

# Using Guiliani RA SDK With Renesas RA8D1-EK within e2Studio

Product:	SDK Guiliani for Renesas RA8D1-EK
Release version:	2.6
Release date:	September 2, 2024

## **Table of Contents**

1	Introduction .....	3
2	Assumed Knowledge.....	3
3	Prerequisites .....	3
4	System Preparation.....	4
4.1	Migration steps for FSP .....	5
5	Run GuilianiDemo .....	5
5.1	Flash application and resources .....	5
5.1.1	Import and compile the application.....	5
5.1.2	Debug application and download resources.....	8
5.1.3	First time startup.....	10
5.2	Demo description.....	11
6	Edit example project on the computer.....	11
7	Load edited example project onto the board .....	11
8	e2Studio Workspace .....	13
8.1	Board Support Package (BSP).....	13
8.1.1	Directory structure.....	14
8.1.2	Build configurations .....	14
8.2	SR_GuilianiDemo .....	14

8.2.1	Directory structure.....	14
8.2.2	Build configurations.....	15
8.3	StreamRuntime.....	15
8.3.1	Directory structure.....	15
8.3.2	Build configurations.....	16
9	Debug Configurations.....	16
10	Annex.....	17
10.1.1	Startup sequence of Guiliani Demo application.....	17

## **List of Figures**

Fig. 1	Connect via USB.....	4
Fig. 2	Import project.....	5
Fig. 3	Import project (2).....	6
Fig. 4	Import project (3).....	6
Fig. 5	Generate BSP Contents.....	7
Fig. 6	Start debug.....	8
Fig. 7	Files to download.....	9
Fig. 8	First time startup.....	10
Fig. 9	TrustZone settings.....	10
Fig. 10	Export Option.....	11
Fig. 11	Exporting Configuration.....	12
Fig. 12	e2Studio Workspace of SDK Project.....	13
Fig. 13	Startup Sequence of Guiliani Demo Application.....	17

## **List of Tables**

Table 1	Directory Structure of BSP Project.....	14
Table 2	Directory Structure of <SDK>\SR_GuilianiDemo.....	14
Table 3	Files in <SDK>\SR_GuilianiDemo\Common Directory.....	14
Table 4	Files in <SDK>\SR_GuilianiDemo\Include and <SDK>\SR_GuilianiDemo\Source Directory.....	15
Table 5	Files in <SDK>\Guiliani\Share Directory.....	15
Table 6	Directory Structure of <SDK>\StreamRuntime.....	15
Table 7	Files in <SDK>\StreamRuntime\Common Directory.....	15
Table 8	Files in <SDK>\StreamRuntime\Include and <SDK>\StreamRuntime\Source Directory.....	16
Table 9	Files in <SDK>\GSE\Share Directory.....	16

## 1 Introduction

The “GuilianiDemo” presents some of the capabilities of the Guiliani HMI framework running on the Renesas RA8D1-EK. It uses the TES D/AVE accelerated rendering-engine (aka “2D Drawing Engine”). Additionally the efficient way of working for GUI application development by using the Guiliani Streaming Editor (GSE) is introduced.

The SDK for RA8D1-EK contains an e2Studio project, which can be used for editing and debugging the Guiliani demo. 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 an e2Studio workspace, which is 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.

For an introduction to the Renesas RA MCU Family and related topics, we recommend reading the “Official Renesas RA Family Beginner’s Guide” by Richard Oed. It can be found on [www.renesas.com](http://www.renesas.com).

## 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)

## 3 Prerequisites

- Installed Guiliani SDK for RA8D1-EK
- Installed e2Studio 2024-04 with FSP 5.4.0
- GNU MCU Eclipse Windows Build Tools (2.9-20170629-1013)

## 4 System Preparation

- Connect the RA8D1-EK target (DEBUG1) via an Micro-USB cable to a Windows PC to power it up. This connector also serves as the direct JLink-connection

