RL78/G14

Utilising the Serial Array Unit (SAU) in Sync Mode Sample Code for e2studio

R01AN1319EG0100
Rev1.00
Oct 12, 2012

Introduction

The purpose of this Application Note is to show the user how to add the associated RL78/G14 sample code to a new or existing e2studio workspace; as well as give an explanation of what the sample code does.

The sample code provided with this Application Note demonstrates usage of the Serial Array Unit (SAU) in synchronous mode. The program runs on the RL78/G14 Renesas Starter Kit (RSK) and performs a loop back test between two SAU channels.

Target Device

RL78/G14

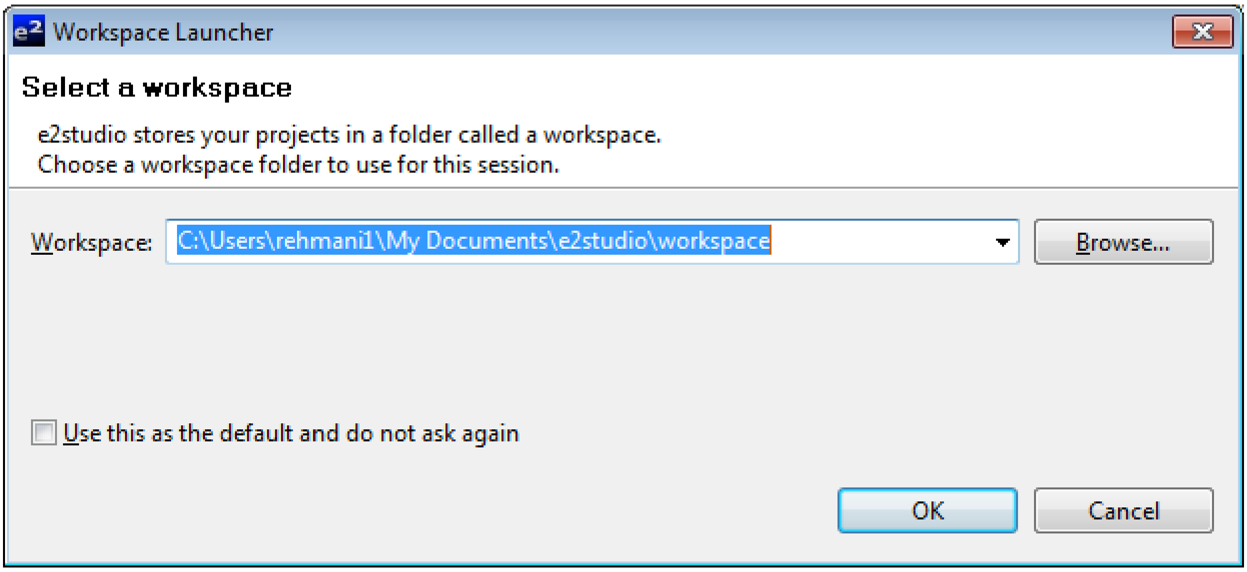
Contents

1. Installation

This section assumes e2studio IDE is already installed on the user's personal computer (PC). Create a new folder and name it as 'RSKRL78G14_Workspace'. Copy the zipped file Sync_Serial.zip, available in the Application Note package downloaded from the website, to this folder. Extract the Sync_Serial.zip file to the RSKRL78G14_Workspace folder.

2. Creating the Project Workspace

Open E2studio IDE by clicking the Windows Start button, select All Programs > Renesas Electronics e2studio > Renesas e2studio.



Workspace Launcher

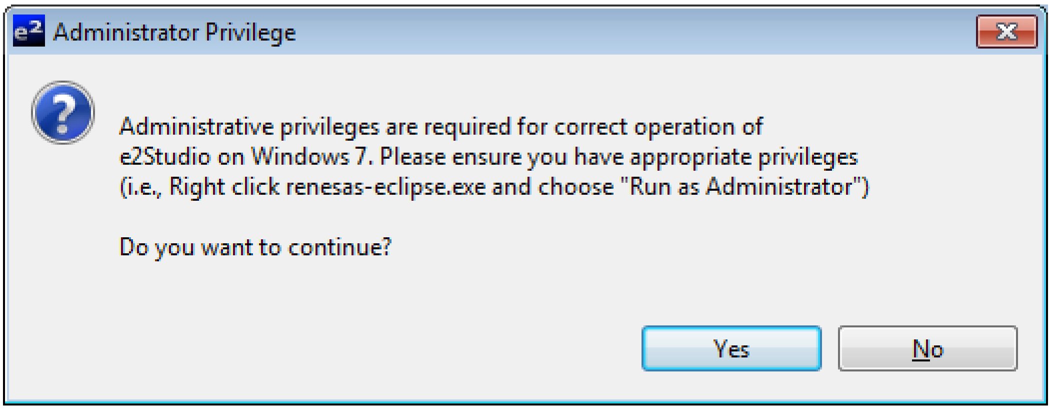
Select a workspace

e2studio stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\Users\rehmani1\My Documents\e2studio\workspace

