

Using Guiliani RZ/A3UL SDK in e2Studio

Product:	SDK Guiliani for Renesas RZ/A3UL
Release version:	2.6
Release date:	April 26, 2024

Table of Contents

1	Introduction	3
2	Assumed Knowledge.....	3
3	Prerequisites	3
4	Configuring the Peripherals	4
4.1	Changing display resolution	4
4.2	Switch from mouse to touchscreen.....	6
5	e2Studio Projects.....	7
5.1	Board Support Package (BSP).....	7
5.1.1	Directory structure.....	8
5.1.2	Build configurations	9
5.2	BSP_Test	10
5.2.1	Directory structure.....	10
5.2.2	Build configurations	10
5.3	SR_GuilianiDemo	11
5.3.1	Directory structure.....	11
5.3.2	Build configurations	12
5.4	StreamRuntime	13

5.4.1	Directory structure.....	13
5.4.2	Build configurations.....	13
6	Debug Configurations	15
7	Annex	16
7.1	Startup sequence of Guiliani Demo application.....	16

List of Figures

Fig. 1	Settings for 1280x720.....	4
Fig. 2	Settings for 1920x1080.....	5
Fig. 3	e2Studio Workspace of SDK Project.....	7
Fig. 4	Debug Configurations.....	15
Fig. 5	Startup Sequence of Guiliani Demo Application.....	16

List of Tables

Table 1	Directory Structure of BSP Project.....	8
Table 2	Directory Structure of BSP_Test Project	10
Table 3	Directory Structure of <SDK>\SR_GuilianiDemo	11
Table 4	Files in <SDK>\SR_GuilianiDemo\Common Directory	11
Table 5	Files in <SDK>\SR_GuilianiDemo\Include and <SDK>\SR_GuilianiDemo\Source Directory.....	11
Table 6	Linker Script in <SDK>\Renesas\BSP\script\	11
Table 7	Files in <SDK>\Guiliani\Share Directory.....	11
Table 8	Directory Structure of <SDK>\StreamRuntime.....	13
Table 9	Files in <SDK>\StreamRuntime\Common Directory.....	13
Table 10	Files in <SDK>\StreamRuntime\Include and <SDK>\StreamRuntime\Source Directory	13
Table 11	Linker Script in <SDK>\Renesas\BSP\script\	13
Table 12	Files in <SDK>\Guiliani\Share Directory.....	13

1 Introduction

The SDK for Renesas RZ/A3UL boards contains several e2Studio projects, which can be used for editing and debugging the own Guiliani application.

e2studio is an eclipse based Integrated Development Environment (IDE). This document describes the different projects, their directory structure and the build configurations included in the e2studio project workspace of the demo.

This guide does not explain how to create an e2Studio project and configure the settings. It rather explains the e2Studio projects, which are already created and included in the SDK so that the user can quickly test the Guiliani demo and do the changes according to his requirements.

2 Assumed Knowledge

- Basic to advanced knowledge of C and C++
- General understanding and hands-on experience of e2Studio or eclipse (If you are not familiar with any of these tools, we recommend you to read “User’s Manual: Getting Started Guide” of e2Studio, available on Renesas website)
- General understanding of the Flexible Software Package (FSP) for RZ/A3UL

3 Prerequisites

- Installed Guiliani SDK for Renesas RZ/A3UL
- Installed e2Studio 2023-04
- Installed Renesas RZ/A FSP 3.0.0

4 Configuring the Peripherals

4.1 Changing display resolution

By default the resolution of the HDMI is set to 1280x720. If a different resolution is required it can be changed by the following steps:

1. Change the timing of the display output in configuration.xml:

In the Stacks-configuration select the module g_display0 in HAL/Common and change the values in Output/Timing.

Property	Value
▼ Module g_display0 Display Driver on r_lcdc	
> General	
> Interrupts	
> Input	
▼ Output	
▼ Timing	
Horizontal total cycles	1650
Horizontal active video cycles	1280
Horizontal back porch cycles	260
Horizontal sync signal cycles	40
Horizontal sync signal polarity	High active
Vertical total lines	750
Vertical active video lines	720
Vertical back porch lines	25
Vertical sync signal lines	5
Vertical sync signal polarity	High active
Data Enable Signal Polarity	High active
Sync edge	Rising edge
> Background	
> Dithering	
> Color Keying	

Fig. 1 Settings for 1280x720

Property	Value
▼ Module g_display0 Display Driver on r_lcd	
> General	
> Interrupts	
> Input	
▼ Output	
▼ Timing	
Horizontal total cycles	2200
Horizontal active video cycles	1920
Horizontal back porch cycles	192
Horizontal sync signal cycles	44
Horizontal sync signal polarity	High active
Vertical total lines	1125
Vertical active video lines	1080
Vertical back porch lines	41
Vertical sync signal lines	5
Vertical sync signal polarity	High active
Data Enable Signal Polarity	High active
Sync edge	Rising edge
> Background	
> Dithering	
> Color Keying	

Fig. 2 Settings for 1920x1080

2. Change the initialization of the display registers in the function `bsp_lcd_init` in the file `BSP/src/rza3ul_smarc/rza3ul_smarc_lcd.c`.

