To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of 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. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may 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.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

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

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be 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, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

User's Manual

Renesas Starter Kit for SH7264

Software Manual

RENESAS SINGLE-CHIP MICROCOMPUTER
SuperH FAMILY

Renesas Electronics
www.renesas.com

Rev.1.0 2009.12

Disclaimer

This software is supplied by Renesas Technology Corp. and is only intended for use with Renesas products. No other uses are authorized.

This software is owned by Renesas Technology Corp. 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 TECHNOLOGY CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR ANY REASON RELATED TO THE 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

Table of Contents

Table of Figures

Chapter 1. Preface

Cautions

This document may be, wholly or partially, subject to change without notice.

Trademarks

Copyright

Website: www.renesas.com

Glossary

ADC	Analogue / Digital Converter	I/O	Input / Output
API	Application Programming Interface	PDL	Peripheral Driver Library
CMT	Compare Match Timer	RSK	Renesas Starter Kit
CPG	Clock Pulse Generator	USB	Universal Serial Bus
DMAC	Direct Memory Access Controller	WDT	Watchdog Timer
HEW	High performance Embedded Workshop		

Chapter 2. Introduction

This document explains the sample code for the Renesas Starter Kit.

It explains by text and diagrams the functionality of the sample code and its interaction with the Renesas Peripheral Driver Library (RPDL).

The Renesas Peripheral Driver Library (hereinafter “RPDL library”) is based upon a unified API for the microcontrollers made by Renesas Technology.

The latest version of this API can be found in the Documentation section of the PDG site (http://www.renesas.com/pdg).

Chapter 3. RSK Sample Code Concept

The diagram below shows the basic structure of all the RSK sample code:

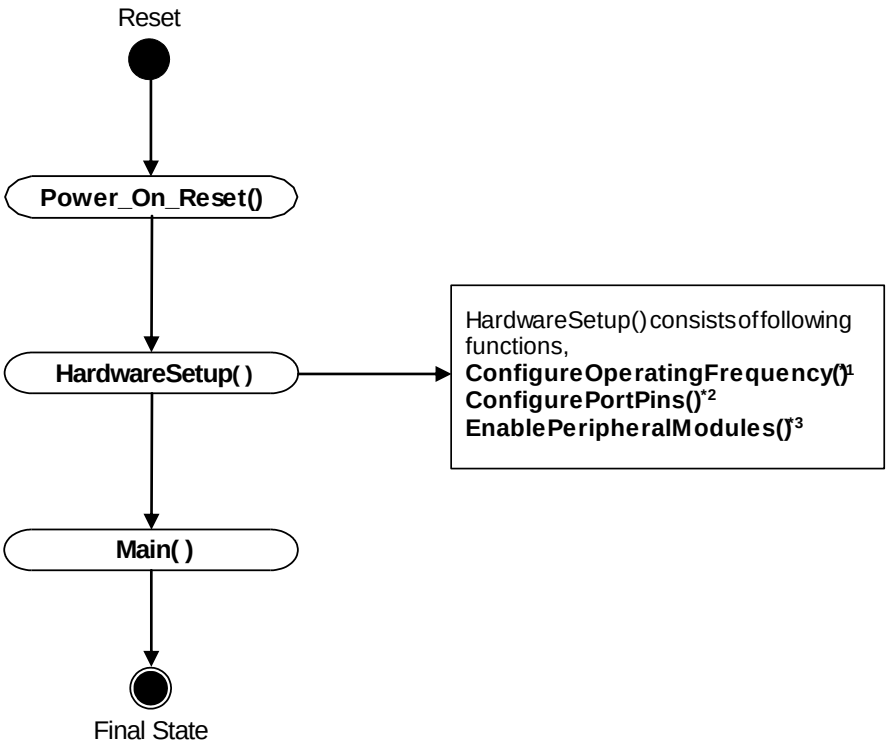

Figure 1: RSK Sample Code

Note:

1. This function initialises the CPU main clock and the PLL.
2. This function configures the CPU port pins as input and output. It also determines the initial state of the output port pin.
3. This function enables or disables the peripheral modules of the CPU.

3.1. List of Sample code

You can click on the sample code title for more description.

	Sample Code	Description
1	Tutorial	Demonstrates the use of debugger and hardware
2	ADC_OneShot	Demonstration of the on-chip ADC module for one shot conversion.
3	ADC_Repeat	Demonstration of the on-chip ADC module for repeat conversion.
4	Sync_Serial	Demonstration of the on-chip SCI module for synchronous serial communication.
5	Timer_Capture	Demonstrates on-chip MTU2 module in input capture mode.
6	Timer_Event	Demonstrates on-chip MTU2 module in event counter mode
7	Timer_Mode	Demonstrates on-chip MTU2 module to generate square waveform.
8	Timer_Compare	Demonstrates on-chip CMT module in output compare mode.
9	RTC	Demonstrates the configuration of on-chip RTC peripheral.
10	BSC	Demonstrates the configuration of external SDRAM using Bus state control.
11	CAN	Demonstrates the communication between 2 CAN nodes.
12	Async Serial	Demonstrates the UART communication between RSK2+SH7264 and PC HyperTerminal.
13	SPI	Demonstrates the configuration & usage of SH7264 on-chip Renesas Serial Peripheral Interface.
14	WDT	Demonstrates the SH7264 on-chip watchdog timer.
15	DMAC	Demonstrates the configuration of DMAC with source and destination in normal transfer mode.
16	Ethernet Driver	Demonstrates the configuration of the on-chip Ethernet Controller.
17	IIC Master	Demonstrates the SH7264 on-chip IIC3 module.
18	USB_LibUSB	Demonstrates the LibUSB support for SH7264 on-chip USB module.
19	USB_MSC	Demonstrates the Mass Storage Device Class support of SH7264 on-chip USB module.
20	USB_HID	Demonstrates the Human Interface Device (HID) Class USB support of SH7264 on-chip USB module.
21	USB_CDC	Demonstrates the Communication Device Class (CDC) support of SH7264 on-chip USB module.

Table 3-1: List of sample code

3.1.1. Tutorial

The tutorial demonstrates the use of the debugger and the hardware. The tutorial code is common for all RSK's.

This code will call three main functions to demonstrate port pin control, interrupt usage and C variable initialization. These functions are shown below,

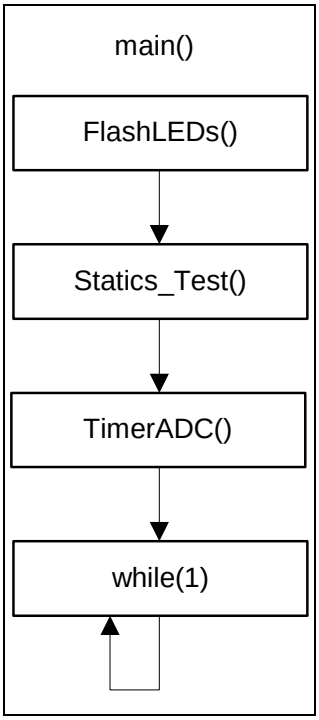

Figure 2: Functions used in Tutorial code

3.1.1.1. Sequence Diagram

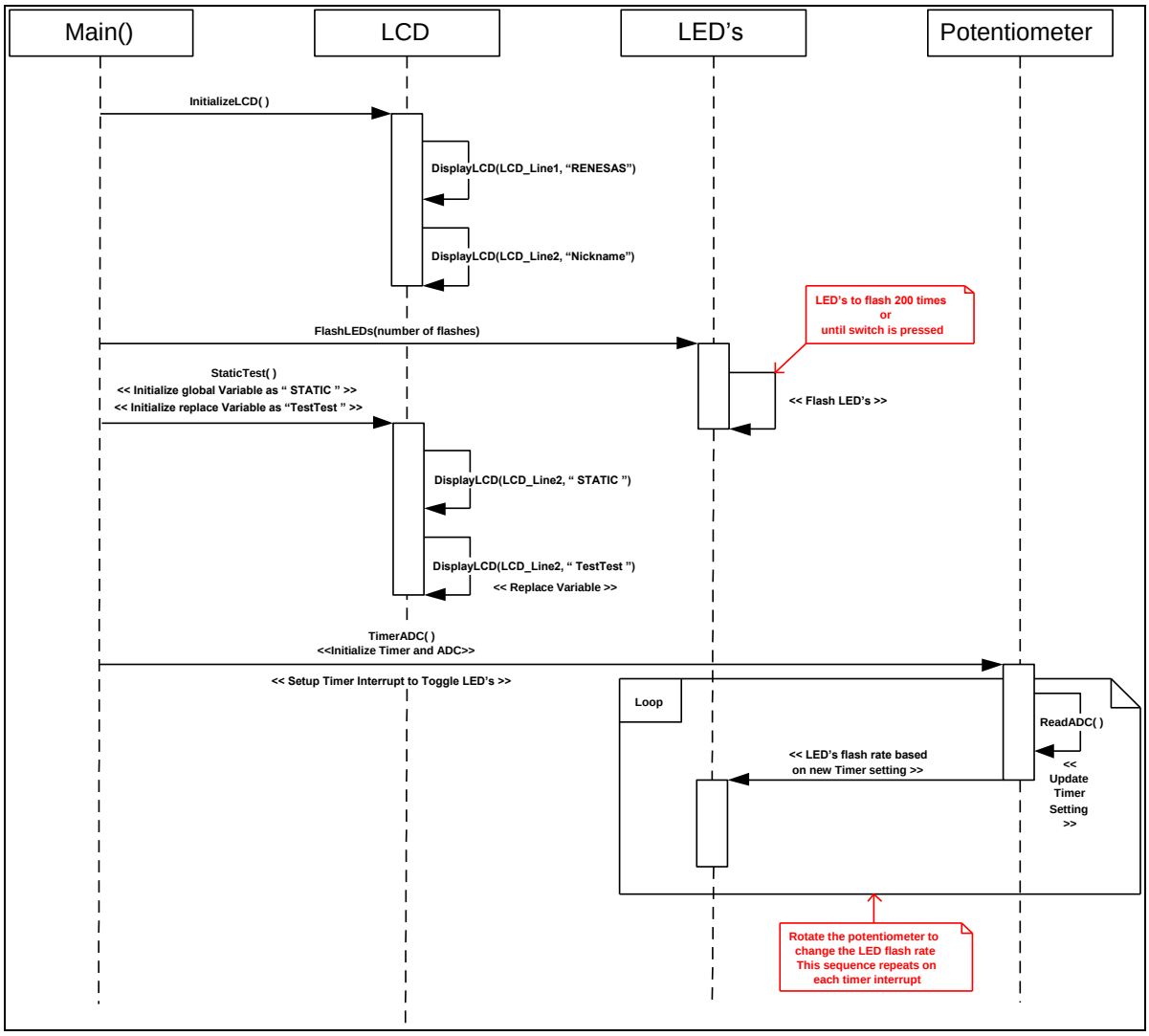

Figure-3: Tutorial Sequence Diagram

3.1.1.2. Description

1- The tutorial code initializes the LCD module and displays “Renesas” on Line-1 and the RSK nickname on Line-2.

2- All the LED’s start flashing. The LED’s flash 200 times or until switch is pressed.

3- After LED flashing, the LCD will display a variable “STATIC” which is then replaced by “TESTTEST” to demonstrate the static variable test.

4- In last step, the flash rate is now controlled by the potentiometer (RV2).

	Function	RPDL API Function
1	FlashLEDs( )	ModifyIOPort
2	TimerADC( )	R_ADC_10_Create R_ADC_10_Control Start_ADC_10 R_MTU_Create R_INTC_Modify check_Wr_Pin_H R_MTU_Control R_MTU_Read

Table 3-2: RPDL API function list for Tutorial code

3.1.2. ADC_OneShot

This sample code demonstrates the use of on-chip ADC module for one shot conversion.

