files ascii.h, r\_okaya\_lcd.h, ascii.c, and r\_okaya\_lcd.c in this folder. Copy these files into the C:\Workspace\SC\_Tutorial\src folder.





#### RSKRX66T

• Similarly, add 'ascii.h' and 'r_okaya_lcd.h' to the	
'Dependencies' folder.	
	Organize  Vew folder
Note: Select the Header file (* .h; * hpp; * .inc).	★ Quick access smc_gen
	This PC
	Desktop  r_okaya_lcd.h  Documents
	Jownloads
	Music     Fictures
	Videos
	teal Local Disk (C;)
	💣 Network
	<pre></pre>
	File name: "r_okaya_lcd.h" "ascii.h" V Header file (".h; *.hpp; *.inc) V
	Open Cancel
Make sure the project tree is the same as the	Project Tree 🛛 📮 🗙
screen shot.	
	2 🕐 🙎 🔳
	□
	R5F566TEAxFP (Microcontroller)
	Smart Configurator (Design Tool)
	CC-RX (Build Tool)
	RX Simulator (Debug Tool)
	Program Analyzer (Analyze Tool)
	Build tool generated files
	Smart Configurator
	🖕 🛄 C Source Files
	ascii.c
	SC_Tutorial.c
	Dependencies
	ascii.h
	r_okaya_lcd.h



Code must be inserted in to the user code area in many files in this project, in the areas delimited by comments as follows:

```
/* Start user code for _xxxxx_. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */
```

Where \_xxxx\_ depends on the particular area of code, i.e. 'function' for insertion of user functions and prototypes, 'global' for insertion of user global variable declarations, or 'include' for insertion of pre-processor include directives. User code inserted inside these comment delimiters is protected from being overwritten by Smart Configurator, if the user needs to subsequently change any of the Smart Configurator-generated code.

In the CS+ Project Tree, expand the 'src/smc\_gen/general' folder and open the file 'r\_cg\_userdefine.h' by double-clicking on it. Insert the following #defines in between the user code delimiter comments as shown below.

/\* Start user code for function. Do not edit comment generated here \*/



/\* End user code. Do not edit comment generated here \*/

In the CS+ Project Tree, expand the 'C Source Files' folder and open the file 'SC\_Tutorial.c' by double-clicking on it. Add header files above the 'main' function as shown below.



Scroll down to the 'main' function and insert the highlighted code as shown below into the beginning of the 'main' function:





#### 5.3.1 SPI Code

The Okaya LCD display is driven by the SPI Master that was configured using Smart Configurator in §4.5.6. In the CS+ Project Tree, expand the 'Smart Configurator/Config\_SCI6' and open the file 'Config\_SCI6.h' by double-clicking on it. Insert the following code in the user code area at the end of the file:

```
/* Start user code for function. Do not edit comment generated here */
/* Exported functions used to transmit a number of bytes and wait for completion */
MD_STATUS R_SCI6_SPIMasterTransmit(uint8_t * const tx_buf, const uint16_t tx_num);
```

 $/\,{}^{\star}$  End user code. Do not edit comment generated here  ${}^{\star}/$ 

Now, open the Config\_SCI6\_user.c file and insert the following code in the user area for global:

```
/* Start user code for global. Do not edit comment generated here */
```

/\* Flag used locally to detect transmission complete \*/
static volatile uint8\_t gs\_sci6\_txdone;

/\* End user code. Do not edit comment generated here \*/

Insert the following code in the transmit end call-back function for SCI6:

```
static void r_Config_SCI6_callback_transmittend(void)
```

```
/* Start user code for r_Config_SCI6_callback_transmitend. Do not edit comment generated here */
```

gs sci6 txdone = TRUE;

```
/* End user code. Do not edit comment generated here */
```

Now insert the following function in the user code area at the end of the file:

/\* Start user code for adding. Do not edit comment generated here \*/

```
* Function Name: R SCI6 SPIMasterTransmit
 Description % \overline{\mathcal{T}} : This function sends SPI6 data to slave device.
 Arguments : tx_buf -
              transfer buffer pointer
          tx_num -
              buffer size
* Return Value : status -
             MD_OK or MD_ARGERROR
MD STATUS R SCI6 SPIMasterTransmit (uint8 t * const tx buf,
                          const uint16 t tx num)
  MD STATUS status = MD OK;
  /* Clear the flag before initiating a new transmission */
  gs_sci6_txdone = FALSE;
  /* Send the data using the API */
  status = R Config SCI6 SPI Master Send(tx buf, tx num);
  /* Wait for the transmit end flag */
  while (FALSE == gs sci6 txdone)
  {
     /* Wait */
  }
  return (status);
}
* End of function R SCI6 SPIMasterTransmit
```

This function uses the transmit end callback function to perform flow control on the SPI transmission to the LCD, and is used as the main API call in the LCD code module.