For 1280x720 and 1920x1080 the values are already present and can be switched by using the `#if` statement.

Note: if you need a different resolution please refer to the documentation of the board or contact Renesas for more information.

4.2 Switch from mouse to touchscreen

By default the BSP will assume that a mouse is connected to the USB-interface. If you want to use a touch-screen you can switch to this input-device.

In the file `<APP>/Common/Source/Platform/FreeRTOS/StreamRuntime_RZA3UL_Funcs.cpp` there is a function “InputInit” where the setup of the input-device is made.

You can either call “mouse_init” or “touch_init” with the appropriate parameters to switch from touch to mouse.

Note: if the used touch-screen does not work out-of-the-box it may send data in an unsupported format. Please refer to the manual and change the driver accordingly. Especially the values for the touch-resolution are important.

5 e2Studio Projects

several e2Studio-projects are available in the SDK within the Renesas subfolder. Launch e2Studio IDE and import the projects into your workspace.

The following four projects are available (Fig. 1):

- BSP: Renesas Board Support Package (BSP) files for RZ/A3UL Evaluation Board Kit
- BSP_Test: A test project to quickly test BSP without Guiliani
- SR_GuilianiDemo: The Guiliani Technical Showcase Demo
- StreamRuntime: The StreamRuntime-application

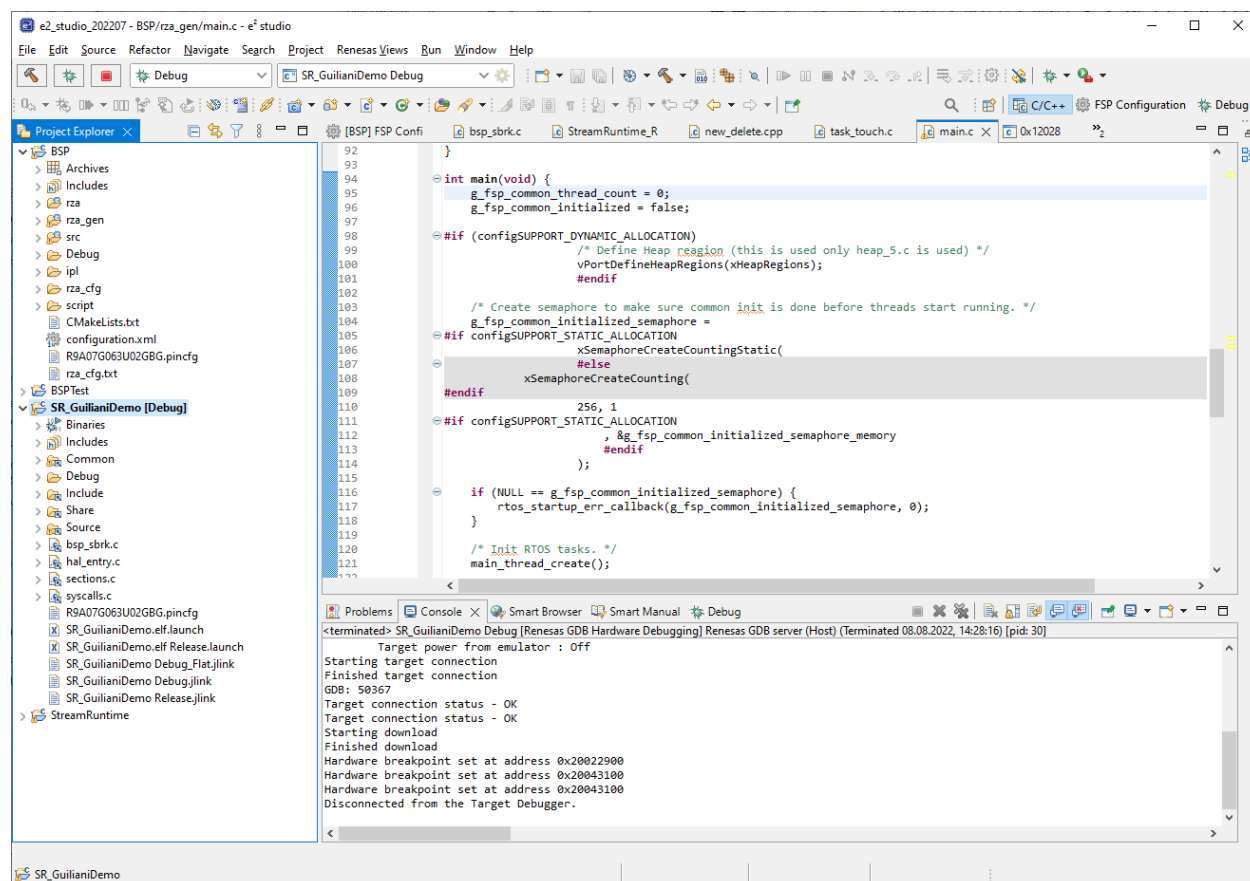


Fig. 3 e2Studio Workspace of SDK Project

5.1 Board Support Package (BSP)

This SDK includes the BSP for the RZ/A3UL Evaluation Board Kit. The BSP contains initialization code for clocks, RAM, caches and peripherals which are specific to the boards. It also includes driver files and a FreeRTOS port for the evaluation board kit.

5.1.1 Directory structure

Directory	Description
rza/aws	Port for FreeRTOS
rza/board	Board-specific files
rza/fsp	FSP-based files for the BSP
rza_gen	Generated files from the FSP
src	Source files for various parts (LCD, Mouse)
ipl	The initial program launcher (IPL)
rza_cfg	Configuration files for FreeRTOS and FSP
script	Linker-scripts and post-build

Table 1 Directory Structure of BSP Project

5.1.2 Build configurations

- **Debug:** It builds board support package for RZ/A3UL Evaluation Board Kit in debug mode. When the project is built, it creates a library libBSP.a in a subfolder Debug, which can be used by the other projects.