3.1.2.1. Sequence Diagram

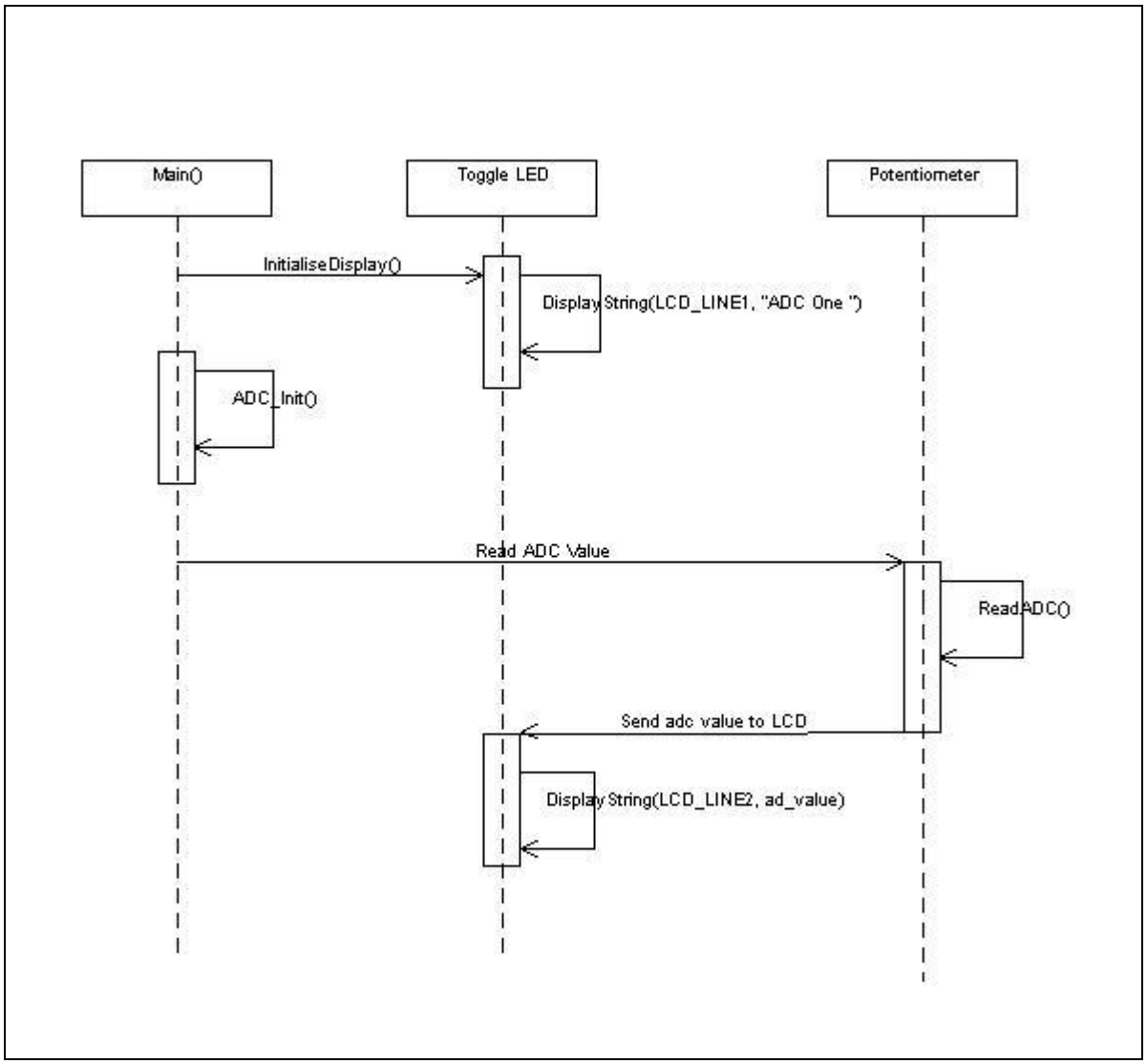

Figure 4: ADC_OneShot Sequence Diagram

3.1.2.2. Description

1- ADC_OneShot sample code initializes the LCD module and display “ADC One” on Line-1

2- The ADC channel is configured for one shot conversion.

3- The result is stored in the “usADC_Result” variable and user may examine the AD conversion result in “usADC_Result” using HEW C watch window.

4- The debug LCD also displays the result of AD conversion in hex format.

5- The AD conversion result is shown on LCD

	Function	RPDL API Function
1	ADC_Init( )	R_ADC_10_Create
2	ReadADC()	R_ADC_10_Control R_ADC_10_Read

Table 3-3: RPDL API function list for ADC_OneShot

3.1.3. ADC_Repeat

This sample code demonstrates the use of on-chip ADC module for repeat conversion.

3.1.3.1. Sequence Diagram

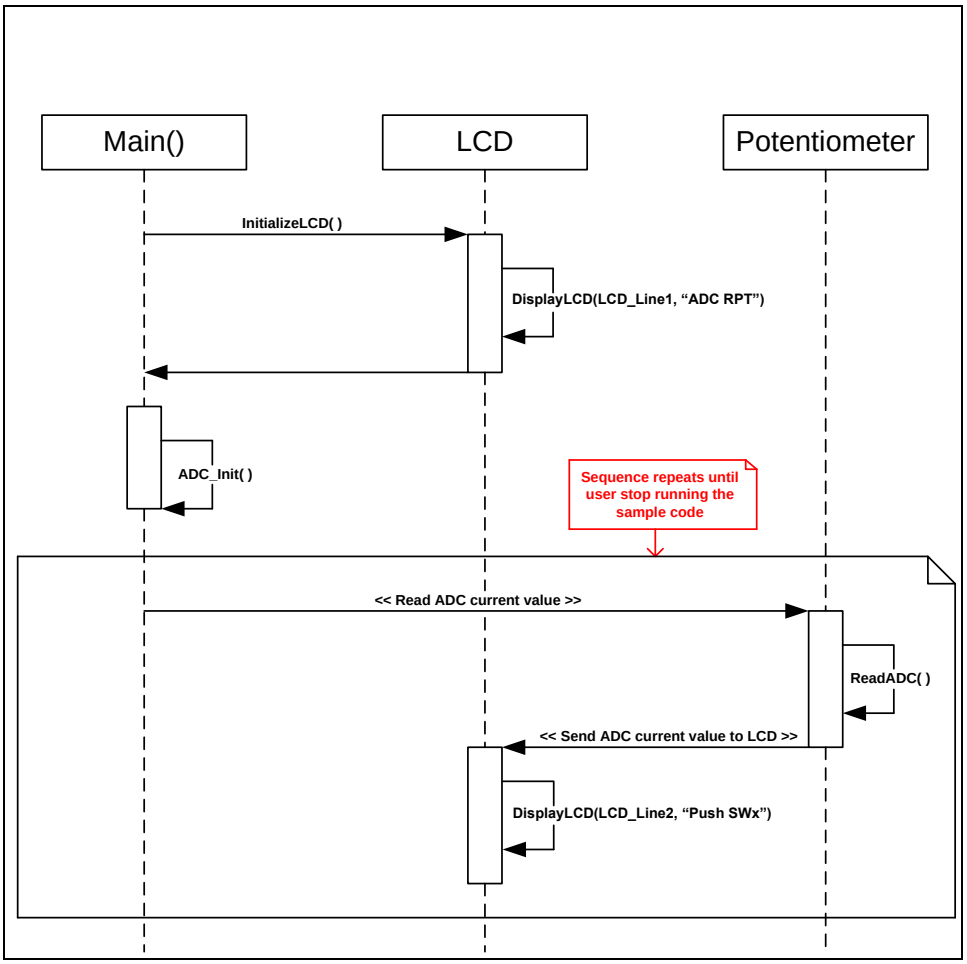

Figure 5: ADC_Repeat Sequence Diagram

3.1.3.2. Description

1- The ADC_Repeat sample code initializes the LCD module and displays “ADC REP” on Line-1 and AD conversion result on Line-2.

2- The ADC channel is configured for AD repeat conversion.

3- The result is stored in the “usADC_Result” variable and user may examine the AD conversion result in “usADC_Result” using HEW C watch window.

4- The debug LCD also displays the result of last AD conversion in hex format.

	Function	RPDL API Function
1	ADC_Init( )	R_ADC_10_Create R_ADC_10_Control
2	ReadADC( )	R_ADC_10_Read

Table 3-4: RPDL API function list for ADC_Repeat

3.1.4. Sync_Serial

This sample code demonstrates synchronous serial Communication between two SCIF channels.

3.1.4.1. Sequence Diagram

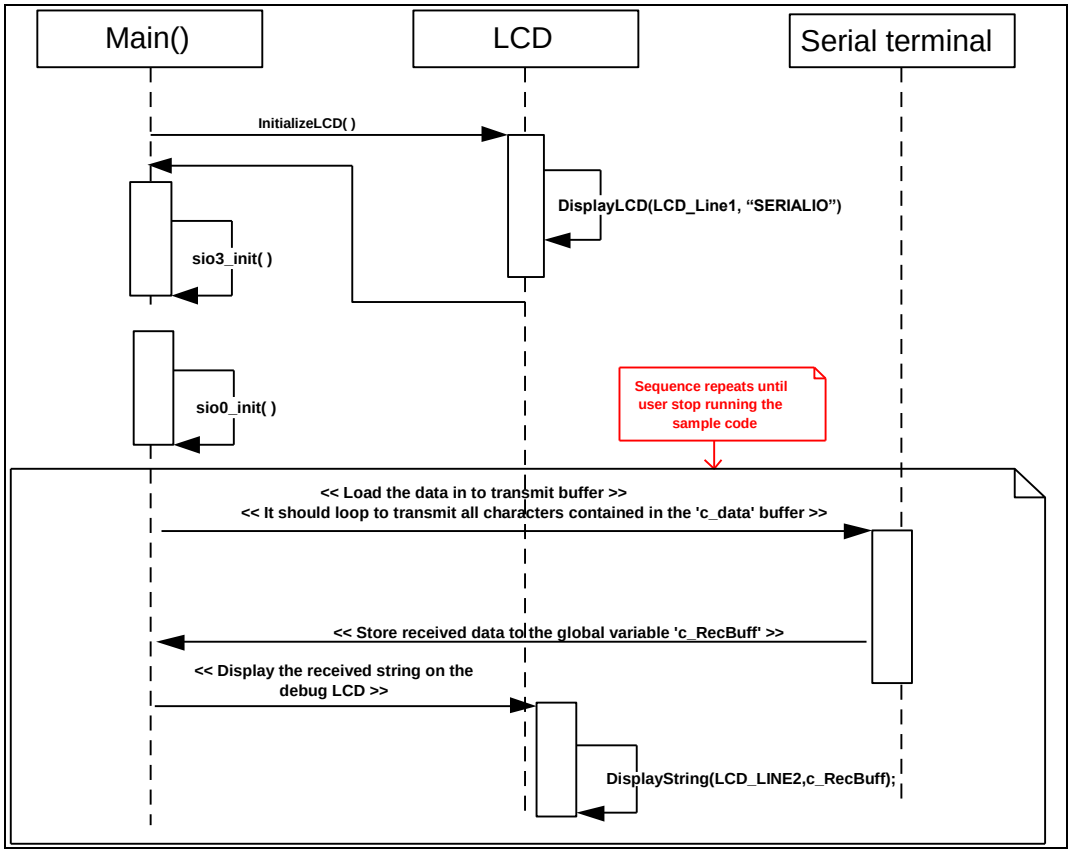

Figure 6: Sync_Serial Sequence Diagram

3.1.4.2. Description

1- The Sync_serial sample code initializes the LCD module and displays “SERIALIO” on Line-1.

2- The serial channel is configured for synchronous data communication. SCI3 is configured as transmitter and SCI0 is configured as receiver.

3- This transmitter sends the serial data to the receiver port, synchronised with the clock.

4- The received data should match with the transmitted data string.

5- Successful reception is displayed on the LCD.

	Function	RPDL API Function
1	sio3_init ()	R_SCI_Create
2	sio0_init ()	R_SCI_Create
3	main ()	R_SCI_Send
		R_PFC_Modify
		R_IO_PORT_Set

Table 3-5: RPDL API function list for Sync_Serial

3.1.5.Timer_Capture

This sample code configures the timer to capture timer counter's value in one of the general purpose registers, on reception of input signal at one of timer's pins.