#### 5.3.2 CMT Code

The LCD code needs to insert delays to meet the timing requirements of the display module. This is achieved using the dedicated timer which was configured using Smart Configurator in §4.5.2. In the CS+ Project Tree, expand the 'Smart Configurator\Config\_CMT0\Config\_CMT0.h' and insert the following code in the user area for function at the end of the file:

/\* Start user code for function. Do not edit comment generated here \*/

void R\_CMT\_MsDelay(const uint16\_t millisec);

/\* End user code. Do not edit comment generated here \*/

Open the file 'Config\_CMT0\_user.c' and insert the following code in the user area for global at the beginning of the file:

/\* Start user code for global. Do not edit comment generated here \*/

static volatile uint8\_t gs\_one\_ms\_delay\_complete = FALSE;

/\* End user code. Do not edit comment generated here \*/

Scroll down to the r\_Config\_CMT0\_cmi0\_interrupt function and insert the following line in the user code area:

```
static void r_Config_CMT0_cmi0_interrupt(void)
```

/\* Start user code for r\_Config\_CMT0\_cmi0\_interrupt. Do not edit comment generated here \*/

gs\_one\_ms\_delay\_complete = TRUE;

/\* End user code. Do not edit comment generated here \*/

}

Then insert the following function in the user code area at the end of the file:

/\* Start user code for adding. Do not edit comment generated here \*/

```
* Function Name: R CMT MsDelay
* Description : Uses CMTO to wait for a specified number of milliseconds
* Arguments
         : uint16 t millisecs, number of milliseconds to wait
* Return Value : None
         *******
void R CMT MsDelay (const uint16 t millisec)
{
  uint16 t ms count = 0;
  do
  {
    R Config CMT0 Start();
    while (FALSE == gs one ms delay complete)
     {
       /* Wait */
    R Config CMT0 Stop();
     gs_one_ms_delay_complete = FALSE;
     ms count++;
  } while (ms count < millisec);</pre>
End of function R CMT MsDelay
```

Select 'Build Project' from the 'Build' menu, or press F7. CS+ will build the project with no errors.

The project may now be run using the debugger as described in §6. The program will display 'RSKRX66T Tutorial Press Any Switch' on 3 lines in the LCD display.

### 5.4 Switch Code Integration

API functions for user switch control are provided with the RSK. Refer to the Tutorial project folder created according to the Quick Start Guide procedure. Locate the files rskrx66tdef.h, r\_rsk\_switch.h and r\_rsk\_switch.c in this folder. Copy these files into the C:\Workspace\SC\_Tutorial\src folder. Add these three files into the project in the same way as the LCD files.

The switch code uses interrupt code in the files Config\_ICU.c, Config\_ICU\_user.c and Config\_ICU.h and timer code in the files Config\_ICU.c, Config\_ICU\_user.c, Config\_CMT1.h, Config\_CMT1.c, Config\_CMT1\_user.c, Config\_CMT2.h, Config\_CMT2.c, and Config\_CMT2\_user.c, as described in §4.5.2 and §4.5.3. It is necessary to provide additional user code in these files to implement the switch press/release detection and de-bouncing required by the API functions in r\_rsk\_switch.c.

#### 5.4.1 Interrupt Code

In the CS+ Project Tree, expand the 'Smart Configurator/Config\_ICU' folder and open the file 'Config\_ICU.h' by double-clicking on it. Insert the following code in the user code area at the end of the file:

/\* Start user code for function. Do not edit comment generated here \*/

/\* Function prototypes for detecting and setting the edge trigger of ICU\_IRQ \*/
uint8 t R ICU IRQIsFallingEdge(const uint8 t irq no);
void R\_ICU\_IRQSetFallingEdge(const uint8\_t irq\_no, const uint8\_t set\_f\_edge);
void R\_ICU\_IRQSetRisingEdge(const uint8\_t irq\_no, const uint8\_t set\_r\_edge);

 $/\,{}^{\star}$  End user code. Do not edit comment generated here  ${}^{\star}/$ 

Now, open the Config\_ICU.c file and insert the following code in the user code area at the end of the file:

/\* Start user code for adding. Do not edit comment generated here \*/

```
* Function Name: R ICU IRQIsFallingEdge
^{\star} Description \, : This function returns 1 if the specified ICU IRQ is set to
* falling eage triggered, triggered, 0 if not
* Return Value : 1 if falling edge triggered, 0 if not
                                          uint8 t R ICU IRQIsFallingEdge (const uint8 t irq no)
   uint8 t falling edge trig = 0x0;
   if (ICU.IRQCR[irq no].BYTE & 04 ICU IRQ EDGE FALLING)
   {
      falling edge trig = 1;
   }
   return (falling_edge_trig);
}
* End of function R ICU IRQIsFallingEdge
```



```
* Function Name: R ICU IRQSetFallingEdge
 Description : This function sets/clears the falling edge trigger for the
*
           specified ICU IRQ.
* Arguments
        : uint8_t irq_no
           uint8 t set f edge, 1 if setting falling edge triggered, 0 if
           clearing
* Return Value : None
               void R ICU IRQSetFallingEdge (const uint8 t irq no, const uint8 t set f edge)
  if (1 == set_f_edge)
  {
     ICU.IRQCR[irq_no].BYTE |= _04_ICU_IRQ_EDGE_FALLING;
  }
  else
  {
     ICU.IRQCR[irq no].BYTE &= (uint8 t) ~ 04 ICU IRQ EDGE FALLING;
  }
*
 End of function R ICU IRQSetFallingEdge
                           *******
* Function Name: R ICU IRQSetRisingEdge
* Description : This function sets/clear the rising edge trigger for the
           specified ICU IRQ.
         : uint8 t irq no
*
 Arguments
          uint8_t set_r_edge, 1 if setting rising edge triggered, 0 if
           clearing
* Return Value : None
               void R_ICU_IRQSetRisingEdge (const uint8_t irq_no, const uint8_t set_r_edge)
  if (1 == set_r_edge)
  {
     ICU.IRQCR[irq_no].BYTE |= _08_ICU_IRQ_EDGE_RISING;
  }
  else
  {
     ICU.IRQCR[irq no].BYTE &= (uint8 t) ~ 08 ICU IRQ EDGE RISING;
  }
}
* End of function R ICU IRQSetRisingEdge
                           ****
```