- **Release:** It builds board support package for RZ/A3UL Evaluation Board Kit in release mode. When the project is built, it creates a library libBSP.a in a subfolder Release, which can be used by other projects.

5.2 BSP_Test

This project allows a user to quickly test BSP of RZ/A3UL Evaluation Board Kit without the need of the Guiliani application. The test program can be flashed on the board and can be debugged. It is a simple blinking application.

5.2.1 Directory structure

Directory	Description
*.c	C source files from the BSP used for explicit linkage
main_thread_entry.cpp	entry for main-thread of the application

Table 2 Directory Structure of BSP_Test Project

5.2.2 Build configurations

- Debug: BSP_Test program from flash of RZ/A3UL Evaluation Board Kit in Debug mode.
- Release: BSP_Test program from flash of RZ/A3UL Evaluation Board Kit in Release mode.

5.3 SR_GuilianiDemo

This project contains the files required for porting Guiliani on the Renesas board.

5.3.1 Directory structure

Directory	Description
Common	Common files over different Guiliani applications
Include	Project specific includes
Source	Project specific sources

Table 3 Directory Structure of <SDK>\SR_GuilianiDemo

File	Description
Platform/*/StreamRuntime*.[c cpp h]	Program entry points (main function) for different platforms
[Include Source]/Platform/FreeRTOS/StreamRuntimeStartup.[cpp h]	Target specific initialization of wrappers and configurations
[Include Source]/Platform/win/pc/StreamRuntimeStartup.h"	Windows specific initialization of wrappers and configurations
[Include Source]/StreamRuntimeConfig.[h cpp]	Loads project configuration
[Include Source]/StreamRuntimeGUI.[h cpp]	Loads GUI

Table 4 Files in <SDK>\SR_GuilianiDemo\Common Directory

File	Description
CustomExtension	Custom extensions.
GUIConfig/User*Resource.h	Resource IDs generated by GSE
GUIConfigCustom/*	Custom IDs for use in Guiliani application.
Demo_*.[cpp h]	Specific code for the different demo parts
MyGUI_SR.[cpp h]	GUI entry point

Table 5 Files in <SDK>\SR_GuilianiDemo\Include and <SDK>\SR_GuilianiDemo\Source Directory

File	Description
fsp_qspi.ld	Linker script for RZ/A3UL Evaluation Board Kit
fsp_sdram.ld	Linker script to select RAM instead of flash

Table 6 Linker Script in <SDK>\Renesas\BSP\script\

File	Description
GUIConfig.cpp	This file contains constants which hold the count of global properties, image resources, font resources, text resources, etc.