3.1.5.1. Sequence Diagram

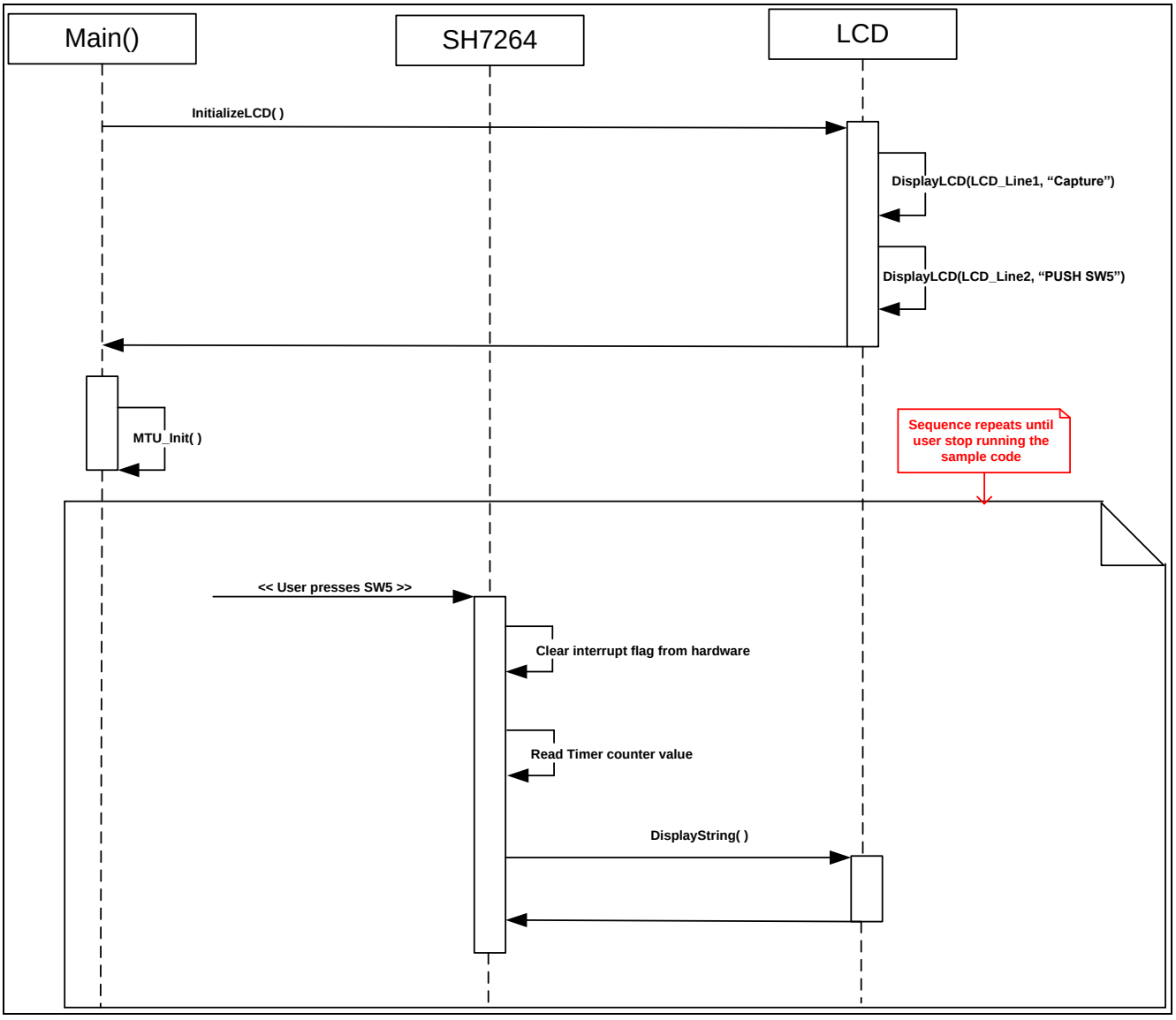

Figure 7: Timer_Capture Sequence Diagram

3.1.5.2. Description

1- The Timer_Capture initializes the timer to capture value of timer's counter in a general purpose register at the instant the user presses switch.

2- On every switch press, it displays value of timer counter register at that instant on LCD.

	Function	RPDL API Function
1	MTU_Init( )	R_MTU_Create R_IO_PORT_Set R_INTC_Modify R_INTC_Modify
2	timer_capture_handler()	R_MTU_Read

Table 3-6: RPDL API function list for Timer Capture

3.1.6. Timer_Event

This sample code configures the timer to count the number of input signal events.

3.1.6.1. Sequence Diagram

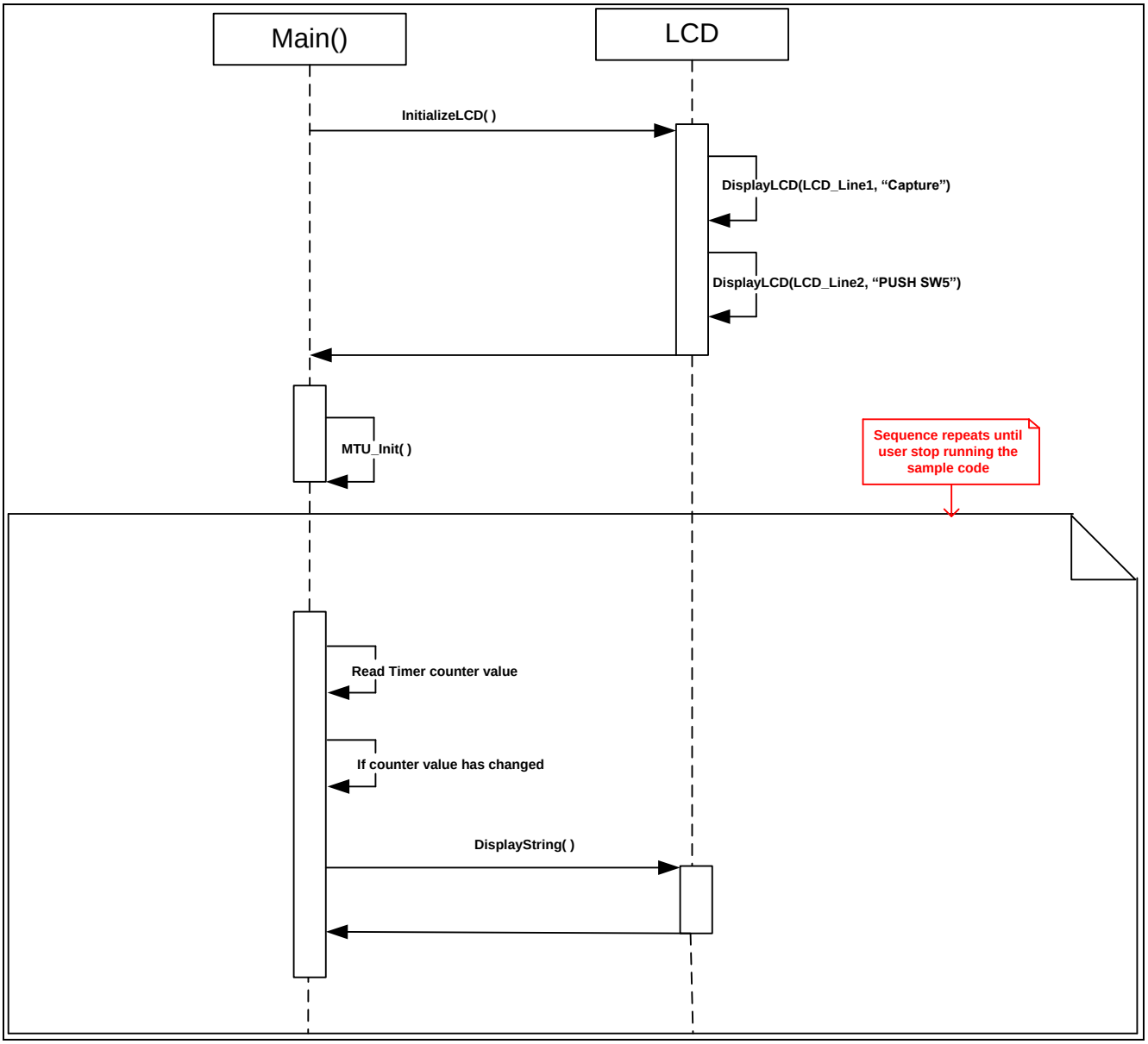

Figure 8: Timer_Event Sequence Diagram

3.1.6.2. Description

1- The Timer_Event initializes the timer to count at user switch press events.

2- On every switch press, the counter increments value by one unit & it is displayed on LCD.

	Function	RPDL API Function
1	MTU_Init( )	R_IO_PORT_Set R_MTU_Create
2	Main()	R_MTU_Read

Table 3-7: RPDL API function list for Timer Event

3.1.7. Timer_Mode

This sample code configures the timer to generate 1KHz waveform. The waveform can be seen on the oscilloscope.

3.1.7.1. Sequence Diagram

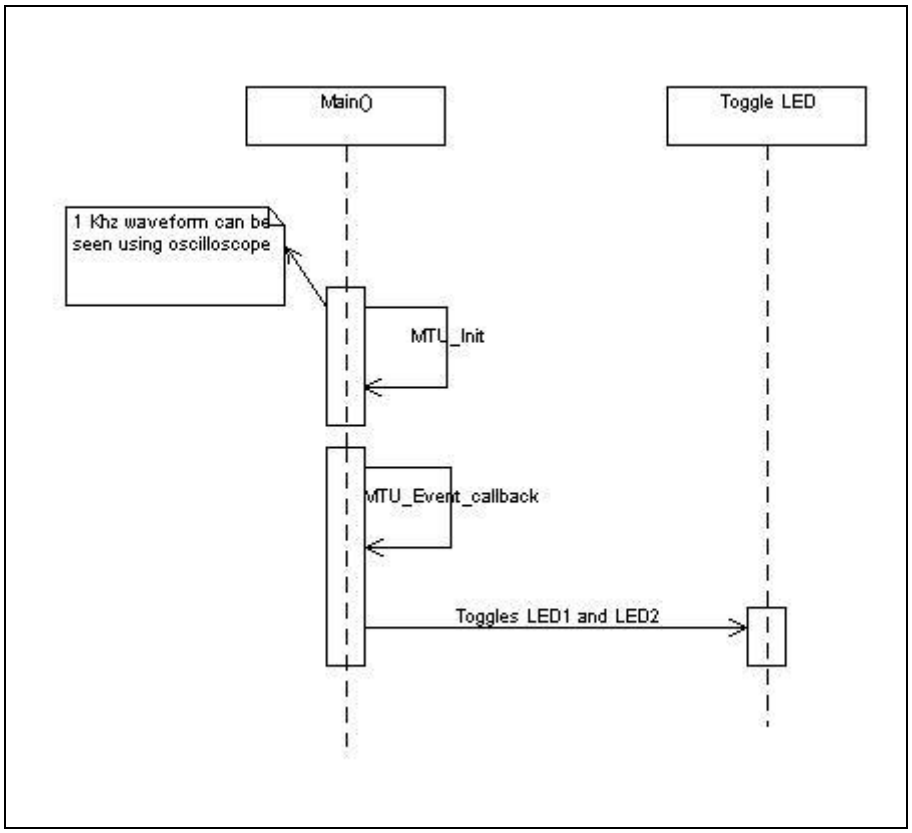

Figure 9: Timer_Mode Sequence Diagram

3.1.7.2. Description

1- The Timer_Mode initializes the timer to generate 1KHz waveform on the respective timer port pin..

2- On every timeout, it toggle LED1

3- The waveform can be seen using oscilloscope.

	Function	RPDL API Function
1	Timer_Init( )	R_MTU_Create R_MTU_Read R_MTU_Control ModifyIOPort R_INTC_Modify

Table 3-8: RPDL API function list for Timer Mode

3.1.8. Timer_Compare

This sample code configures the compare match timer to generate an interrupt when timer counter value matches with the one in compare match constant register.