/\* End user code. Do not edit comment generated here \*/



Open the Config\_ICU\_user.c file and insert the following code in the user code area for include near the top of the file:

/\* Start user code for include. Do not edit comment generated here \*/
/\* Defines switch callback functions required by interrupt handlers \*/
#include "r\_rsk\_switch.h"

/\* End user code. Do not edit comment generated here \*/

In the same file insert the following code in the user code area inside the function r\_Config\_ICU\_irq0\_interrupt:

/\* Start user code for r\_Config\_ICU\_irq0\_interrupt. Do not edit comment generated here \*/
/\* Switch 1 callback handler \*/
R\_SWITCH\_IsrCallback1();

 $/\,{}^{\star}$  End user code. Do not edit comment generated here  $\,{}^{\star}/$ 

In the same file insert the following code in the user code area inside the function r\_Config\_ICU\_irq9\_interrupt:

/\* Start user code for r\_Config\_ICU\_irq9\_interrupt. Do not edit comment generated here \*/
/\* Switch 2 callback handler \*/
R\_SWITCH\_IsrCallback2();
/\* End user code. Do not edit comment generated here \*/



#### 5.4.2 De-bounce Timer Code

In the Project Tree, expand the 'Smart Configurator\Config\_CMT1' folder and open the 'Config\_CMT1\_user.c' file and insert the following code in the user code area for include near the top of the file:

/\* Start user code for include. Do not edit comment generated here \*/
/\* Defines switch callback functions required by interrupt handlers \*/
#include "r\_rsk\_switch.h"

 $/\star$  End user code. Do not edit comment generated here  $\star/$ 

In the 'Config\_CMT1\_user.c' file insert the following code in the user code area inside the function r\_Config\_CMT1\_cmi1\_interrupt:

/\* Start user code for r\_Config\_CMT1\_cmi1\_interrupt. Do not edit comment generated here \*/
/\* Stop this timer - we start it again in the de-bounce routines \*/
R\_Config\_CMT1\_Stop();
/\* Call the de-bounce call back routine \*/
R\_SWITCH\_DebounceIsrCallback();
/\* End user code. Do not edit comment generated here \*/

In the Project Tree, expand the 'Smart Configurator\Config\_CMT2' folder and open the 'Config\_CMT2\_user.c' file and insert the following code in the user code area for include near the top of the file:

/\* Start user code for include. Do not edit comment generated here \*/
/\* Defines switch callback functions required by interrupt handlers \*/
#include "r rsk switch.h"

/\* End user code. Do not edit comment generated here \*/

Open the 'Config\_CMT2\_user.c' file and insert the following code in the user code area inside the function r\_Config\_CMT2\_cmi2\_interrupt:

/\* Start user code for r\_Config\_CMT2\_cmi2\_interrupt. Do not edit comment generated here \*/
/\* Stop this timer - we start it again in the de-bounce routines \*/
R\_Config\_CMT2\_Stop();
/\* Call the de-bounce call back routine \*/
R\_SWITCH DebounceIsrCallback();

/\* End user code. Do not edit comment generated here \*/



#### 5.4.3 Main Switch and ADC Code

In this part of the tutorial we add the code to act on the switch presses to activate A/D conversions and display the result on the LCD. In §4.5.7 we configured the ADC to be triggered from the ADTRG0# pin, SW3. In this code, we also perform software triggered A/D conversion from the user switches SW1 and SW2, by reconfiguring the ADC trigger source on-the-fly once an SW1 or SW2 press is detected.