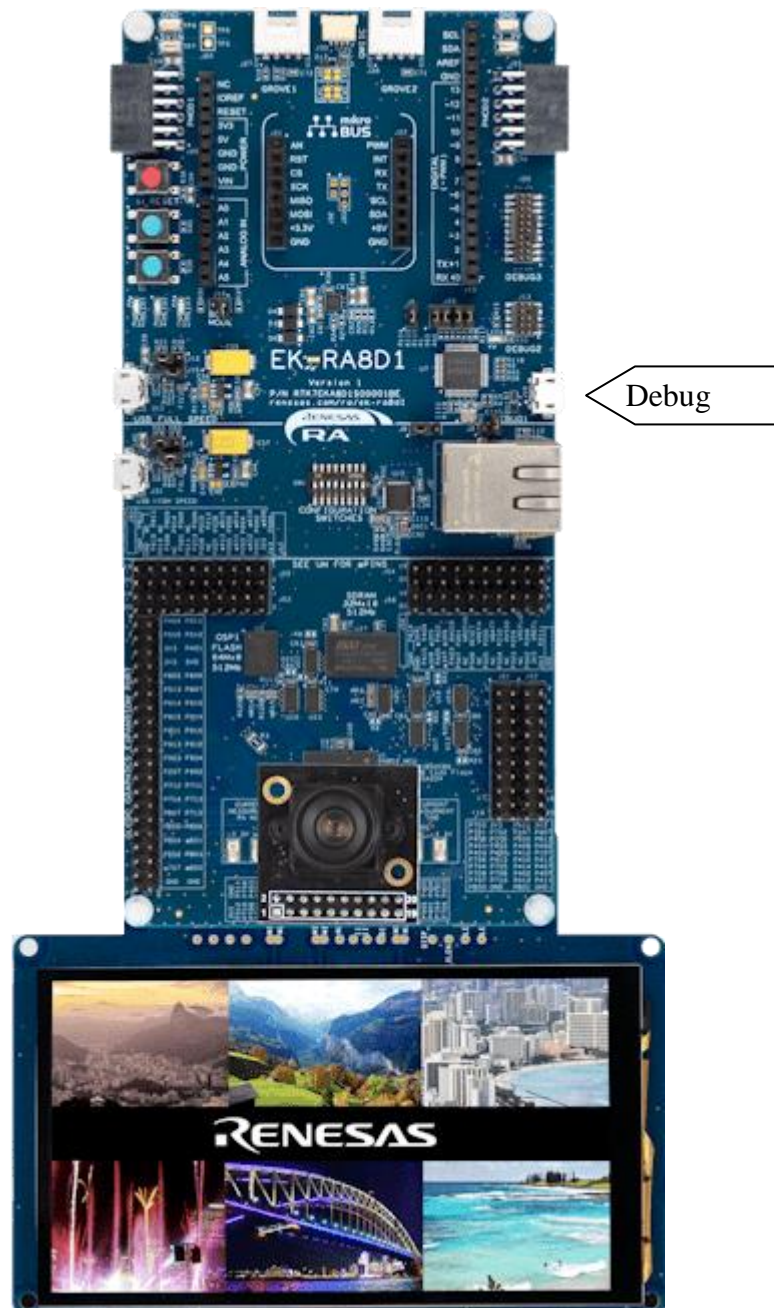


Fig. 1 Connect via USB

## 4.1 Migration steps for FSP

If you are using a different version of the FSP than FSP 5.4.0 there might be some changes necessary to compile the projects without errors. Please refer to <https://github.com/renesas/fsp/releases> on how to do the necessary changes.

## 5 Run GuilianiDemo

### 5.1 Flash application and resources

To get ready running GuilianiDemo on your target-board the following steps are necessary:

- Open e2studio

#### 5.1.1 Import and compile the application

Open the import dialog with “File->Import”