3.1.8.1. Sequence Diagram

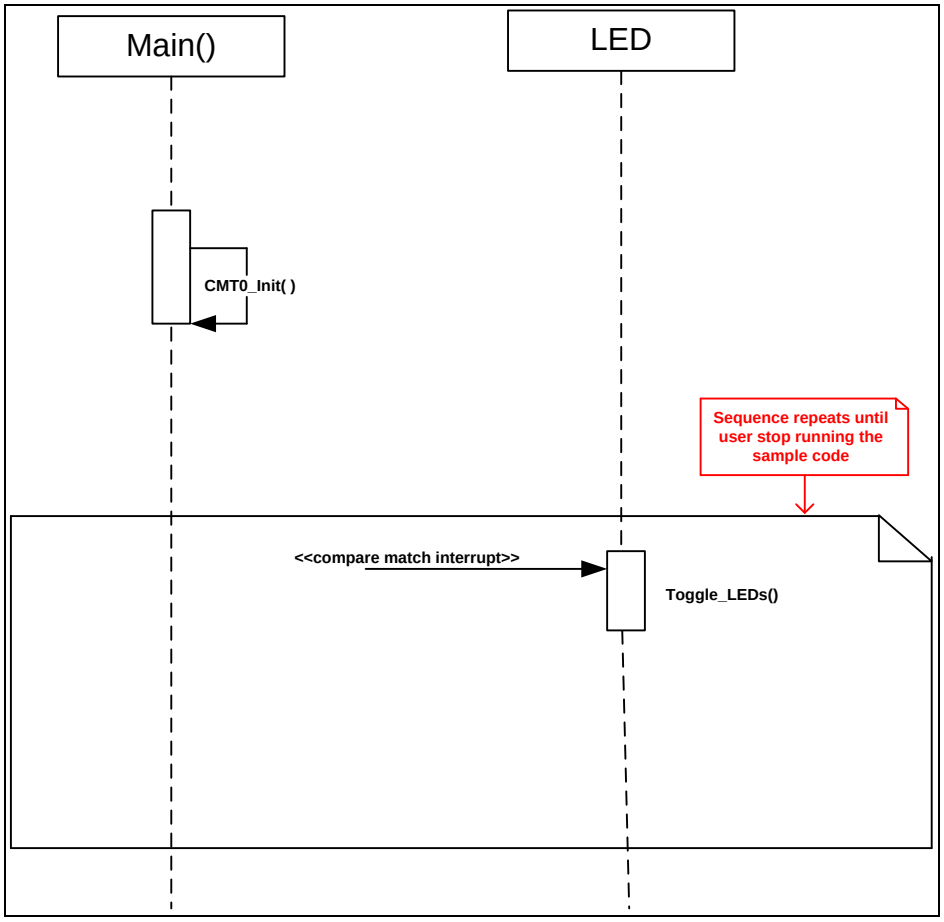

Figure 10: Timer_Compare Sequence Diagram

3.1.8.2. Description

1- The Timer_Compare initializes the compare match timer to generate an interrupt when timer counter value matches the value in compare match constant register.

2- The interrupt service routine toggles LED's whenever it is executed.

	Function	RPDL API Function
1	CMT0_Init ()	R_CMT_Create

Table 3-9: RPDL API function list for Timer Compare

3.1.9. RTC

This sample code configures the RTC to keep track of time of day in hours, minutes and seconds.

3.1.9.1. Sequence Diagram

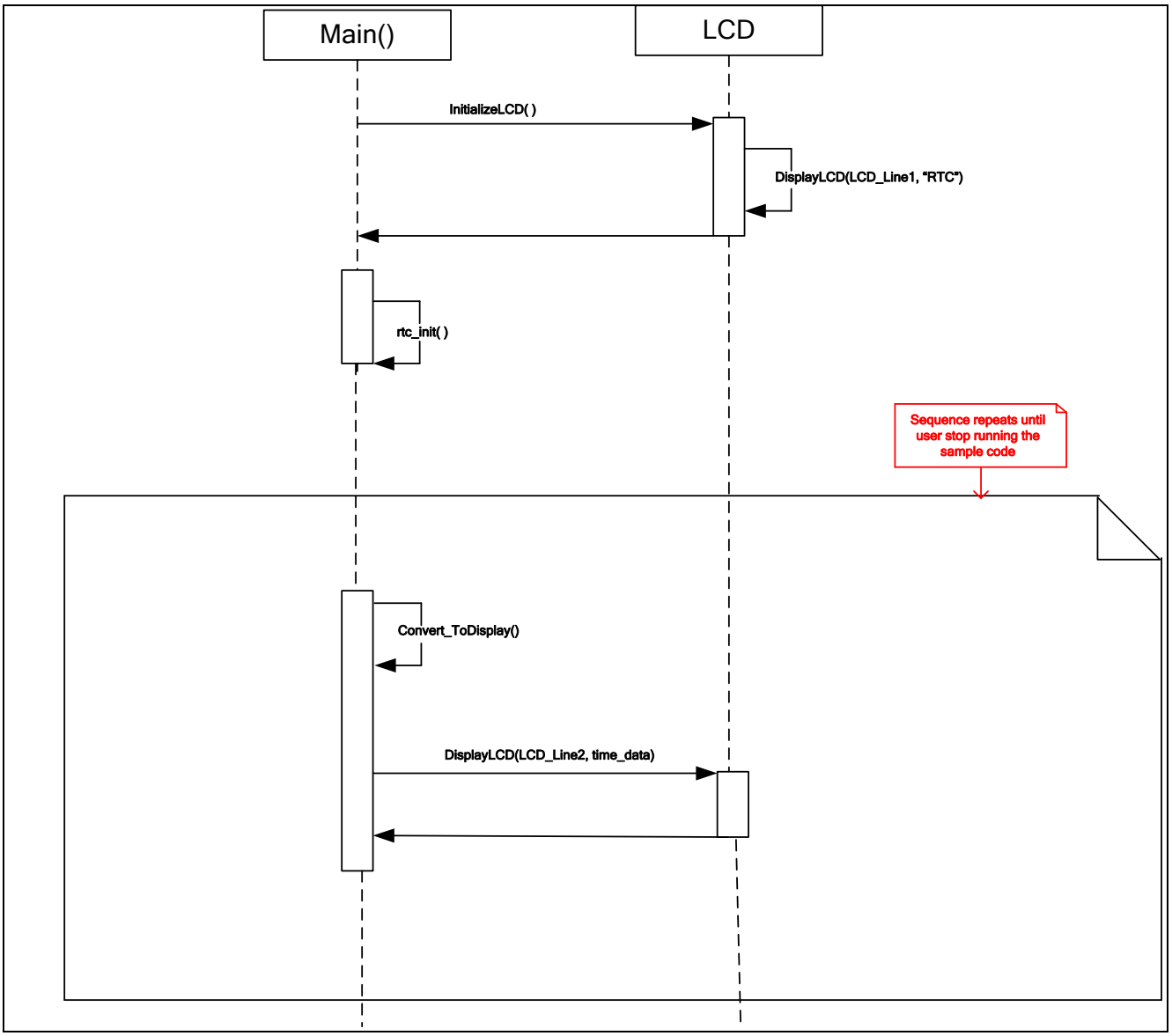

Figure 11: RTC Sequence Diagram

3.1.9.2. Description

1- The Timer_Compare initializes the compare match timer to generate an interrupt when timer counter value matches the value in compare match constant register.

2- The interrupt service routine toggles LED's whenever it is executed.

	Function	RPDL API Function
-	-	-

Table 3-10: RPDL API function list for RTC

3.1.10. BSC

This sample code uses Bus State Controller to display image on a TFT LCD screen.

3.1.10.1. Sequence Diagram

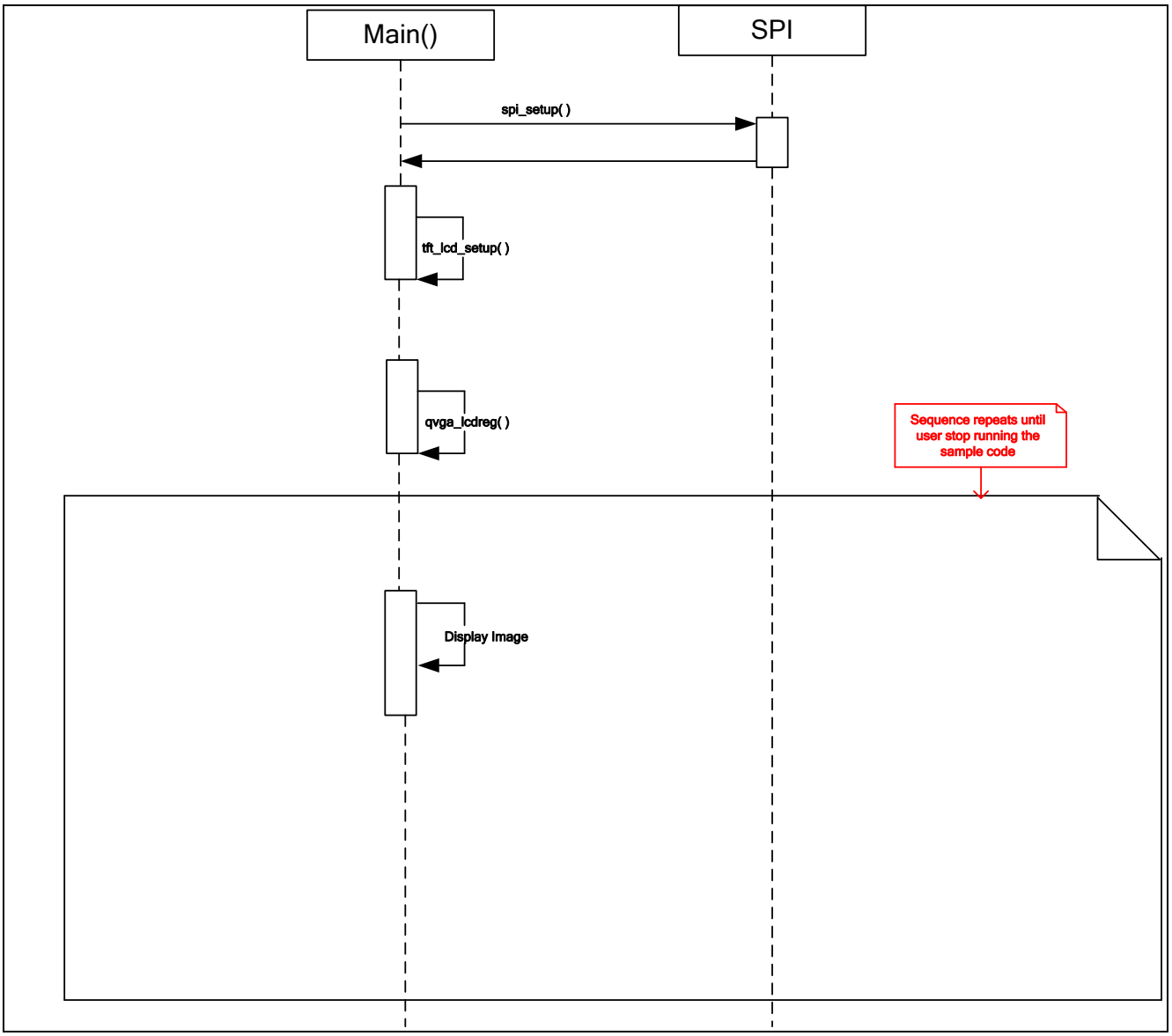

Figure 12: BSC Sequence Diagram

3.1.10.2. Description

1- The BSC sample application initializes SPI, LCD controller & Bus State Controller to display image on a TFT LCD screen.

2- By default Red, Green and Blue colour bars are displayed on the TFT LCD.

3- If complete version of HEW is available then this application can be configured to display an image instead of colour bars by

   a. Uncomment the "USE_IMAGE_FILE" macro.

   b. Including the image file in the project.

   c. Adding a memory section for image data in ROM area.

	Function	RPDL API Function
1.	tft_lcd_setup	R_PFC_Modify R_PFC_Write
2.	spi_setup	R_IO_PORT_Set R_PFC_Modify

Table 3-11: RPDL API function list for BSC

3.1.11. CAN

This sample code supports communication using CAN ((RCAN) 11-bits Standard ID) protocol between two boards using polling mode.