In the CS+ Project Tree, expand the 'Smart Configurator\general' folder and open the file 'r\_cg\_userdefine.h' by double-clicking on it. Insert the following code the user code area, resulting in the code shown below

/\* Start user code for function. Do not edit comment generated here \*/

#define TRUE (1)
#define FALSE (0)
extern volatile uint8 t g adc trigger;

/\* End user code. Do not edit comment generated here \*/

In the Project Tree, expand the 'C Source Files' folder and Open the file 'SC\_Tutorial.c' and add the highlighted code, resulting in the code shown below:

#include "r\_smc\_entry.h"
#include "r\_okaya\_lcd.h"
#include "r\_og\_userdefine.h"
#include "Config S12AD0.h"
#include "r\_rsk\_switch.h"
/\* Variable for flagging user requested ADC conversion \*/
volatile uint8\_t g\_adc\_trigger = FALSE;
/\* Prototype declaration for cb switch\_press \*/
static void cb\_switch\_press (void);
/\* Prototype declaration for get\_adc \*/
static uint16\_t get\_adc(void);
/\* Prototype declaration for lcd display adc \*/
static void lcd\_display\_adc (const uint16\_t adc\_result);



Next add the highlighted code below in the main function and the code inside the while loop, resulting in the code shown below:

```
void main(void)
{
    /* Initialize th<mark>e switch module */</mark>
    R SWITCH Init();
    /* Set the call back function when SW1 or SW2 is pressed */
    R SWITCH SetPressCallback(cb_switch_press);
    /* Initialize the debug LCD */
    R LCD Init();
    /* Displays the application name on the debug LCD */
    R LCD Display(0, (uint8 t *)" RSKRX66T ");
R LCD Display(1, (uint8 t *)" Tutorial ");
R_LCD_Display(2, (uint8 t *)" Press Any Switch ");
    /* Start the A/D converter */
R_Config_S12AD0_Start();
    while (1U)
    {
         uint16 t adc result;
         /* Wait for user requested A/D conversion flag to be set (SW1 or SW2) */
         if (TRUE == g_adc_trigger)
         {
               * Call the function to perform an A/D conversion */
             adc result = get adc();
             /* Display the result on the LCD */
             lcd_display_adc(adc_result);
             /* Reset the flag */
             g adc trigger = FALSE;
         /* SW3 is directly wired into the ADTRGOn pin so will
            cause the interrupt to fire */
         else if (TRUE == g_adc_complete)
             /* Get the result of the A/D conversion */
             R Config S12AD0 Get ValueResult(ADCHANNEL0, &adc result);
             /* Display the result on the LCD */
             lcd_display_adc(adc_result);
             /* Reset the flag */
             g adc complete = FALSE;
        else
         {
             /* do nothing */
         }
    }
}
```

Then add the definition for the switch call-back, get\_adc and lcd\_display\_adc functions adding at the below of the main function, as shown below:



```
/* set the flag indicating a user requested A/D conversion is required */
     g adc trigger = TRUE;
      /* Clear flag */
     g switch flag = 0x0;
   }
}
* End of function cb switch press
                     * Function Name : get_adc
* Description : Reads the ADC result, converts it to a string and displays
            it on the LCD panel.
        : none
* Argument
* Return value : uint16_t adc value
                           *****
static uint16 t get adc (void)
{
  /* A variable to retrieve the adc result */
  uint16 t adc result;
   /* Stop the A/D converter being triggered from the pin ADTRGOn */
  R_Config_S12AD0_Stop();
   /* Start a conversion */
  R S12AD0 SWTriggerStart();
   /* Wait for the A/D conversion to complete */
  while (FALSE == g_adc_complete)
   {
     /* Wait */
  }
  /* Stop conversion */
  R S12AD0 SWTriggerStop();
   /* Clear ADC flag */
  g adc complete = FALSE;
  R Config S12AD0 Get ValueResult(ADCHANNEL0, &adc result);
   /* Set AD conversion start trigger source back to ADTRGOn pin */
  R Config S12AD0 Start();
  return (adc_result);
}
* End of function get adc
                * Function Name : lcd display adc
* Description : Converts add result to a string and displays
            it on the LCD panel.
        : uint16_t adc result
* Argument
* Return value : none
                static void lcd_display_adc (const uint16_t adc_result)
   /* Declare a temporary variable */
  uint8 t a;
  /* Declare temporary character string */
  char lcd buffer[11] = " ADC: XXXH";
  /\star Convert ADC result into a character string, and store in the local.
    Casting to ensure use of correct data type. */
  a = (uint8_t) ((adc_result & 0x0F00) >> 8);
  lcd buffer[6] = (char) ((a < 0x0A) ? (a + 0x30) : (a + 0x37));
  a = (uint8_t) ((adc result & 0x00F0) >> 4);
```

In the Project Tree, expand the 'Smart Configurator\Config\_S12AD0' folder and open the file 'Config\_S12AD0.h' by double-clicking on it. Insert the following code in the user code area for function, resulting in the code shown below:

```
/* Start user code for function. Do not edit comment generated here */
```

```
/* Flag indicates when A/D conversion is complete */
extern volatile uint8_t g_adc_complete;
/* Functions for starting and stopping software triggered A/D conversion */
void R_S12AD0_SWTriggerStart(void);
void R_S12AD0_SWTriggerStop(void);
```

/\* End user code. Do not edit comment generated here \*/

Open the file 'Config\_S12AD0.c' by double-clicking on it. Insert the following code in the user code area for adding at the end of the file, resulting in the code shown below:

```
/* Start user code for adding. Do not edit comment generated here */
```