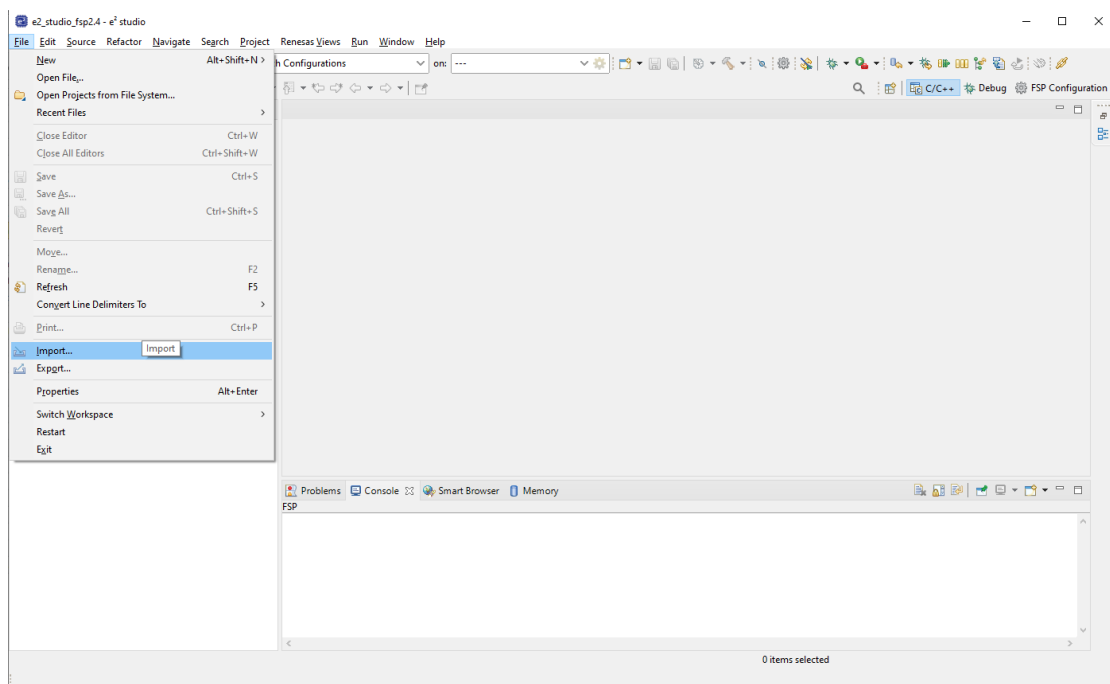
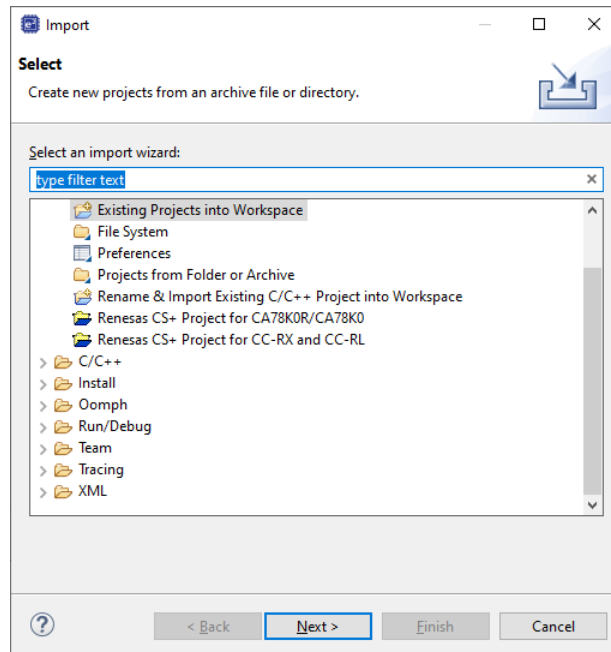


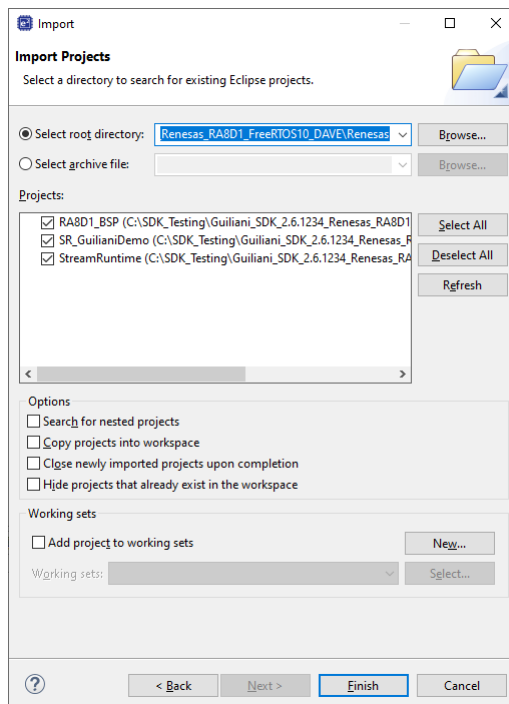
Fig. 2 Import project

Select “Existing Projects into Workspace” and click Next



**Fig. 3 Import project (2)**

Use “Browse” to select the Renesas-subfolder of the SDK and click on “Finish”



**Fig. 4 Import project (3)**

The projects are now imported into the workspace

Double-click on the file “configuration.xml” in the BSP-project to open it in the FSP-Configurator and click “Generate Project Contents” to create necessary files for the BSP.

