

RZ/A1LU group

R01AN4413EJ0100

Rev.1.00

GUI Sample Program

Jun 29, 2018

Introduction

This document describes the way to connect TES Guiliani and the users application runs on RZ/A1 Software Package.

Target Device

RZ/A1LU

Contents

1. Overview	3
2. Developing GUI Sample program.....	3
2.1 The Way to Launch GUI Sample Program	3
2.2 The Behavior of GUI Sample Program	3
2.3 Outline of Flow to Develop GUI (modify GUI Sample Program)	4
2.3.1 Invoking TES Guiliani Stream Editor	4
2.3.2 Open & Editing GUI Sample Project.....	4
2.3.3 Simulating edited GUI.....	4
2.3.4 Exporting edited GUI	4
2.3.5 Download GUI Resources	4
2.3.6 Editing User Application	4
2.3.7 Building and Running Application Program	4
3. How to connection between GUI and hardware	5
3.1 Overview of GUI Sample.....	5
3.2 How to implement Switch Function	6
3.2.1 GUI Editor	6
3.2.2 User Application.....	7
3.3 How to implement Real Time Clock Function	8
3.3.1 GUI Editor	8
3.3.2 User Application.....	8
4. Tips for developing GUI application	10
4.1 Adding new screen image (new dialog).....	10
4.1.1 GUI Editor	10
4.2 Adding new screen image transition	11
4.2.1 GUI Editor	11
4.3 Sharing the value between objects	12
4.3.1 GUI Editor	12
4.3.2 User Application.....	12
4.4 Changin images and Adding new images	13
4.4.1 GUI Editor	13

1. Overview

This document describes the usage of GUI Sample Program included in the RZ/A1 Software Package, and the way to modify this sample program. You can develop your GUI application by modifying this sample program using this document.

2. Developing GUI Sample program

The way to launch GUI Sample program and the outline of flow to develop GUI are described in this section.

2.1 The Way to Launch GUI Sample Program

By following the following sequence, you can launch GUI Sample Program.

1. Running the application program.

Run the application. For details, please refer to Quick Start Guide(R01QS0024).

2. Launching the GUI Sample program.

Input “gui[enter]” to the command console.

```
RZ/A1LU RZ/A Software Package Ver.3.01.0256  
Copyright (C) Renesas Electronics Europe.  
  
REE> gui
```

Figure 2-1 Command to launch the GUI Sample Program

2.2 The Behavior of GUI Sample Program

Figure 2-2 shows the image of the screen displayed on Stream it! board.