/**************************************	* * * * * * * * * * * * * * * * * * * *
<pre>* Function Name: R S12AD0 SWTriggerStart * Description : This function starts the AD0 converter * Arguments : None * Return Value : None ************************************</pre>	•
<pre>void R_S12AD0_SWTriggerStart(void)</pre>	
<pre>{     IR(S12AD, S12ADI) = 0U;     IEN(S12AD, S12ADI) = 1U;     S12AD.ADCSR.BIT.ADST = 1U; }</pre>	
/*************************************	
/**************************************	* * * * * * * * * * * * * * * * * * * *
<pre>* Function Name: R S12AD0 SWTriggerStop * Description : This function stops the AD0 converter.</pre>	
* Arguments : None * Return Value : None	
**************************************	***************************/
<pre>{     S12AD.ADCSR.BIT.ADST = 0U;     IEN(S12AD, S12ADI) = 0U;     IR(S12AD, S12ADI) = 0U; }</pre>	
/*************************************	

/\* End user code. Do not edit comment generated here \*/



Open the file Config\_S12AD0\_user.c and insert the following code in the user code area for global, resulting in the code shown below:

/\* Start user code for global. Do not edit comment generated here \*/
/\* Flag indicates when A/D conversion is complete \*/
volatile uint8\_t g\_adc\_complete;

 $/\star$  End user code. Do not edit comment generated here  $\star/$ 

Insert the following code in the user code area of the r\_Config\_S12AD0\_interrupt function, resulting in the code shown below:

```
static void r_Config_S12AD0_interrupt(void)
{
    /* Start user code for r_Config_S12AD0_interrupt. Do not edit comment generated here */
    g_adc_complete = TRUE;
    /* End user code. Do not edit comment generated here */
}
```

Select 'Build Project' from the 'Build' menu, or press F7. CS+ will build the project with no errors.

The project may now be run using the debugger as described in §6. When any switch is pressed, the program will perform an A/D conversion of the voltage level on the ADPOT line and display the result on the LCD panel. Return to this point in the SC\_Tutorial to add the UART user code.



### 5.5 Debug Code Integration

API functions for trace debugging via the RSK serial port are provided with the RSK. Refer to the Tutorial project folder created according to the Quick Start Guide procedure. Locate the files r\_rsk\_debug.h and r\_rsk\_debug.c in this folder. Copy these files into the C:\Workspace\SC\_Tutorial\src folder. Add these two files into the project in the same way as the LCD files.

In the r\_rsk\_debug.h file, ensure the following macro definition is included:

```
/* Macro for definition of serial debug transmit function - user edits this */
#define SERIAL_DEBUG_WRITE (R_SCI11_AsyncTransmit)
```

This macro is referenced in the r\_rsk\_debug.c file and allows easy re-direction of debug output if a different debug interface is used.

### 5.6 UART Code Integration

#### 5.6.1 SCI Code

In the CS+ Project Tree, expand the 'Smart Configurator\Config\_SCI11' folder and open the file 'Config\_SCI11.h' by double-clicking on it. Insert the following code in the user code area at the end of the file:

/\* Start user code for function. Do not edit comment generated here \*/

```
/* Exported functions used to transmit a number of bytes and wait for completion */
MD_STATUS R_SCI11_AsyncTransmit(uint8_t * const tx_buf, const uint16_t tx_num);
```

/\* Character is used to receive key presses from PC terminal \*/
extern uint8\_t g\_rx\_char;

 $/\ast$  End user code. Do not edit comment generated here  $\ast/$ 

Open the file 'Config\_SCI11\_user.c'. Insert the following code in the user area for global near the beginning of the file:

/\* Start user code for global. Do not edit comment generated here \*/

/\* Global used to receive a character from the PC terminal \*/
uint8\_t g\_rx\_char;

/\* Flag used locally to detect transmission complete \*/
static volatile uint8\_t gs\_scil1\_txdone;

/\* End user code. Do not edit comment generated here \*/

In the same file, insert the following code in the user code area inside the r\_Config\_SCI11\_callback\_transmittend function:

static void r\_Config\_SCI11\_callback\_transmitend (void)
{
 /\* Start user code for r\_Config\_SCI11\_callback\_transmitend. Do not edit comment generated here \*/
 gs\_scill\_txdone = TRUE;
 /\* End user code. Do not edit comment generated here \*/
}



In the same file, insert the following code in the user code area inside the r\_Config\_SCI11\_callback\_receiveend function:

```
static void r_Config_SCI11_callback_receiveend(void)
{
    /* Start user code for r_Config_SCI11_callback_receiveend. Do not edit comment generated here */
    /* Check the contents of g rx char */
    if (('c' == g_rx_char) || ('C' == g_rx_char))
    {
        g_adc_trigger = TRUE;
    }
    /* Set up SCI11 receive buffer and callback function again */
    R_Config_SCI11_Serial_Receive((uint8_t *)&g_rx_char, 1);
    /* End user code. Do not edit comment generated here */
}
```