**Note: loading the xml-file may take some time. Please be patient, otherwise errors might occur which might damage the project.**

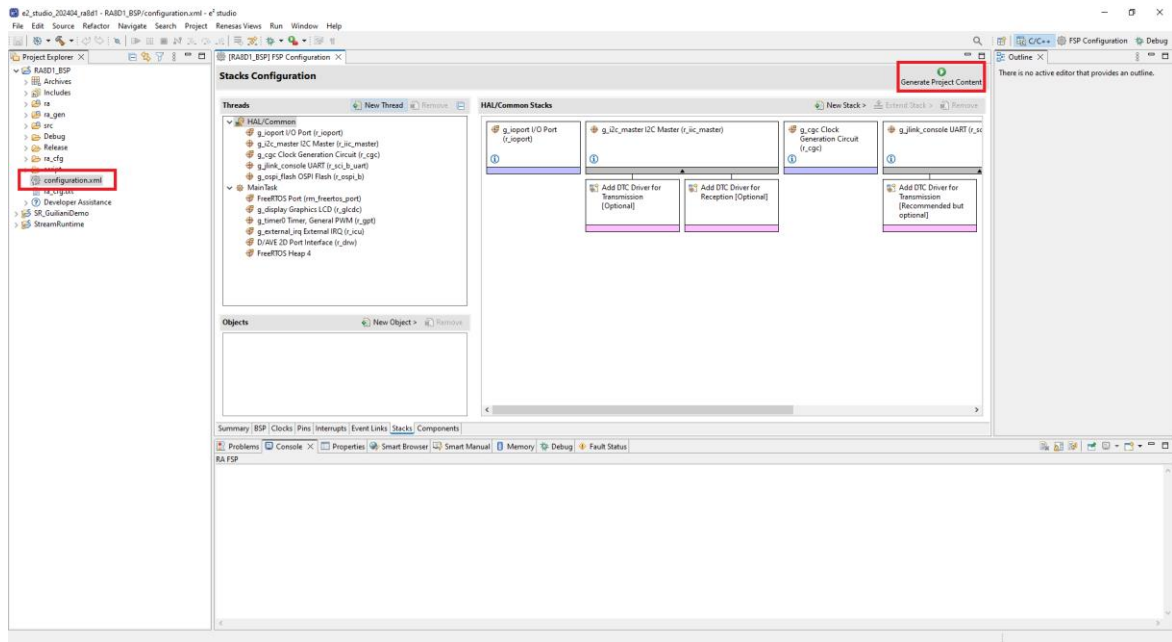


Fig. 5 Generate BSP Contents

Build the project BSP and the after that the application.

## 5.1.2 Debug application and download resources

Start a debugging-session for the application which will download the binary and resources to the board. Just right-click on the project you want to debug and select “Debug As” and then “Debug Configuration...”.