□ Use this as the default and do not ask again

Select <OK>.

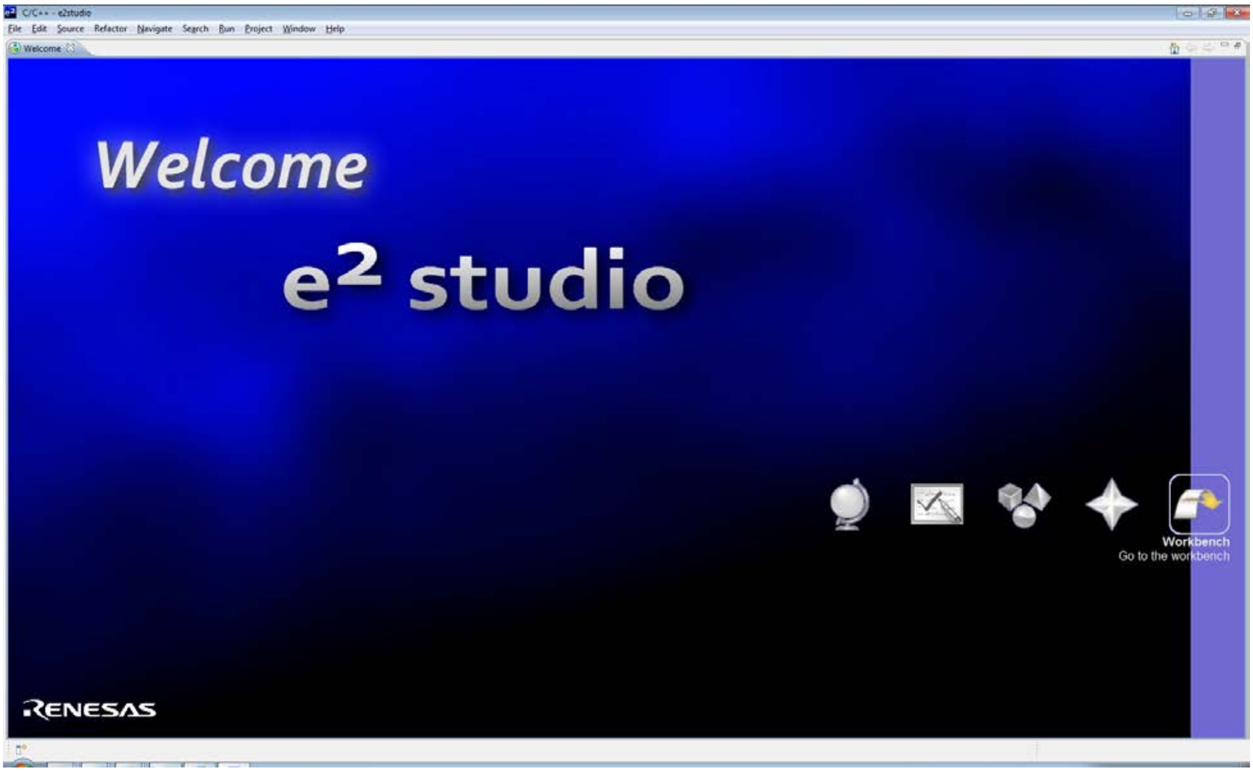


Administrator Privilege

Administrative privileges are required for correct operation of e2Studio on Windows 7. Please ensure you have appropriate privileges (i.e., Right click renesas-eclipse.exe and choose "Run as Administrator")