At the end of the file, in the user code area for adding, add the following function definition:

```
Function Name: R SCI11 AsyncTransmit
 Description : This function sends SCI11 data and waits for the transmit end flag.
ш.
* Arguments : tx_buf -
              transfer buffer pointer
          tx_num -
4
             buffer size
* Return Value : status -
MD_STATUS R_SCI11_AsyncTransmit(uint8 t * const tx buf, const uint16 t tx num)
{
  MD STATUS status = MD OK;
  /* Clear the flag before initiating a new transmission ^{\star/}
  gs_scill_txdone = FALSE;
  /* Send the data using the API */
  status = R Config SCI11 Serial Send(tx buf, tx num);
  /* Wait for the transmit end flag */
  while (FALSE == gs_scill_txdone)
  {
     /* Wait */
  }
  return (status);
}
* End of function R_SCI11_AsyncTransmit
    *****
```



#### 5.6.2 Main UART code

In the Project Tree, expand the 'C Source Files' folder and open the file 'SC\_Tutorial.c'. Add the following declaration to above the 'main' function:

```
#include "r_smc_entry.h"
#include 'r_okaya_lcd.h"
#include "r_cg_userdefine.h"
#include "Config_S12AD0.h"
#include "r rsk switch.h"
#include "r rsk debug.h"
#include "Config_SCI11.h"
/* Variable for flagging user requested ADC conversion */
volatile uint8 t g adc trigger = FALSE;
/* Prototype declaration for cb_switch_press */
static void cb switch press (void);
/* Prototype declaration for get_adc */
static uint16_t get_adc(void);
/* Prototype declaration for lcd display adc */
static void lcd_display_adc (const uint16_t adc_result);
/* Prototype declaration for uart display adc */
static void uart_display_adc(const uint8_t gs_adc_count, const uint16_t adc_result);
/* Variable to store the A/D conversion count for user display */
static uint8_t gs_adc_count = 0;
```

Add the following highlighted code to the main function:

```
void main (void)
{
    /* Initialize the switch module */
    R SWITCH Init();
    /* Set the call back function when SW1 or SW2 is pressed */
    R_SWITCH_SetPressCallback(cb_switch_press);
     /* Initialize the debug LCD */
    R LCD Init();
     /* Displays the application name on the debug LCD */
    R_LCD_Display(0, (uint8_t *)" RSKRX66T ");
R_LCD_Display(1, (uint8_t *)" Tutorial ");
R_LCD_Display(2, (uint8_t *)" Press Any Switch ");
     /* Start the A/D converter */
    R Config S12AD0 Start();
    /* Set up SCI11 receive buffer and callback function */
R_Config_SCI11_Serial_Receive((uint8_t *)&g_rx_char, 1);
     /* Enable SCI11 operations */
    R_Config_SCI11_Start();
    while (1U)
     {
         uint16 t adc result;
         /* Wait for user requested A/D conversion flag to be set (SW1 or SW2) */
         if (TRUE == g_adc_trigger)
         {
              /* Call the function to perform an A/D conversion */
              adc result = get adc();
              /* Display the result on the LCD */
              lcd display adc(adc result);
              /* Increment the gs_adc_count */
              if (16 == (++gs adc count))
              {
                   gs adc count = 0;
              }
              /* Send the result to the UART */
```

RENESAS

}

```
uart display adc(gs adc count, adc result);
        /* Reset the flag */
        g_adc_trigger = FALSE;
    /* SW3 is directly wired into the ADTRGOn pin so will
      cause the interrupt to fire */
    else if (TRUE == g_adc_complete)
    {
        /* Get the result of the A/D conversion */
        R Config S12AD0 Get ValueResult(ADCHANNELO, &adc result);
        /* Display the result on the LCD */
        lcd_display_adc(adc_result);
        /* Increment the gs adc count */
        if (16 == (++gs adc count))
        {
            gs adc count = 0;
        }
        /* Send the result to the UART */
        uart_display_adc(gs adc count, adc result);
        /* Reset the flag */
        g_adc_complete = FALSE;
    }
   else
   {
        /* do nothing */
    }
}
```

Then, add the following function definition in the end of the file:

```
* Function Name : uart_display_adc
Description : Converts add result to a string and sends it to the UART1.
* Argument
           : uint8_t : gs_adc_count
              uint16 t: adc result
* Return value : none
               static void uart display adc (const uint8 t gs adc count, const uint16 t adc result)
{
   /* Declare a temporary variable */
  char a:
   /* Declare temporary character string */
  static char uart buffer[] = "ADC xH Value: xxxH\r\n";
   /* Convert ADC result into a character string, and store in the local.
    Casting to ensure use of correct data type. */
  a = (char)(gs_adc_count & 0x000F);
uart_buffer[4] = (char)((a < 0x0A) ? (a + 0x30) : (a + 0x37));</pre>
   a = (char) ((adc_result \& 0x0F00) >> 8);
   uart buffer[14] = (char) ((a < 0x0A) ? (a + 0x30) : (a + 0x37));
   a = (char)((adc_result & 0x00F0) >> 4);
  uart buffer[15] = (char) ((a < 0x0A) ? (a + 0x30) : (a + 0x37));
  a = (char) (adc result & 0x000F);
   uart buffer[16] = (char) ((a < 0x0A) ? (a + 0x30) : (a + 0x37));
   /* Send the string to the UART */
   R DEBUG Print(uart buffer);
}
                              * End of function uart_display_adc
                        ********
```

Select 'Build Project' from the 'Build' menu, or press F7. CS+ will build the project with no errors.