Table 7 Files in <SDK>\Guiliani\Share Directory

5.3.2 Build configurations

There are two configurations available for SR_GuilianiDemo project.

1. **Debug:** Debug configuration for RZ/A3UL Evaluation Board Kit. The demo application runs from QSPI flash. Choose this configuration to debug the application.
2. **Release:** Release configuration for RZ/A3UL Evaluation Board Kit. The application runs from QSPI flash. Choose this configuration to test the performance.

5.4 StreamRuntime

This project contains the files required for porting Guiliani on the Renesas board.

5.4.1 Directory structure

Directory	Description
Common	Common files over different Guiliani applications
Include	Project specific includes
Source	Project specific sources

Table 8 Directory Structure of <SDK>\StreamRuntime

File	Description
Platform/*/StreamRuntime*.[cpp h]	Program entry points (main function) for different platforms
[Include Source]/Platform/FreeRTOS/StreamRuntimeStartup.[cpp h]	Target specific initialization of wrappers and configurations
[Include Source]/Platform/win/pc/StreamRuntimeStartup.h"	Windows specific initialization of wrappers and configurations
[Include Source]/StreamRuntimeConfig.[h cpp]	Loads project configuration
[Include Source]/StreamRuntimeGUI.[h cpp]	Loads GUI

Table 9 Files in <SDK>\StreamRuntime\Common Directory

File	Description
CustomExtension	Custom extensions.
GUIConfig/User*Resource.h	Resource IDs generated by GSE
GUIConfigCustom/*	Custom IDs for use in Guiliani application.
Demo_*. [cpp h]	Specific code for the different demo parts
MyGUI_SR.[cpp h]	GUI-application entry point

Table 10 Files in <SDK>\StreamRuntime\Include and <SDK>\StreamRuntime\Source Directory

File	Description
fsp_qspi.ld	Linker script for RZ/A3UL Evaluation Board Kit
fsp_sdram.ld	Linker script to select RAM instead of flash

Table 11 Linker Script in <SDK>\Renesas\BSP\script\

File	Description
GUIConfig.cpp	This file contains constants which hold the count of global properties, image resources, font resources, text resources, etc.

Table 12 Files in <SDK>\Guiliani\Share Directory

5.4.2 Build configurations

There are two configurations available for StreamRuntime project.

1. Debug: Debug configuration for RZ/A3UL Evaluation Board Kit. The demo application runs from QSPI flash. Choose this configuration to debug the application.
2. Release: Release configuration for RZ/A3UL Evaluation Board Kit. The application runs from QSPI flash. Choose this configuration to test the performance.

6 Debug Configurations

Under *Run* → *Debug Configurations* → *Renesas GDB Hardware Debugging* menu of e2Studio, debug configurations are created for each build configuration present in e2Studio workspace (Fig. 2). The name of each debug configuration is a combination of the project name and its build configuration. For example *SR_GuilianiDemo HardwareDebug* configuration is for project *SR_GuilianiDemo* with *HardwareDebug* configuration.

After a project is built, its debug configuration can be launched by clicking on button *Debug*. This will flash the binary file on the board and start debugging.