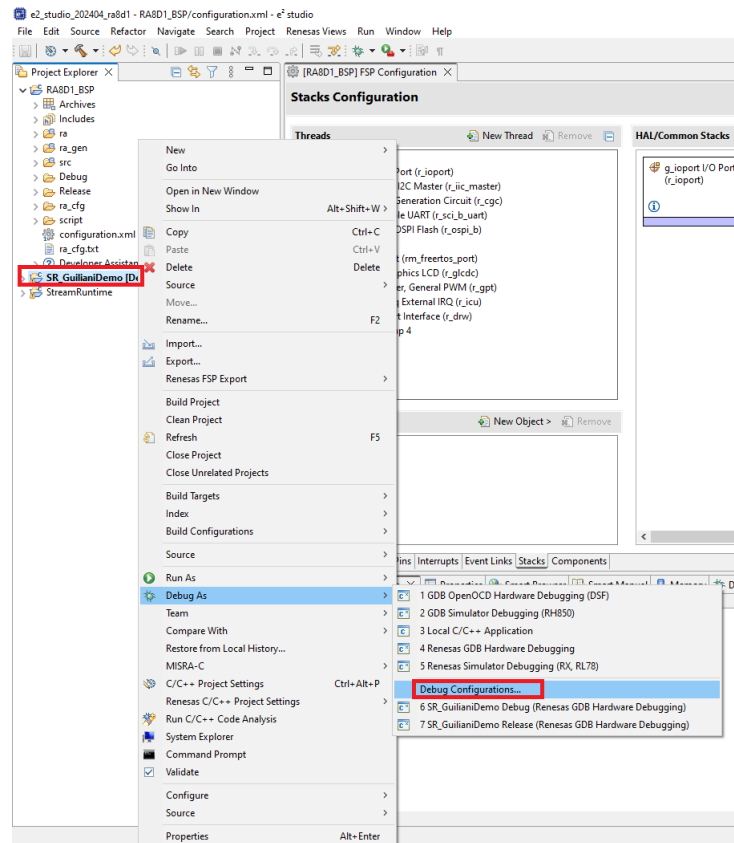
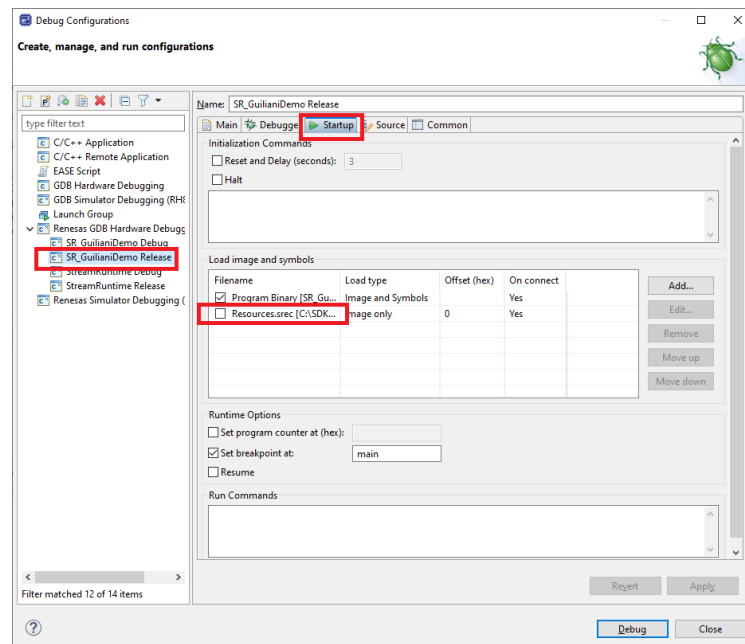


Fig. 6 Start debug



This will open a dialog showing the existing Debug Configurations.

Select the desired configuration and click on the Startup-tab on the right side to show see the files which are downloaded to the board.



**Fig. 7 Files to download**

If you don't want to flash the resources every time a debug starts you can de-select the Resources.srec file in the list.

### 5.1.3 First time startup

When first flashing a Guiliani application the TrustZone settings need to be updated on the board. Click on the menu “Run – Renesas Debug Tools” and select the “Renesas Device Partition Manager”.

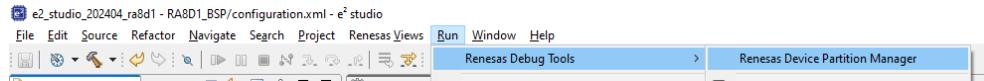


Fig. 8 First time startup

In the dialog fill in the settings shown below and press on the “Run”-button. After the download is complete you can start a debug-session as usual.