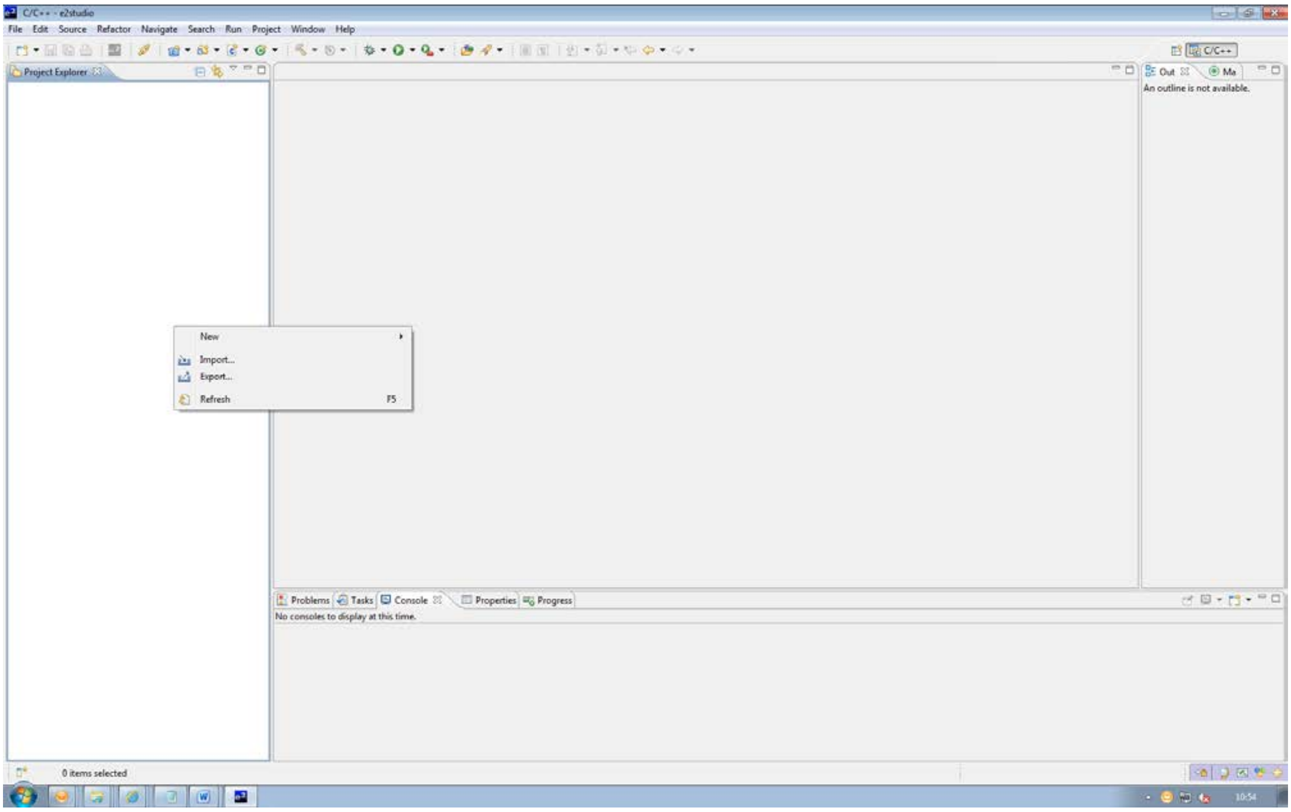
Do you want to continue?

Select <Yes> to Administrator Privilege dialog.



On the welcome screen select Go the Workbench icon as shown above.