3.1.11.1. Sequence Diagram

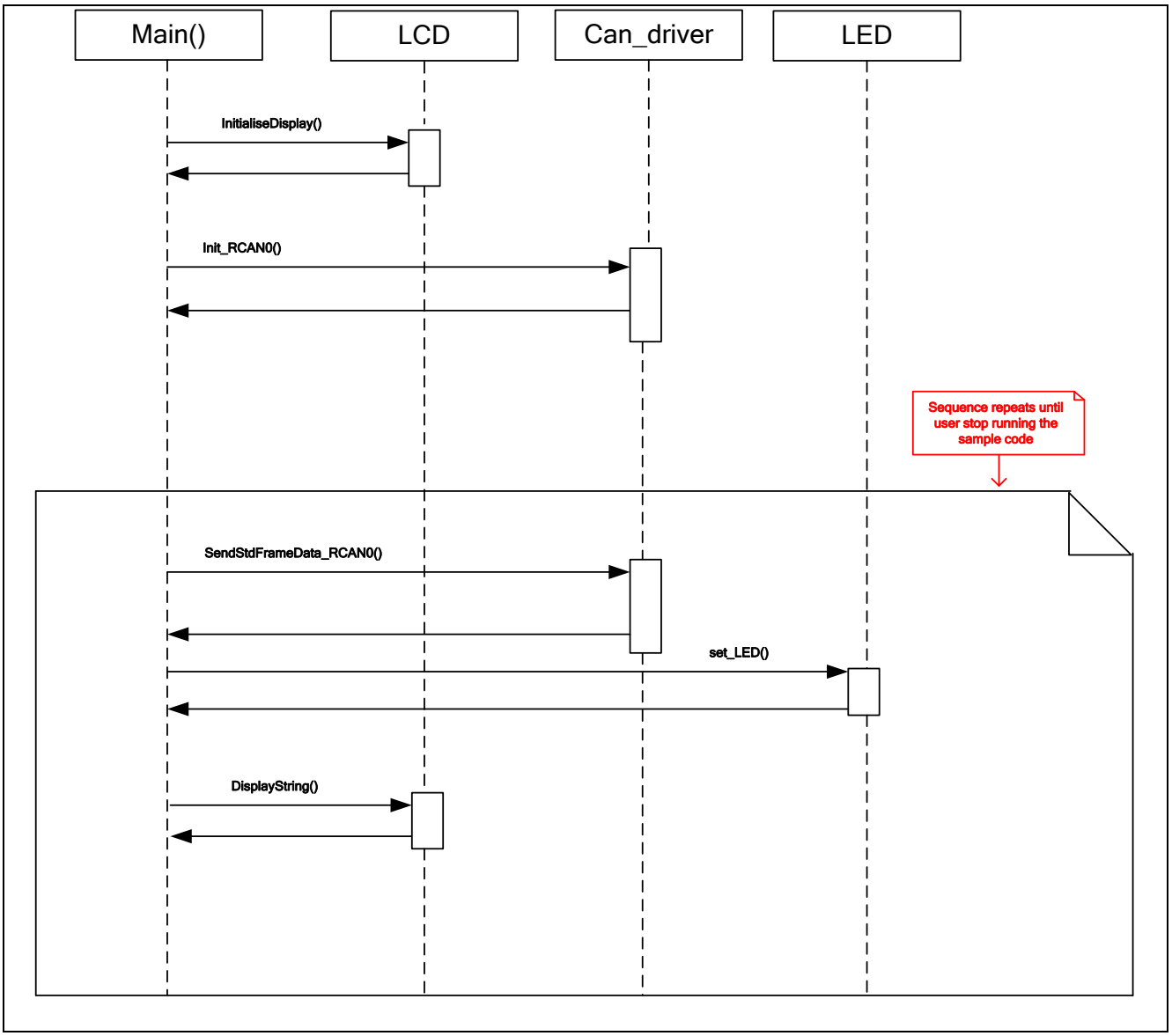

Figure 13: CAN Sequence Diagram

3.1.11.2. Description

1- The sample code communicates between two boards using CAN protocol.

2- If the sample is built with macro “BOARD_TRANSMIT” defined, then that board will act as CAN transmitter.

3- If the sample is built without “BOARD_TRANSMIT” macro defined, then that board will act as CAN receiver.

4- Connect two boards via 3-wire CAN connecting cable. Make one RSK as Transmitter and other one as Receiver.

5- If the data received is the same as the expected data, then CAN receiver turns LED1 ON. In case of error, it switches on LED2 along with suitable message on LCD screen.

6- Similarly if data transmit operation succeeds at CAN transmitter then it turns on LED1 and in case of error, LED2.

	Function	RPDL API Function
1.	set_LED	R_IO_PORT_Wr_Pin_H
2.	Init_RCAN0	R_IO_PORT_Set R_PFC_Modify
3.	Init_RCAN1	R_IO_PORT_Set R_PFC_Modify

Table 3-12: RPDL API function list for CAN

3.1.12. Async Serial

This sample code establishes UART communication between RSK2+SH7264 and PC HyperTerminal. The RSK sends incrementing numbers to PC HyperTerminal.

3.1.12.1. Sequence Diagram

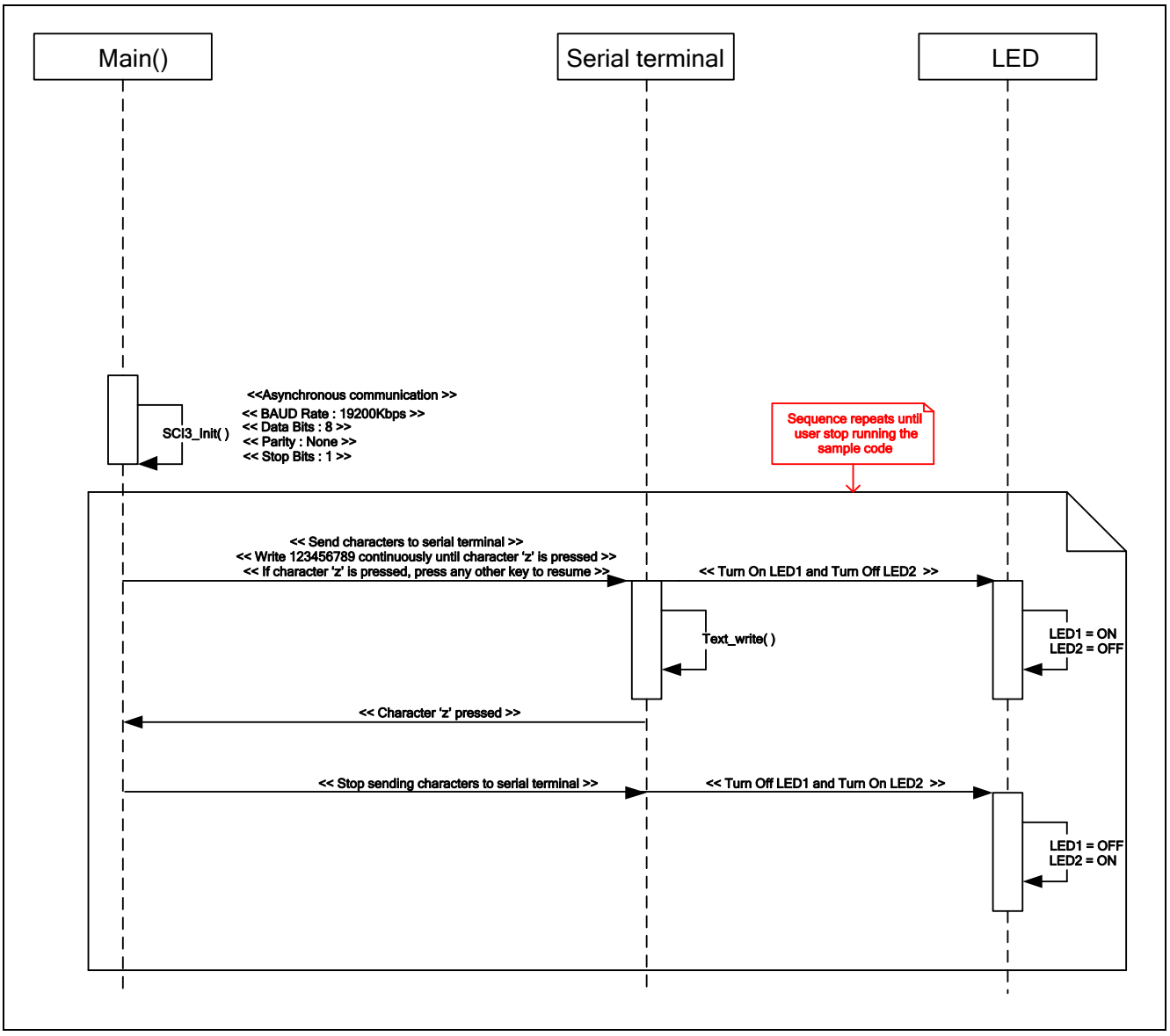

Figure 14: Async_Serial Sequence Diagram

3.1.12.2. Description

1- The Async_Serial sample code initializes the “Uart” in asynchronous communication mode.

2- The serial channel is configured as: 19200bps baud rate, 8 data bits, 1 stop bit and no parity.

3- This code communicates with the terminal software on the PC (e.g. Hyper Terminal on Windows).

4- Incrementing data (0 to 9) is sent to the terminal window repeatedly.

5- Sending data can be stopped by pressing ‘z’ on the PC keyboard.

6- To resume, press any key other than ‘z’.

	Function	RPDL API Function
1	SCI3_Init()	R_SCI_Create R_SCI_Receive
2	text_write()	R_SCI_Send

Table 3-13: RPDL API function list for Async_Serial

3.1.13. SPI

This sample demonstrates configuration & usage of SH7264 on-chip Renesas Serial Peripheral Interface.

3.1.13.1.Sequence Diagram

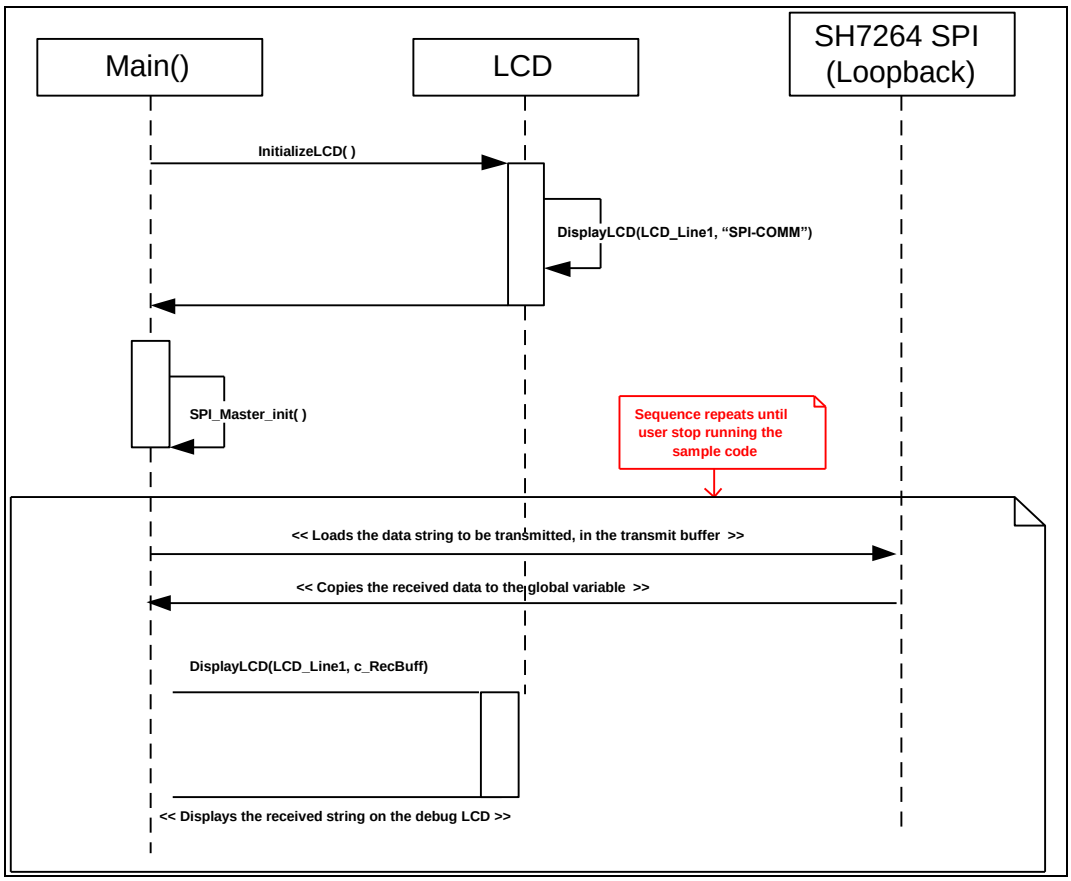

Figure 15: SPI Sequence Diagram

3.1.13.2. Description

1- The SPI sample code initializes the LCD module and displays “SPI-COMM” on Line-1.

2- The SPI channel is configured to communicate from MCU to itself i.e. in loop back mode.

3- The data contained in buffer is transmitted from the Master pin.

4- Successful reception of data over a serial interface is displayed on the LCD on Line-2.

	Function	RPDL API Function
1	SPI_Master_init ()	R_PFC_Modify R_IO_PORT_Set

Table 3-14 RPDL API function list for SPI

3.1.14. WDT

This sample application demonstrates SH7264 on-chip watchdog timer.