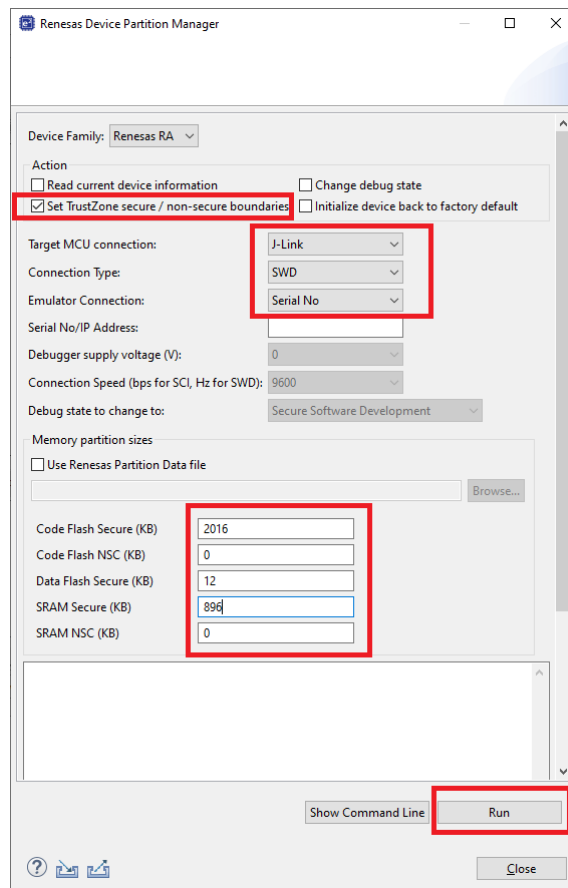


Fig. 9 TrustZone settings

## 5.2 Demo description

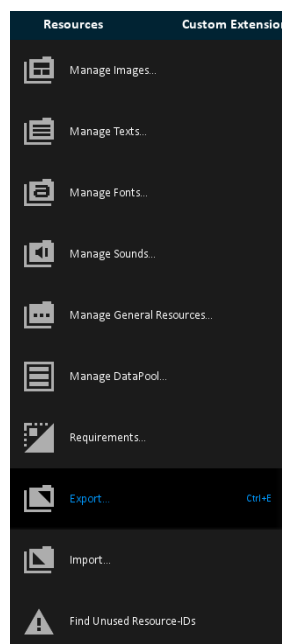
This demo has a main screen in which you can choose between several options. In each of them you can test the functionality implemented using the Guiliani API.

## 6 Edit example project on the computer

Please read chapter 5 in the document “GSE Getting started.pdf” to learn about how to edit the GuilianiDemo (see /documentation folder).

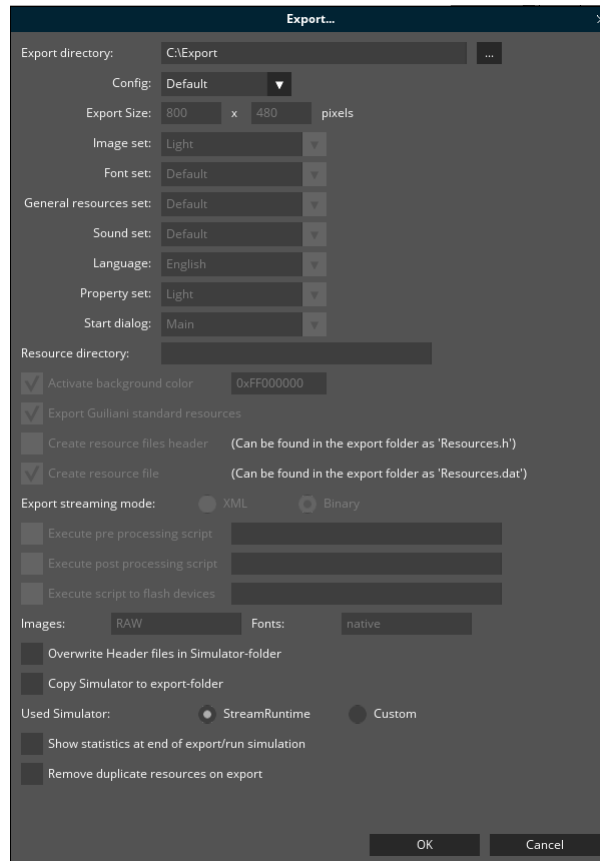
## 7 Load edited example project onto the board

- In the previous step we were able to edit a GSE project and see the modifications on a running simulation on the computer. Now we will see those changes on the board.
- Connect the board to the computer via Micro-USB-cable and to the power supply.
- With the GSE project open, export the resources (i.e. the GuilianiDemo application including all pictures etc.) by clicking Resources → Export (Fig. 5).



**Fig. 10 Export Option**

- In the Export window (Fig. 6) start-dialog and all other settings will have been selected according to the export-profile. Please select the correct export-folder by clicking on the “...”-button in the first line and navigating to the folder “<SDK-Root>/Export/RA8D1”



**Fig. 11 Exporting Configuration**

- Click on *OK*
- After the export is done, go to e2studio and start a debug session for the GuilianiDemo
- Use “Load Ancillary File” to load up the file “Resources.dat” to address 0x60000000
- Now you can start playing with the GuilianiDemo demo on the board. Navigate to “SCRATCHPAD” and you will see your first own Guiliani HMI application running on the target board.