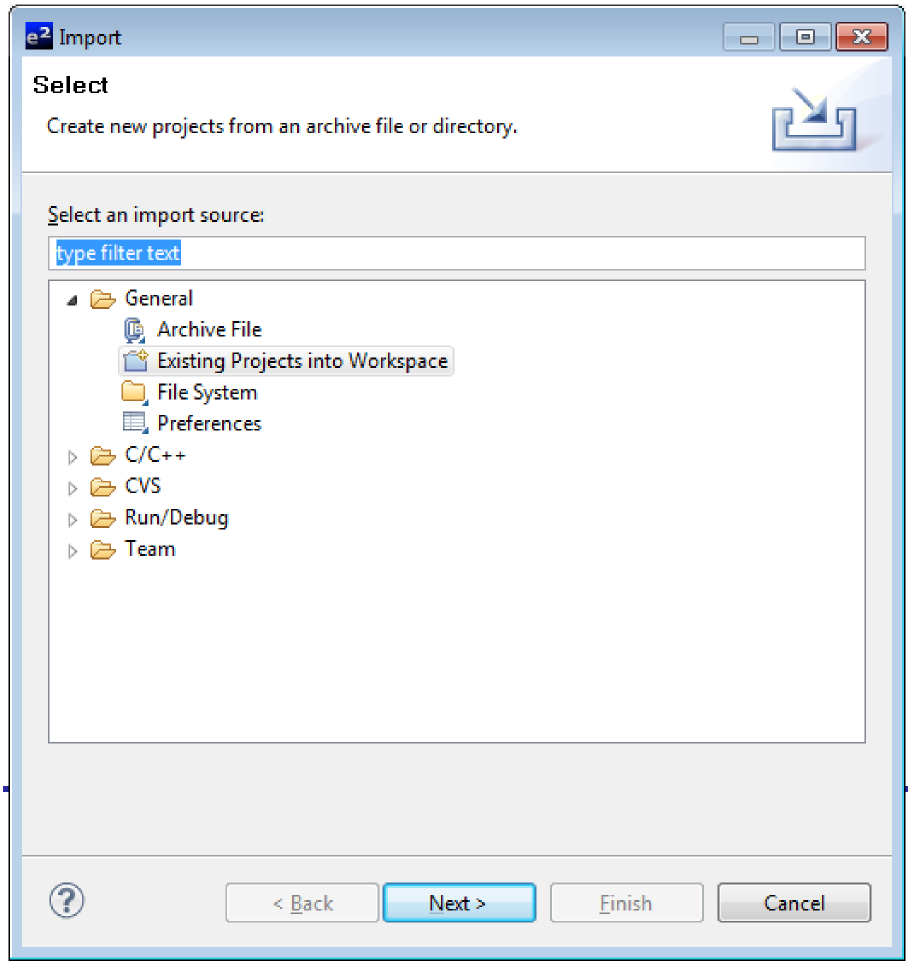
1. Once the e2studio environment has initialised, right click in the project explorer window and click <Import…>



An outline is not available.

No consoles to display at this time.

2. The Import dialog will now appear . Expand the "General" folder icon, and select "Existing Projects into Workspace", then click 'Next'.



Create new projects from an archive file or directory.

Select an import source:

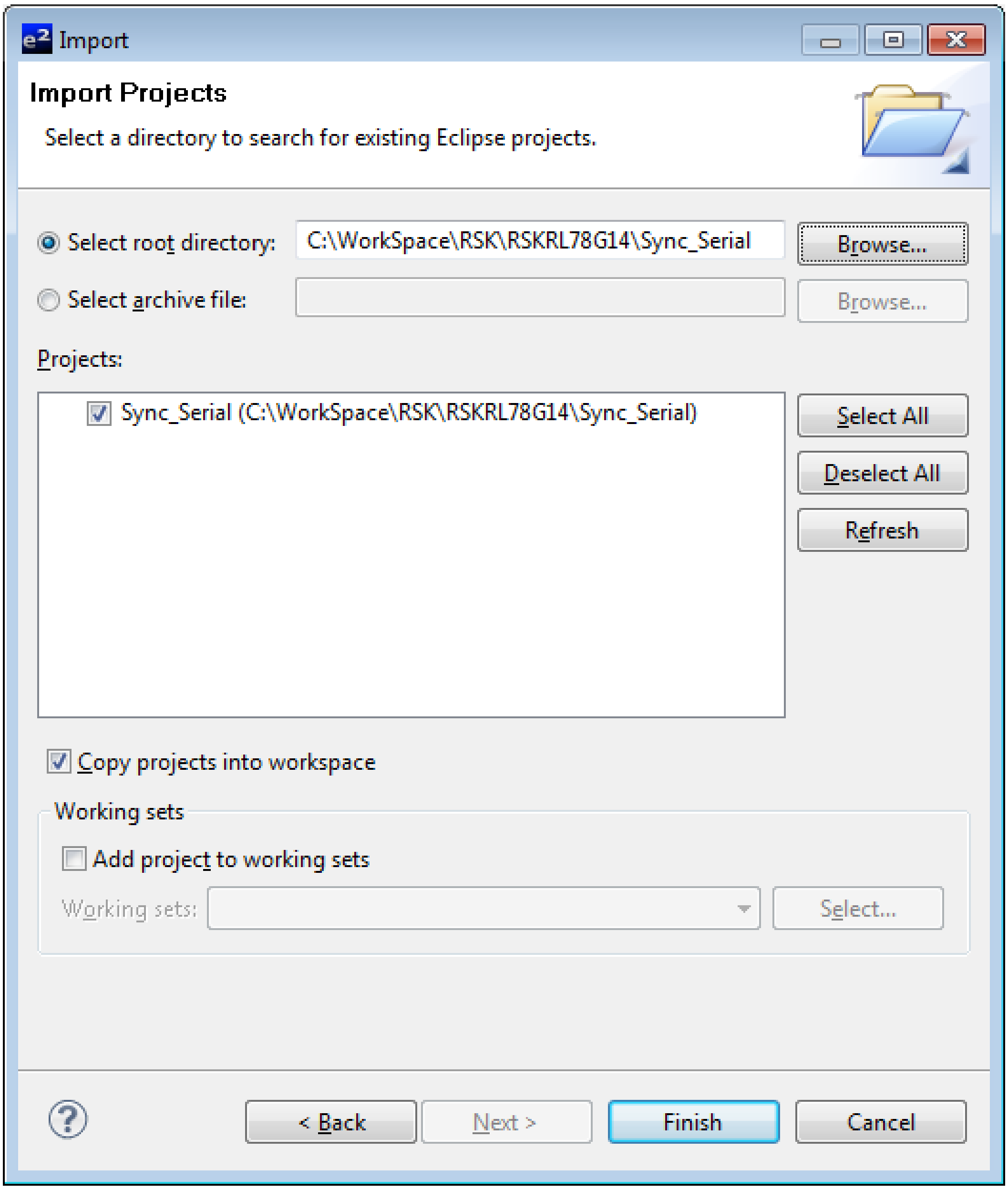
type filter text

3. The Import Dialog will now appear ,specify the project to import .Click the “Browse” button and locate the directory: C:\Workspace\RSK\RSKRL78G14

And also ensure that the ‘Copy projects into workspace’ option is ticked, and then click <Finish>

Navigate to the unzipped Sync_Serial folder located in RSKRL78G14 Workspace folder.

Select the Sync_Serial folder



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: C:\WorkSpace\RSK\RSKRL78G14\Sync_Serial

○ Select archive file:

Projects:

☑ Sync_Serial (C:\WorkSpace\RSK\RSKRL78G14\Sync_Serial)

☑ Copy projects into workspace

Working sets

□ Add project to working sets

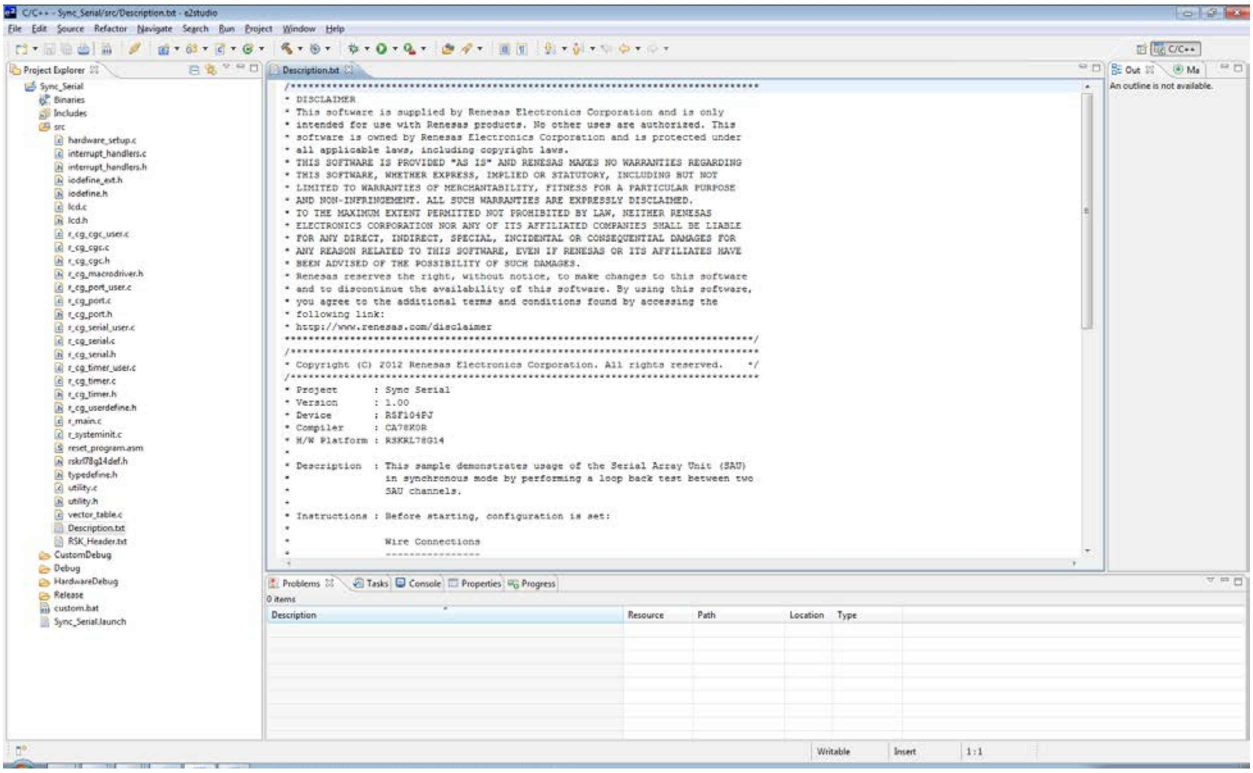
Working sets:

Click <Finish>.

The IDE e2studio will load the project.

3. Opening Sample Code and Source Files

Once the project has been opened, the source code and all dependant files can be opened in the editor by expanding the folders in the Project Tree window and double clicking the files listed. All files have been grouped according to their file type.



```
/*******************************************************************************
* DISCLAIMER
* This software is supplied by Renesas Electronics Corporation and is only
* intended for use with Renesas products. No other uses are authorized. This
* software is owned by Renesas Electronics Corporation and is protected under
* all applicable laws, including copyright laws.
* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
* THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
* LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
* AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
* ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
* ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
* BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
* Renesas reserves the right, without notice, to make changes to this software
* and to discontinue the availability of this software. By using this software,
* you agree to the additional terms and conditions found by accessing the
* following link:
* http://www.renesas.com/disclaimer
*******************************************************************************/
/*******************************************************************************
* Copyright (C) 2012 Renesas Electronics Corporation. All rights reserved.    */
/*******************************************************************************
* Project      : Sync Serial
* Version      : 1.00
* Device       : R5F104PJ
* Compiler     : CA78K0R
* H/W Platform : RSKRL78G14
*
* Description  : This sample demonstrates usage of the Serial Array Unit (SAU)
*                in synchronous mode by performing a loop back test between two
*                SAU channels.
*
* Instructions : Before starting, configuration is set:
*
*                Wire Connections
*                ----------------
```

4. Source Code Functionality

The source code project is specifically written to run on the appropriate RSK. However this source code can be useful as an example even without the RSK.

The project was written using source files containing API functions generated using Code Generator. The project will contain a C source file ‘r_main.c’. This source file includes the C function main(). All source files and dependant files whose filenames are prefixed with ‘r_’ were generated using Code Generator.

5. Code Execution

Before starting, configuration is set:

Wire Connections

JA2, pin 13 (SCL20) to J2, pin 15 (SCL21)

JA2, pin 14 (SI20) to J2, pin 13 (SO21)

JA2, pin 15 (SO20) to J2, pin 14 (SI21)

1. Build and download the sample code to the RSK.

2. Right click the variable gCSI20_RXBuffer and select 'Register to Watch 1'. Do the same for the variable

gCSI20_TXBuffer, gCSI21_RXBuffer and gCSI21_TXBuffer.

3. Click the 'Resume' button to start the program execution. The name of the sample will be displayed on the debug LCD.

4. The sample will then begin the data transfer. When the SAU activity is complete, the second line of the debug LCD will display either "Success" or "Failure".

5. You can observe the contents of gCSI20_RXBuffer being filled

with the contents of gCSI21_TXBuffer. The contents should match. The same should be done regarding gCSI20_TXBuffer and gCSI21_RXBuffer.

6. If the debug LCD reports "Failure", or shows "CSI0 Err" or "CSI1 Err" without changing; check the configuration specified is set and click on the "Restart" button to start again.

6. Website, Inquiries and Support

Renesas Electronics Website
    http://www.renesas.com/
Inquiries
    http://www.renesas.com/inquiry
Support
    http://www.renesas.com/rskrl78g14
All trademarks and registered trademarks are the property of their respective owners.

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Oct 12, 2012	—	First edition issued

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.
   — 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.
   — 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.
   — 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 one with a different type number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.