3.1.14.1. Sequence Diagram

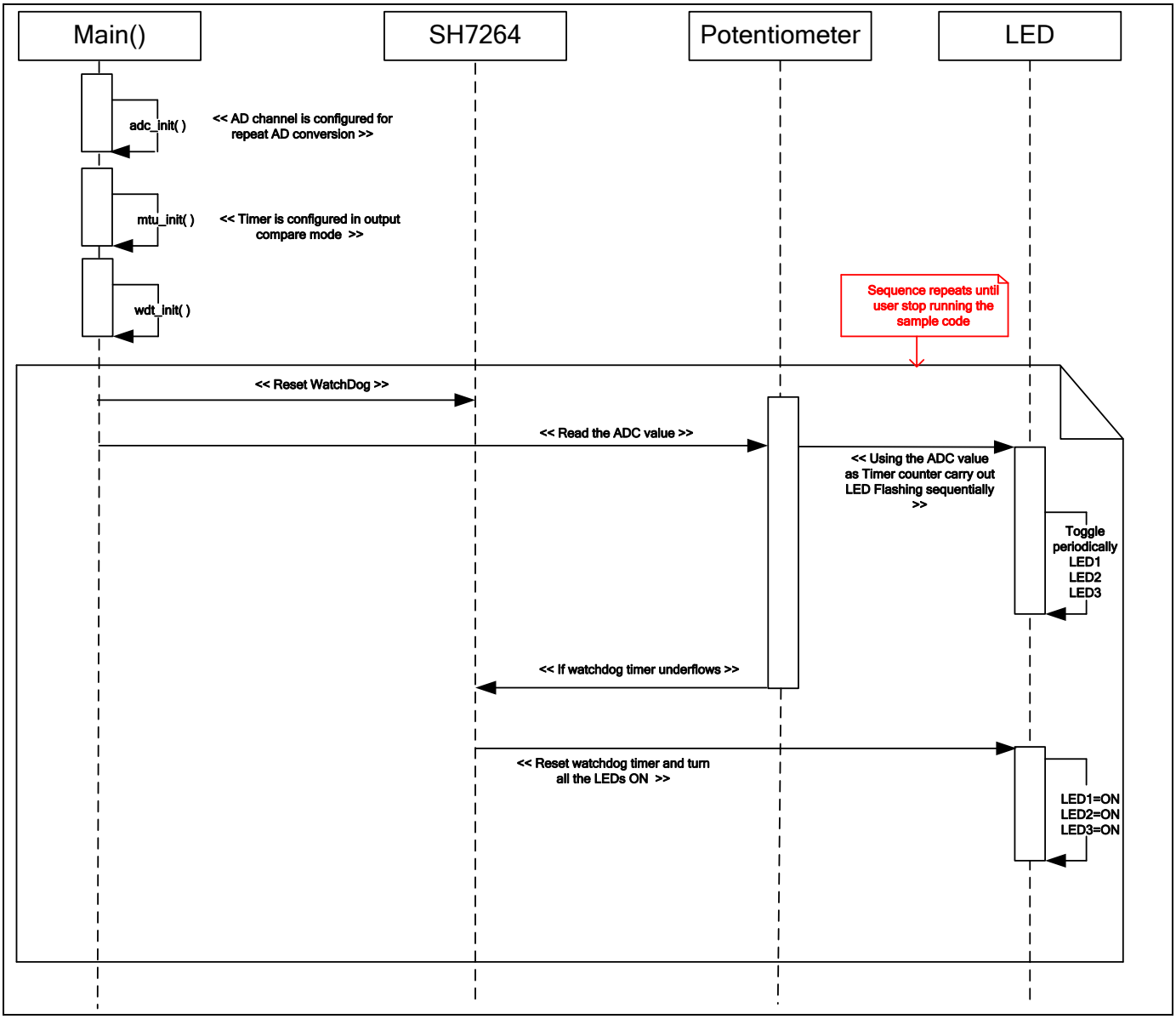

Figure 16: WDT Sequence Diagram

3.1.14.2. Description

1- The WDT sample code initializes the Timer in output compare mode, and AD channel in scan mode.

2- The LEDs flash sequentially with the timing change controlled using the POT (AD Channel).

3- The watchdog timer is configured such that, if the MTU2 timer interrupt if more than the watchdog timer period, will generate an interrupt and reset the watchdog.

4- If watchdog is activated, all LEDs turn ON.

	Function	RPDL API Function
1	mtu_init ()	R_MTU_Create R_INTC_Modify R_ADC_10_Read R_MTU_Control
2	adc_init ()	R_ADC_10_Create R_ADC_10_Control Start_ADC_10

Table 3-15: RPDL API function list for WDT

3.1.15. DMAC

This sample code uses Direct Memory Access Controller to transfer data from source buffer to destination buffer.

3.1.15.1. Sequence Diagram

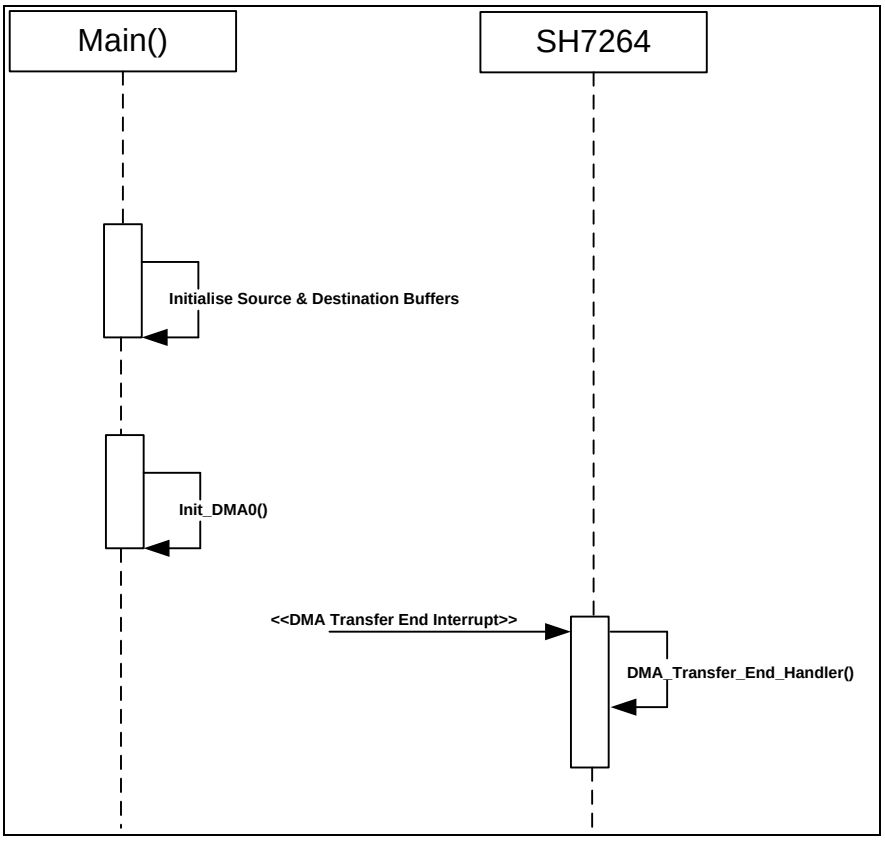

Figure 17: DMAC Sequence Diagram

3.1.15.2. Description

1- The DMAC sample application initializes source & destination buffers to default values.

2- Once DMAC is initialised, data transfer from source to destination buffer starts.

3- On data transfer completion, DMAC is configured to generate an interrupt which fires an interrupt service routine to turn ON LED one indicating DMA status.

	Function	RPDL API Function
1.	Init_DMA0	R_DMAC_Create
2.	spi_setup	R_IO_PORT_Set R_PFC_Modify

Table 3-16: RPDL API function list for DMAC

3.1.16. Ethernet Driver

This sample code uses Ethernet Driver to transmit, / receive Ethernet frames. It also runs various Ethernet configuration & operational tests, to verify proper operation of the driver.

3.1.16.1. Sequence Diagram

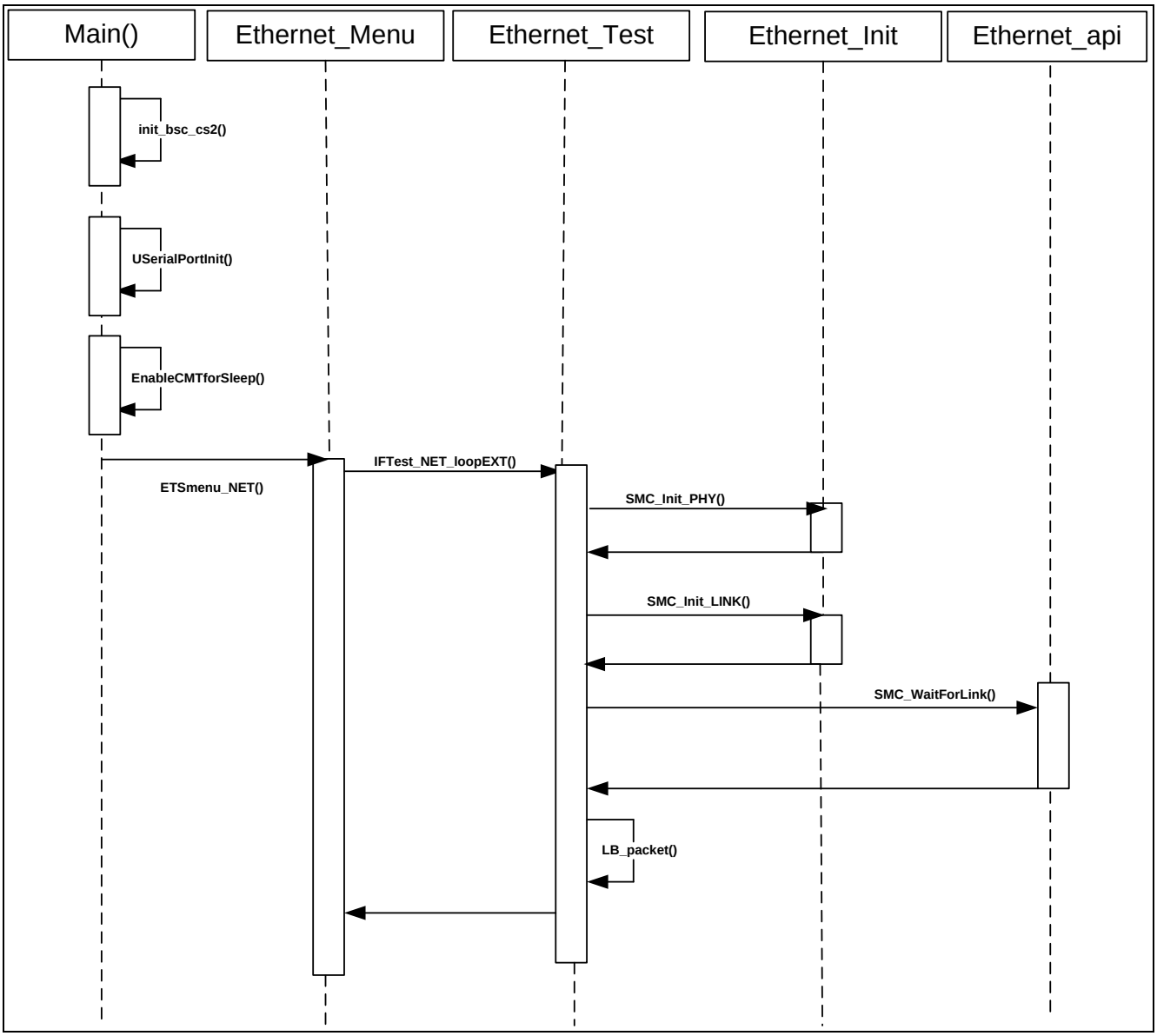

Figure 18: Ethernet Sequence Diagram for External Link Test

3.1.16.2. Description

1- The application first asks user if verbose level needs to be changed..

2- Afterwards main menu is displayed on serial terminal.

3- The menu comprises of following operational tests & configuration options for the Ethernet controller-

   a. Display the MAC address assigned to Ethernet controller.

   b. Check if Ethernet controller's interrupt is working properly.

   c. Data loopbacked from MAC layer is received correctly.

   d. Data loopbacked from PHY layer is received correctly.

   e. Data loopbacked from Ethernet cable is received correctly.

   f. Modify MAC address.

   g. Transmit ARP packet over Ethernet link.

	Function	RPDL API Function
1.	USerialPortInit	R_SCI_Create R_SCI_Receive
2.	Uputchar	R_SCI_Send

Table 3-17: RPDL API function list for Ethernet Driver

3.1.17. IIC Master

This application uses IIC master driver to read from and write to a slave EEPROM chip connected via an IIC interface.