The project may now be run using the debugger as described in §6. Connect the RSK G1CUSB0 port to a USB port on a PC. If this is the first time the RSK has been connected to the PC then a device driver will be installed automatically. Open Device Manager, the virtual COM port will now appear under 'Port (COM & LPT)' as 'RSK USB Serial Port (COMx)', where x is a number.

Open a terminal program, such as HyperTerminal, on the PC with the same settings as for SCI11 (see §4.5.5). When any switch is pressed, or when 'c' is sent via the COM port, the program will perform an A/D conversion of the voltage level on the ADPOT line and display the result on the LCD panel and send the result to the PC terminal program via the SCI11. Return to this point in the SC\_Tutorial to add the LED user code.

### 5.7 LED Code Integration

In the Project Tree, expand the 'C Source Files' folder and open the file 'SC\_Tutorial.c'. Add the following declaration to the above the 'main' function:

```
#include "r_smc_entry.h"
#include "r_okaya_lcd.h"
#include "r cg userdefine.h"
#include "Config_S12AD0.h"
#include "r_rsk_switch.h"
#include "r_rsk_debug.h"
#include "Config SCI11.h"
#include "rskrx66tdef.h"
/* Variable for flagging user requested ADC conversion */
volatile uint8 t g adc trigger = FALSE;
/* Prototype declaration for cb switch press */
static void cb switch press (void);
/* Prototype declaration for get adc */
static uint16 t get adc(void);
/* Prototype declaration for lcd display adc */
static void lcd display adc (const uint16 t adc result);
/* Prototype declaration for uart_display_adc */
static void uart_display_adc(const uint8_t gs_adc_count, const uint16_t adc_result);
/* Variable to store the A/D conversion count for user display */
static uint8_t gs_adc_count = 0;
/* Prototype declaration for led display count */
```

static void led\_display\_count(const uint8\_t count);

Add the following highlighted code to the main function:

```
void main(void)
{
    /* Initialize the switch module */
    R_SWITCH_Init();
    /* Set the call back function when SW1 or SW2 is pressed */
    R_SWITCH_SetPressCallback(cb_switch_press);
    /* Initialize the debug LCD */
    R_LCD_Init();
    /* Displays the application name on the debug LCD */
    R_LCD_Display(0, (uint8_t *)" RSKRX66T ");
    R_LCD_Display(1, (uint8_t *)" Tutorial ");
    R_LCD_Display(2, (uint8_t *)" Press Any Switch ");
    /* Start the A/D converter */
    R_Config_S12AD0_Start();
```



```
/* Set up SCI11 receive buffer and callback function */
R_Config_SCI11_Serial_Receive((uint8_t *)&g rx char, 1);
/* Enable SCI11 operations */
R Config SCI11 Start();
while (1U)
{
    uint16 t adc result;
    /* Wait for user requested A/D conversion flag to be set (SW1 or SW2) */
    if (TRUE == g adc trigger)
    {
        /* Call the function to perform an A/D conversion */
        adc_result = get_adc();
        /* Display the result on the LCD */
        lcd display adc(adc result);
        /* Increment the gs_adc_count and display using the LEDs */
        if (16 == (++gs_adc_count))
        {
            qs adc count = 0;
        led_display_count(gs_adc_count);
        /* Send the result to the UART */
        uart_display_adc(gs_adc_count, adc_result);
/* Reset the flag */
        g_adc_trigger = FALSE;
    }
    /* SW3 is directly wired into the ADTRGOn pin so will
      cause the interrupt to fire */
    else if (TRUE == g_adc_complete)
    {
        /* Get the result of the A/D conversion */
        R_Config_S12AD0_Get_ValueResult(ADCHANNEL0, &adc_result);
        /* Display the result on the LCD */
        lcd display adc(adc result);
        /* Increment the gs_adc_count and display using the LEDs */
        if (16 == (++gs_adc_count))
        {
            gs_adc count = 0;
        led display count(gs adc count);
        /* Send the result to the UART */
        uart_display_adc(gs_adc_count, adc_result);
        /* Reset the flag */
        g adc complete = FALSE;
    }
    else
    {
        /* do nothing */
    }
}
```

}



Then, add the following function definition in the user code area at the end of the file:

Select 'Build Project' from the 'Build' menu, or press F7. CS+ will build the project with no errors.

The project may now be run using the debugger as described in §6. The code will perform the same but now the LEDs will display the gs\_adc\_count in binary form.