## 8 e2Studio Workspace

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

The following four projects are available (Fig. 2):

- BSP: Pre-configured Board Support Package (BSP) for RA8D1
- SR\_GuilianiDemo: The Guiliani demo
- StreamRuntime: The StreamRuntime demo

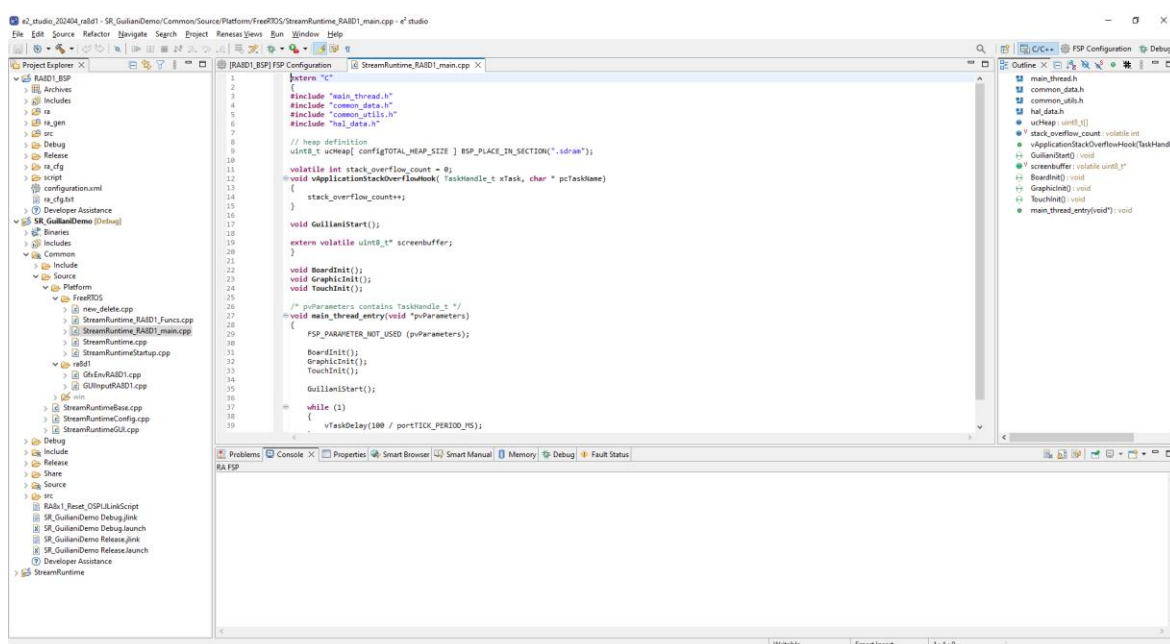


Fig. 12 e2Studio Workspace of SDK Project

### 8.1 Board Support Package (BSP)

This SDK includes a pre-configured BSP-project. 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.

**Note: To compile the BSP you first have to generate the project contents based on the used board. To do this, open the file “configuration.xml” and click “Generate Project Contents”. Now the BSP-project can be compiled.**

## 8.1.1 Directory structure

Directory	Description
src	Minimal sources for the BSP-project

Table 1 Directory Structure of BSP Project

## 8.1.2 Build configurations

- Debug: It builds BSP-project in debug mode. When the project is built, it creates a library libBSP.a in a subfolder Debug, which can be used by SR\_GuilianiDemo, StreamRuntime and BSP\_Test projects.
- Release: It builds BSP-project in release mode. When the project is built, it creates a library libBSP.a in a subfolder Release, which can be used by SR\_GuilianiDemo, StreamRuntime and BSP\_Test projects.

## 8.2 SR\_GuilianiDemo

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

### 8.2.1 Directory structure

Directory	Description
Common	Common files over different Guiliani applications
GuilianiDemo	Contains GSE projects with different resolutions
Include	Project specific includes
Source	Project specific sources

Table 2 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_FreeRTOS.[cpp h]	Target specific initialization of wrappers and configurations
[Include Source]/Platform/win/pc/StreamRuntimeStartup_FreeRTOS.h	Windows specific initialization of wrappers and configurations
[Include Source]/StreamRuntimeConfig.[h cpp]	Loads project configuration
[Include Source]/StreamRuntimeGUI.[h cpp]	Loads GUI

Table 3 Files in <SDK>\SR\_GuilianiDemo\Common Directory

File	Description
CustomExtension	Custom extensions.
GUICfg/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 4 Files in <SDK>\SR\_GuilianiDemo\Include and <SDK>\SR\_GuilianiDemo\Source Directory**

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

**Table 5 Files in <SDK>\Guiliani\Share Directory**

## 8.2.2 Build configurations

There are two configurations available for SR\_GuilianiDemo project.