3.1.17.1. Sequence Diagram

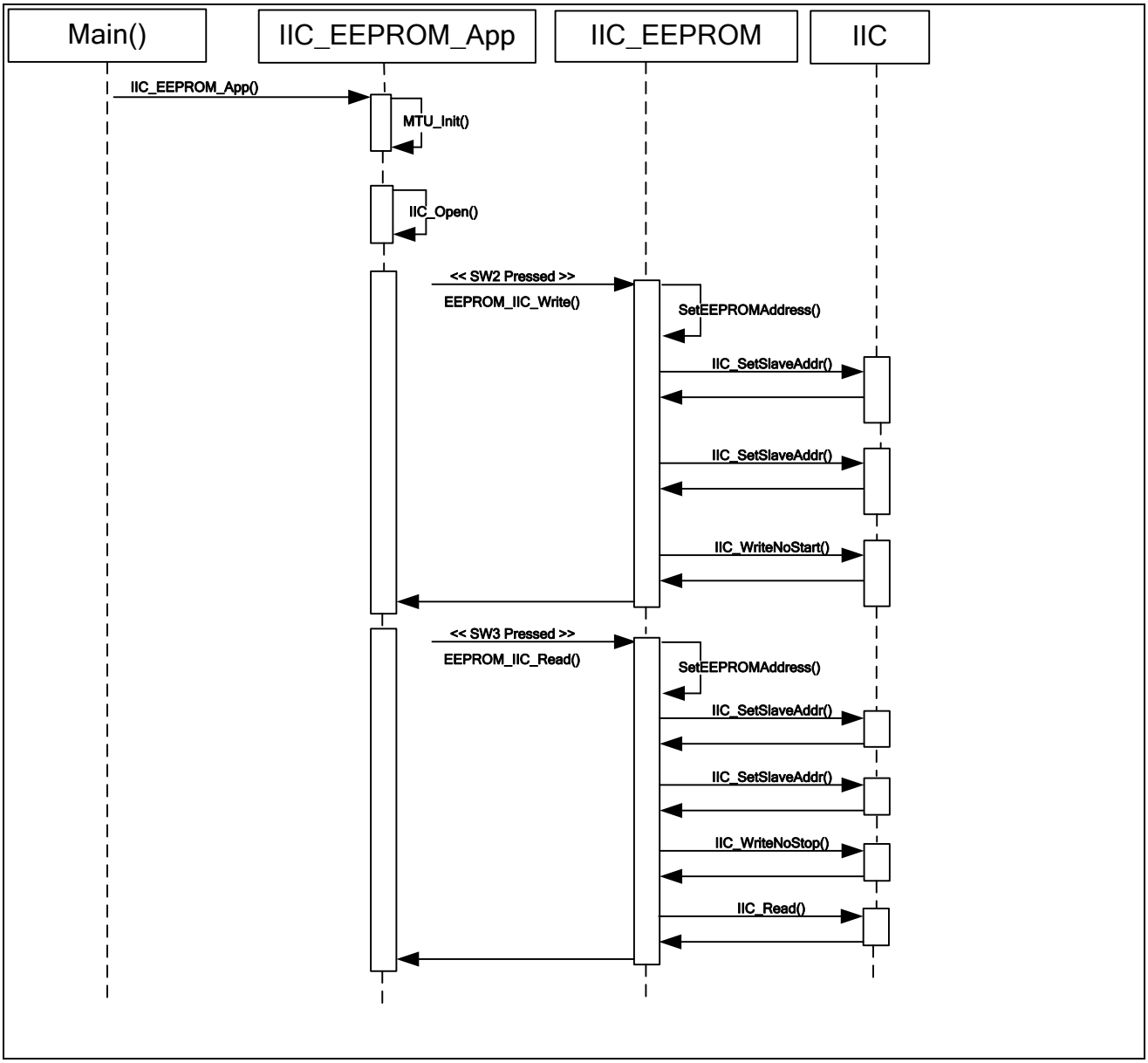

Figure 19: IIC Master Sequence Diagram for write & read operation

3.1.17.2. Description

1- The IIC application uses IIC master driver to read from and write to slave IIC EEPROM device.

2- When switch 2 is pressed, the application writes to the EEPROM memory a string of data.

3- When switch 3 is pressed, the application reads from EEPROM memory the data written in last EEPROM write operation.

	Function	RPDL API Function
1.	ConfigureSwitchTimer	R_MTU_Create R_INTC_Modify
2	Int_switch_Poll	R_MTU_Control R_ADC_10_Create R_ADC_10_Control Start_ADC_10 R_ADC_10_Read R_ADC_10_Create R_MTU_Read R_MTU_Control

Table 3-18: RPDL API function list for IIC Master

3.1.18. USB_LibUSB

This sample code uses LibUSB for communication between Host machine & target machine.

3.1.18.1. Sequence Diagram

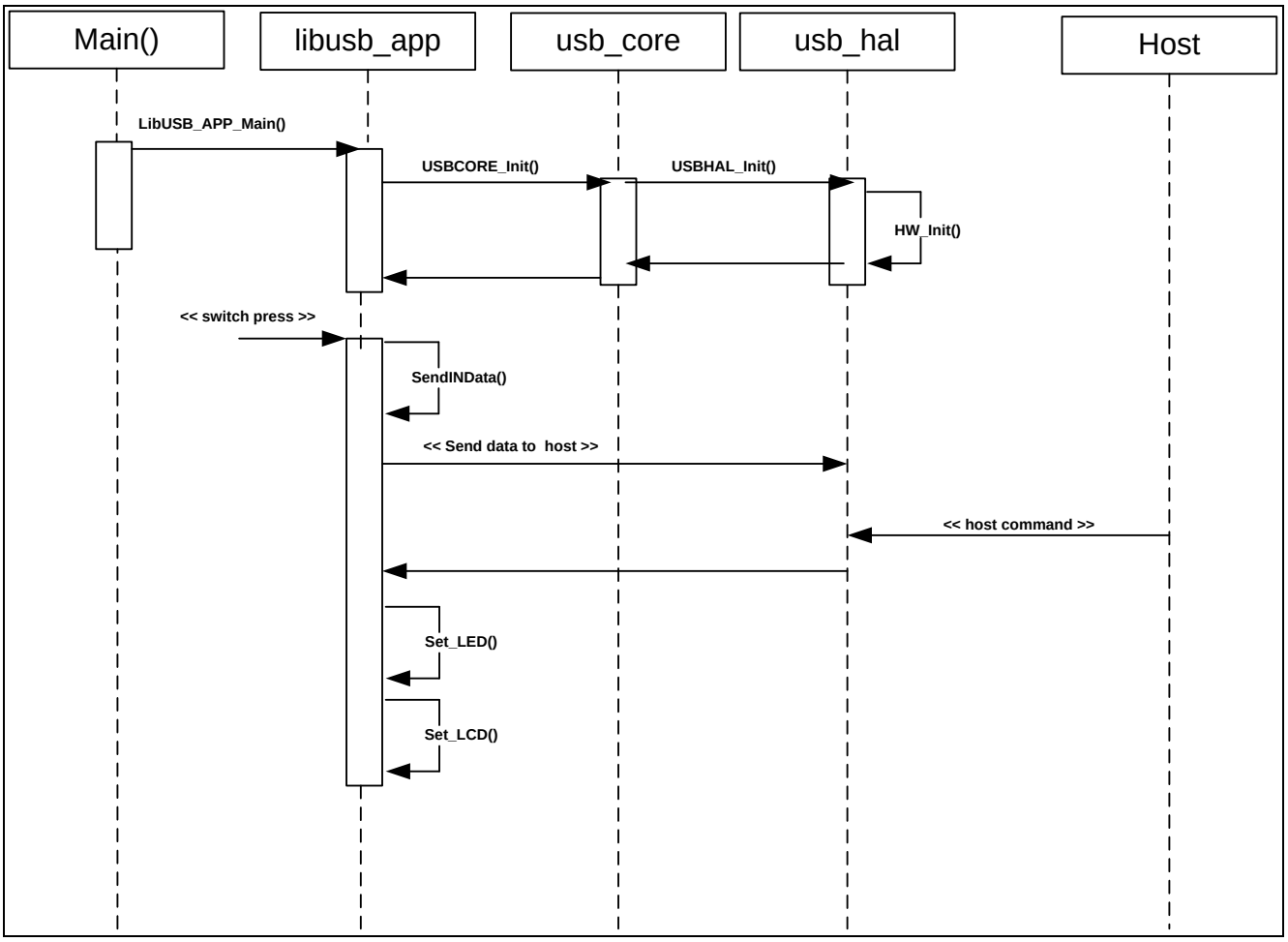

Figure 20: LibUSB Sequence Diagram

3.1.18.2. Description

1- The sample application communicates with Host machine with the help of LibUSB library.

2- It can receive user input in the form of:

   a. Changing potentiometer's pin's position on target board OR

   b. Command from Host machine's application.

3- It can operate three peripheral devices:

   a. LED

   b. LCD

   c. ADC.

4- Host machine's application can be used to switch ON or OFF LED's.

5- Host machine's application can be used to display character string to LCD display.

6- Host machine's application can be used to read ADC value.

	Function	RPDL API Function
1	ConfigureADC	R_ADC_10_Create
2	DecipherOUT	R_ADC_10_Control Start_ADC_10 R_ADC_10_Read
3	read_LED	R_IO_PORT_ReadControl
4	set_PIN	R_IO_PORT_Wr_Pin_H
5	read_PIN	R_IO_PORT_Read
6	toggle_LED	R_IO_PORT_Modify
7	io_init_scif()	R_SCI_Create
8	sio_write()	R_SCI_Send

Table 3-19: RPDL API function list for LibUSB

3.1.19. USB_MSC

This sample code uses USB Mass Storage Class (MSC) to emulate the SH7264 board as a mass storage device.

3.1.19.1. Sequence Diagram

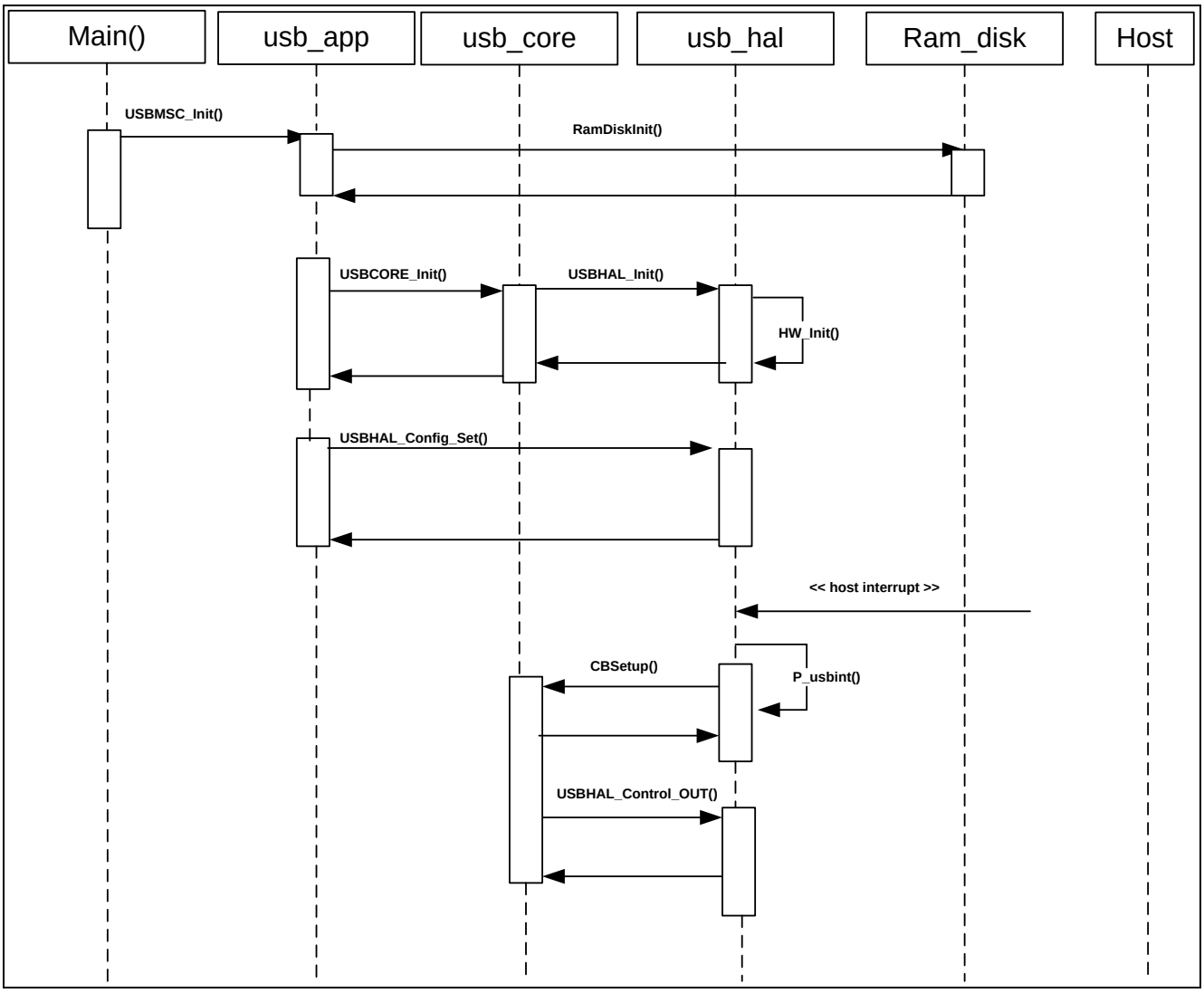

Figure 21: USB_MSC Sequence Diagram

3.1.19.2. Description

1- The sample application uses USB mass storage class to emulate RSK2+SH7264 board as a mass storage device to host machine.