# 6. Debugging the Project

•	In the 'Project Tree' pane, right-click the 'RX Simulator (Debug Tool)'. Select 'Using Debug Tool -> RX E2 Lite'.	Project Tree       # X         2       2         2       3         3       8         3       8         4       5         5       5         5       5         5       5         6       7         6       7         7       8         7       8         7       8         8       9         9       7         10       10         10       10         10       10         10       10         11       10         11       10         11       10         12       10         13       10         14       10         14       10         15       10         16       10         16       10         17       10         18       10         19       10         10       10         10       10         10       10         10       10 <td< th=""><th><ul> <li>RX E2</li> <li>RX E2 Lite</li> <li>RX E1 (Serial)</li> <li>RX E1(JTAG)</li> <li>RX E20(Serial)</li> <li>RX E20(JTAG)</li> <li>RX Simulator</li> </ul></th></td<>	<ul> <li>RX E2</li> <li>RX E2 Lite</li> <li>RX E1 (Serial)</li> <li>RX E1(JTAG)</li> <li>RX E20(Serial)</li> <li>RX E20(JTAG)</li> <li>RX Simulator</li> </ul>
•	Double-click 'RX E2 Lite (Debug Tool)' to display the debugger tool properties. Under 'Clock', change the main clock frequency to 8 MHz, Communications method 'JTAG' and operating frequency to 120MHz.' Under 'Connection with Target Board', change 'Power target from the emulator. (MAX 200mA) to 'Yes'. All other settings can remain at their	Property     RX E2 Property     Internal ROM/RAM     Size of internal ROM/RAM     Size of internal ROM/RBytes]     Size of internal RAM/(KBytes]     Size of DataFlash memory[KBytes]     Clock     Main clock frequency[MHz]     Operating frequency[MHz]     Allow changing of the clock source on writing internal flash memory     Connection with Emulator     Emulator serial No.     Connection with Target Board     Power target from the emulator.(MAX 200mA)     Interface for sumption the normal	512 80 32 EXTAL 8.0000 160.0000 No Yes USER I/F
	defaults.	Interface for supplying the power Supply voltage [V] Communications method	USER I/F 3.3 JTAG
•	Connect the E2 Lite to the PC and the RSK E1/E2 Lite connector. Connect the Pmod LCD to the PMOD1 connector. From the 'Debug' menu select 'Download' to start the debug session and download code to the target.		



# 7. Running the Smart Configurator Tutorial

## 7.1 Running the Tutorial

Once the program has been downloaded onto the RSK device, the program can be executed. Click the 'Go' button or press F5 to begin the program from the current program counter position. It is recommended that you run through the program once first, and then continue to the Tutorial manual to review the code.





## 8. Additional Information

#### **Technical Support**

For details on how to use CS+, refer to the help file by opening CS+, then selecting Help > Help Contents from the menu bar.



For information about the RX66T group microcontroller refer to the RX66T Group Hardware Manual.

For information about the RX assembly language, refer to the RX Family Software Manual.

#### **Technical Contact Details**

#### Please refer to the contact details listed in section 8 of the "Quick Start Guide".

General information on Renesas microcontrollers can be found on the Renesas website at: <u>https://www.renesas.com/</u>

#### Trademarks

All brand or product names used in this manual are trademarks or registered trademarks of their respective companies or organisations.

#### Copyright

This document may be, wholly or partially, subject to change without notice. All rights reserved. Duplication of this document, either in whole or part is prohibited without the written permission of Renesas Electronics Europe Limited.

© 2018 Renesas Electronics Europe Limited. All rights reserved. © 2018 Renesas Electronics Corporation. All rights reserved.



<b>REVISION HISTORY</b>	RSKRX66T Smart Configurator Tutorial Manual

Rev.	Date		Description		
		Page	Summary		
1.00	Sep 30, 2018	_	First Edition issued		

Renesas StarterKit Manual:Smart ConfiguratorTutorialManualPublication Date:Rev. 1.00Sep 30, 2018Published by:Renesas Electronics Corporation



#### **Renesas Electronics Corporation**

SALES OFFICES

Refer to "http://www.renesas.com/" for the latest and detailed information.

http://www.renesas.com

 Reness Electronics Corporation

 TOYOSU FORESIA. 32-224 Toyosu, Kotoku, Tokyo 135-0061, Japan

 Reness Electronics America Inc.

 1011 Murphit Ranch Road, Milpilac, CA 8005, U.S.A.

 Tei: +1-402-432-8080, Fax: +1-08-434-5351

 Reness Electronics Canada Limited

 2521 Yongs Electronics Canada Limited

 7251 Yongs Electronics Canada Limited

 7251 Yongs Electronics Europe Limited

 Dukes Meadow, Milliboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K

 Tei: +4-1905-237-2004

 Reness Electronics Europe GmbH

 Arcadiastrassa (10, 40472 Dissellooff, Germany

 Tei: +49-705-237-1004

 Reness Electronics China Co., Ltd.

 Roness Electronics (Shanghai) Co., Ltd.

 Rone A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China

 Tei: +89-70-226-0888, Fax: +86-104225-7679

 Renessa Electronics (Shanghai) Co., Ltd.

 Unit 301, Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China

 Tei: +89-70-226-0888, Fax: +86-104225-7679

 Renessa Electronics Mang Kong Limited

 Unit 101, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China

 Tei: +89-26-888, Fax: +86-104225-7679

 Renessa Electronics Mang Kong Limited

 Unit 101, Tower A, Central

RX66T Group



R20UT4153EG0100