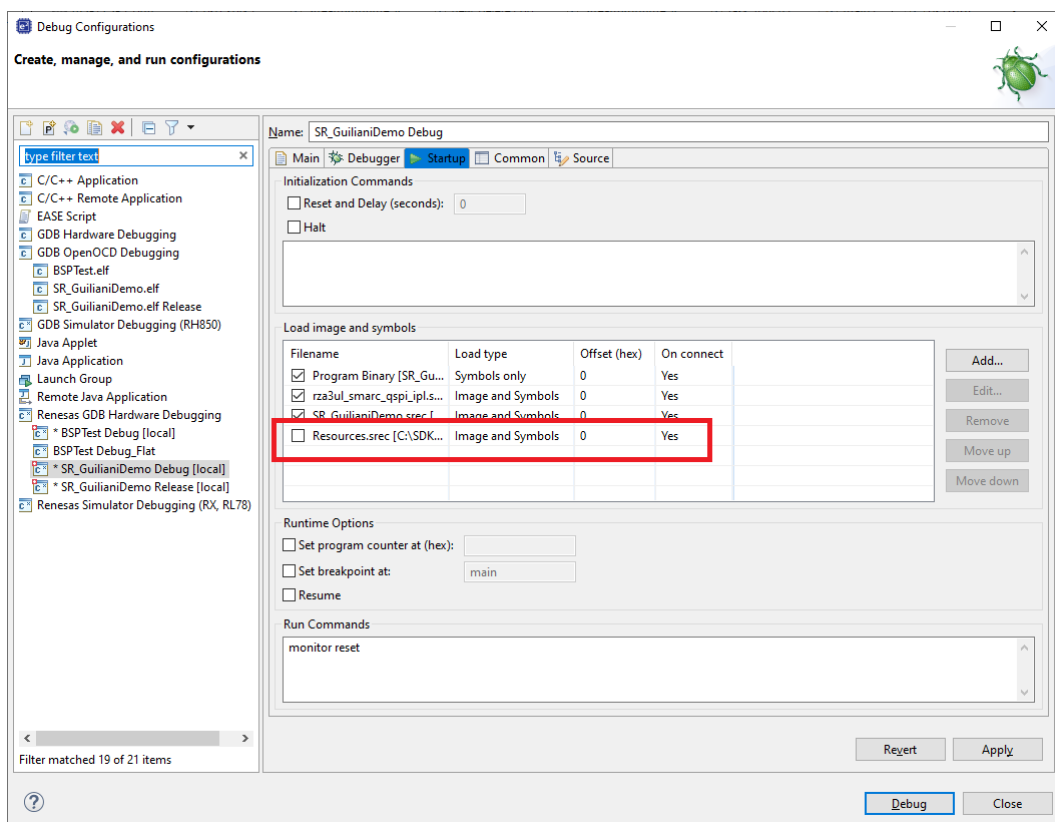


Fig. 4 Debug Configurations

Note: if the resource-file was not changed and should not be downloaded to the board, deselect the item “Resources.srec” in the “Load image and symbols”-list. This will decrease the time needed for flashing.

7 Annex

7.1 Startup sequence of Guiliani Demo application

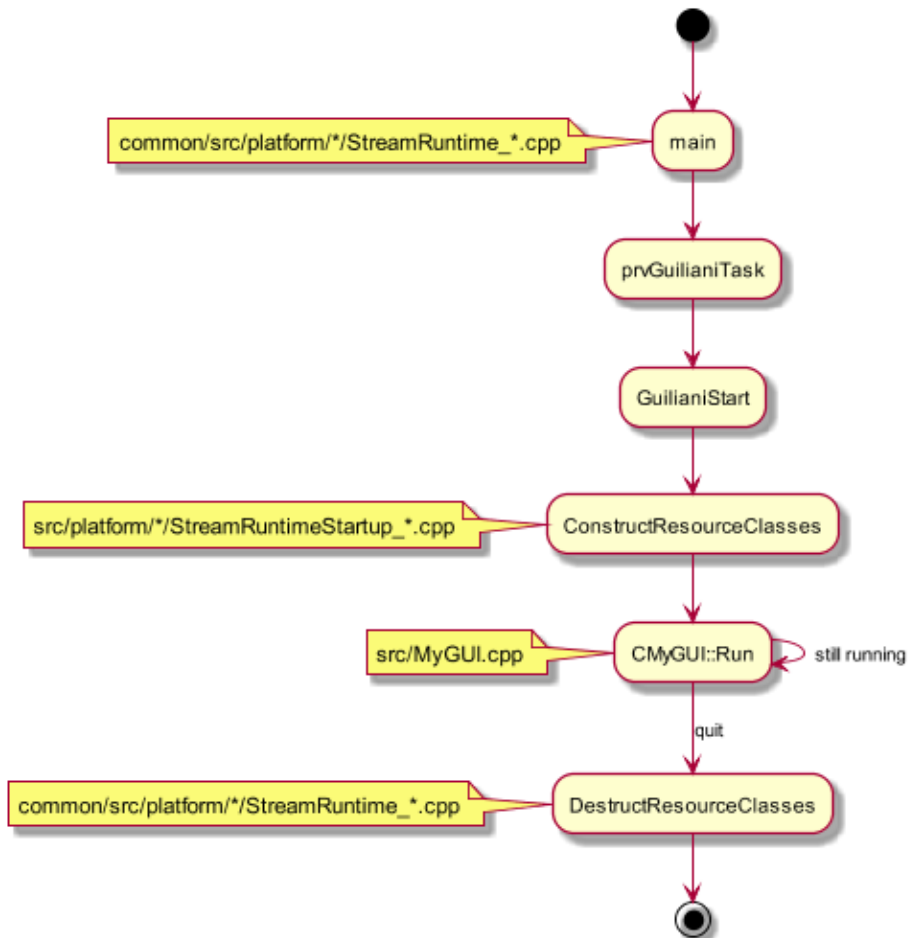


Fig. 5 Startup Sequence of Guiliani Demo Application