1. Debug: Debug configuration. The demo application runs from internal flash. Choose this configuration to debug the application.
2. Release: Release configuration. The application runs from internal flash. Choose this configuration to test the performance.

## 8.3 StreamRuntime

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

### 8.3.1 Directory structure

Directory	Description
Common	Common files over different Guiliani applications
GuilianiDemo	Contains GSE projects with different resolutions
Include	Project specific includes
Source	Project specific sources

**Table 6 Directory Structure of <SDK>\StreamRuntime**

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

**Table 7 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 8 Files in <SDK>\StreamRuntime\Include and <SDK>\StreamRuntime\Source Directory

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

Table 9 Files in <SDK>\GSE\Share Directory

### 8.3.2 Build configurations

There are two configurations available for StreamRuntime project.

1. Debug: The demo application runs from QSPI flash. Choose this configuration to debug the application.
2. Release: The application runs from QSPI flash. Choose this configuration to test the performance.

## 9 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.



## 10 Annex

### 10.1.1 Startup sequence of Guiliani Demo application

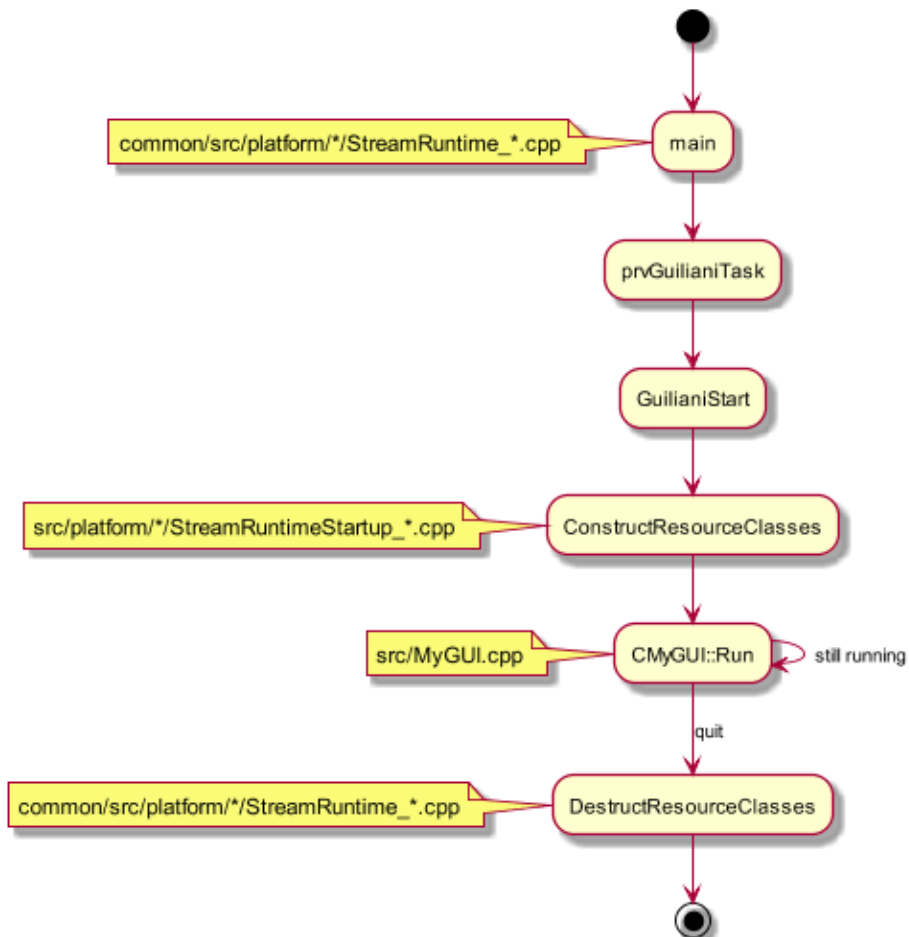


Fig. 13 Startup Sequence of Guiliani Demo Application