2- Once connection is established, the RSK is configured at host machine as a memory drive.

3- Files can be created, edited & destroyed on this memory drive from the host machine like any other mass storage device.

4- The device uses a RAM disk & FAT file system to store its data upto 4 KB of size.

	Function	RPDL API Function
1	io_init_scif()	R_SCI_Create
2	sio_write()	R_SCI_Send

Table 3-20: RPDL API function list for USB MSC

3.1.20. USB_HID

This application demonstrates the Human Interface Device Class USB support of SH7264 on-chip USB module.

3.1.20.1. Sequence Diagram

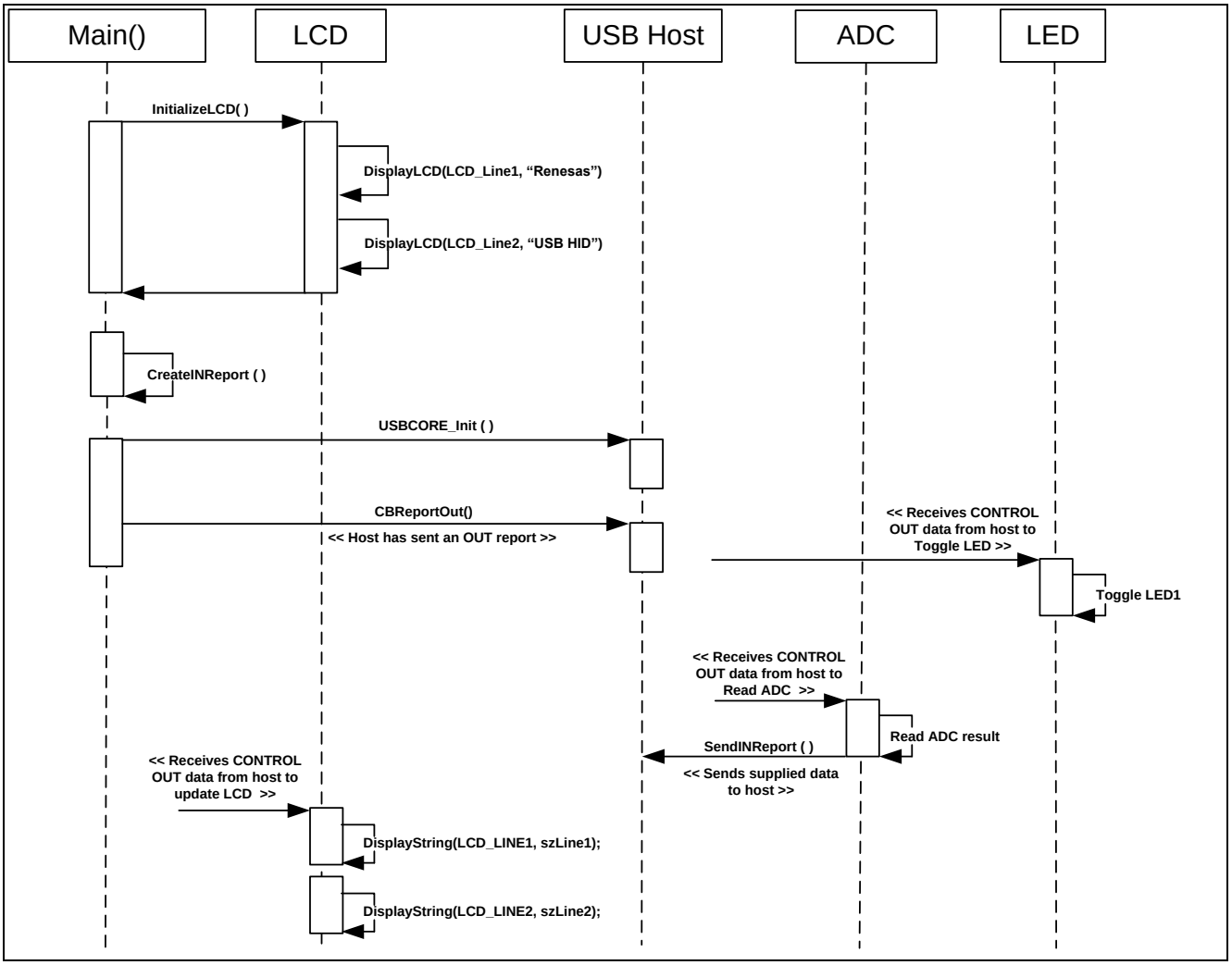

Figure 22: USB_HID Sequence Diagram

3.1.20.2. Description

1- The HID host sample application is configured for communication between a Windows host PC and RSK.

2- The application initializes USB port such that windows will detect the new device and automatically load the intrinsic HID class driver

3- The process is called to make connection from application to the target.

4- After successful connection between host and RSK using 'Connect' button, the information about the device is displayed on Windows application.

5- The "Toggle LED" button enables a LED on the RSK to be toggled on and off.

6- The "Read ADC" button will command the RSK to read its ADC and return the value back to the host where it will be displayed.

7- The "Set LCD" button allows the text of the LCD on the RSK to be changed.

	Function	RPDL API Function
1	USBHID_Init ( )	R_ADC_10_Create R_ADC_10_Control R_ADC_10_Read R_IO_PORT_Modify R_IO_PORT_Wr_Pin_H R_CMT_Create R_CMT_Read
2	io_init_scif()	R_SCI_Create
3	sio_write()	R_SCI_Send

Table 3-21: RPDL API function list for USB_HID

3.1.21. USB_CDC

The CDC sample application uses communication device class to demonstrate communication with a Windows PC using a standard terminal program.

3.1.21.1. Sequence Diagram

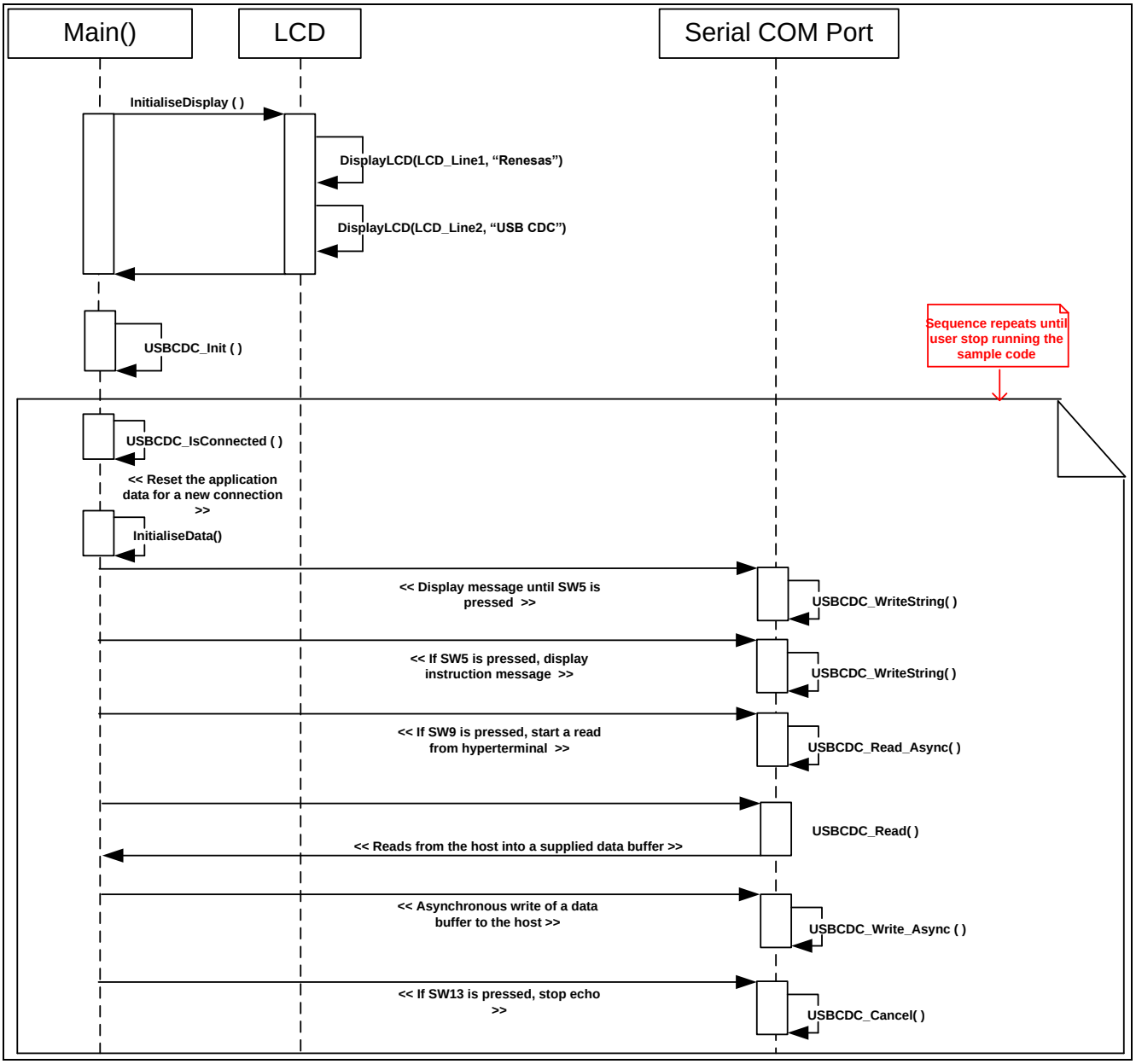

Figure 23: USB_CDC Sequence Diagram

3.1.21.2. Description

1- The CDC host sample application is configured for communication between a Windows host PC and RSK.

2- The application initializes USB port such that windows will detect the new device and run the “Found New Hardware Wizard”.

3- After completing the installation of the CDC USB driver, following repeating message will be displayed on the hyperterminal program.

“Renesas USB CDC Sample, Press Switch SW5.”

4- To demonstrate two-way communication pressing SW9 will put the RSK into echo mode. Anything typed on the Terminal will be read by the RSK and then echoed back to the terminal.

5- Pressing SW13 will cancel this echo mode.

	Function	RPDL API Function
1	USBCDC_Init	R_SCI_Create R_ADC_10_Read R_ADC_10_Control R_MTU_Read R_MTU_Control
2	io_init_scif()	R_SCI_Create
3	sio_write()	R_SCI_Send

Table 3-22: RPDL API function list for USB_CDC

Chapter 4. Additional Information

For details on how to use High-performance Embedded Workshop (HEW), refer to the HEW manual available on the CD or installed in the Manual Navigator.

For information about the SH/7264 microcontrollers refer to *the SH/7264 Group Hardware Manual*

For information about the SH/7264 assembly language, refer to the *SH-2A SH2A-FPU Software Manual*

For information about the E10A Emulator, please refer to the *E10A-USB Emulator User's Manual*

Online technical support and information is available at: www.renesas.com/renesas_starter_kits

Technical Contact Details

America: techsupport.rta@renesas.com

Europe: tools.support.eu@renesas.com

Japan: csc@renesas.com

General information on Renesas Microcontrollers can be found on the Renesas website at: www.renesas.com

Revision History	Renesas Starter Kit for SH7216 Software Manual

Rev.	Date	Description	
		Page/Section	Summary
0.1	03-12-2009	ALL	First Draft copy for review.

Renesas Starter Kit
Software Manual
Publication Date Rev.1.0 17.DEC.2009
Published by: Renesas Technology Europe Ltd.
Duke's Meadow, Millboard Road, Bourne End
Buckinghamshire SL8 5FH, United Kingdom

Renesas Starter Kit for SH7264
Software Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REG10J0175-0100