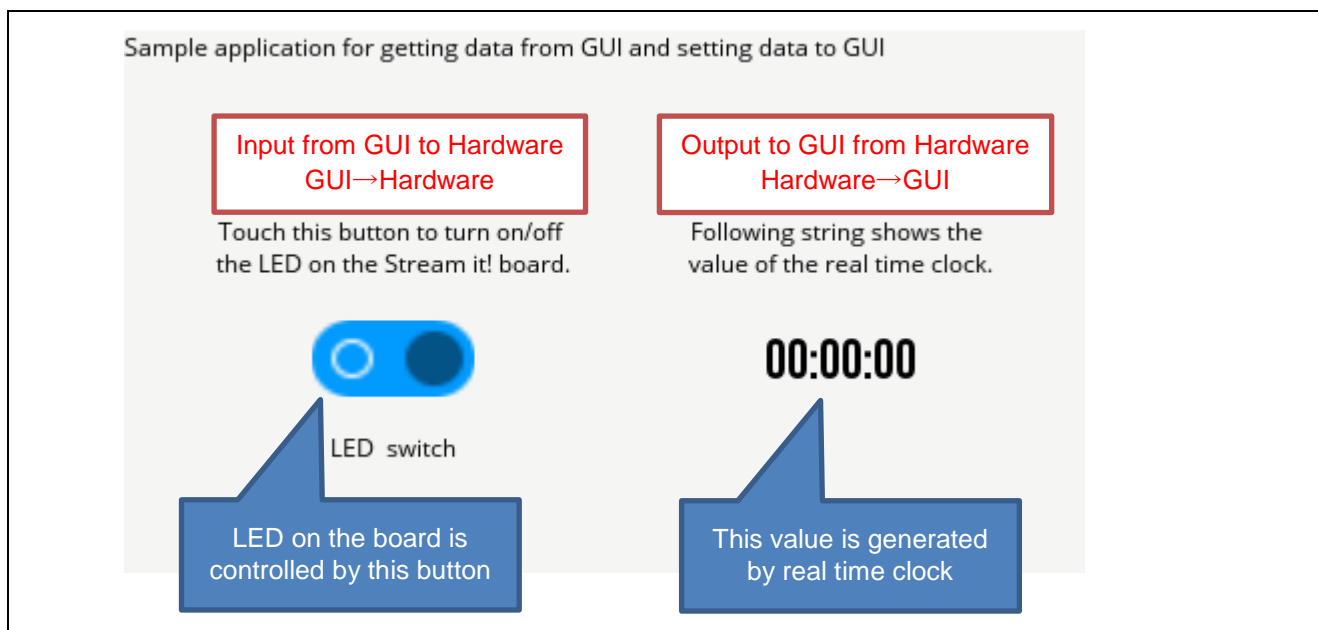


Figure 2-2 The image displayed on the board

2.3 Outline of Flow to Develop GUI (modify GUI Sample Program)

The flow to modify GUI Sample Program is shown in this section.

2.3.1 Invoking TES Guiliani Stream Editor

1. Execute "RZA1LU_Sample\src\tes\GSE\GSE.exe" on PC
GSE.exe is the WYSIWYG PC Editor.

2.3.2 Opening & Editing GUI Sample Project

1. Select "File" menu → "Open Project", and select following file:
— " RZA1LU_Sample\src\tes\GUI_Sample\GuilianiDemo\480x272\GuilianiDemo.gpr"
2. The way to edit the GUI and application is described in section 3.How to connection between GUI and hardware

2.3.3 Simulating edited GUI

You can simulate the GUI you edited on PC.

Select "File" menu → "Run Simulation", then simulation dialogue will be appeared. Press "Run" on the dialogue.

2.3.4 Exporting edited GUI

For running edited GUI on Stream it! V2 board, export GUI Sample Project.

1. Select "File" menu → "Save Project" to save the project.
2. Select "Resource" menu → "Export", then export dialogue will be appeared.
3. Select following directory and press "OK".
— "RZA1LU_Sample\src\tes\GUI_Sample\Include\GUIConfig"

2.3.5 Downloading GUI Resources

Download the GUI Resources to the board. For detail, refer Quick Start Guide(R01QS0024).

2.3.6 Editing User Application

If user system requires some handshake between user sample application and GUI, such as passing data from H/W to GUI or from GUI to H/W, user needs to modify the user sample application. For more detail how to hand some data between GUI and user sample application, please refer to section 3.

2.3.7 Building and Running Application Program

Build and run the application project. For detail, refer Quick Start Guide(R01QS0024).

3. How to connection between GUI and hardware

In this section, the way for connection between GUI and hardware is described. In GUI Sample Program, some Guiliani APIs are used. For detail of these APIs and the other APIs, refer to following URL:

https://www.guiliani.de/mediawiki/downloads/Guiliani_doc_2.2/index.html

In this section the modification needed for GUI editor is described in the subsection “GUI Editor”. And the modification needed for your own application except GUI is described in the subsection “User Application”.

Please mind that each Guiliani API has to be called from “prvGuilianiTask” context.

3.1 Overview of GUI Sample

The switch object and text field object are used in this GUI Sample. Please refer following figure to learn overview of this GUI Sample. The black arrow indicates input from GUI. In the other hand, red arrow indicates output to GUI.

DataPool and Object ID are needed to exchange data between user application and Guiliani library.

After creating an image of system on GSE, the DataPool and Object ID should be exported and be included to system. Both of Guiliani Library and Sample Application hands some data according to DataPool and Object ID.

Regarding switch input, the DataPool provides a callback to user application when status of switch object changed. When status of switch object is changed on GUI, Guiliani Library will give a callback for Sample Application.

Regarding time value output, Sample Application set current time to Guiliani Library after reading RTC register value. Sample Application calls a set function of Guiliani Library with Object ID.

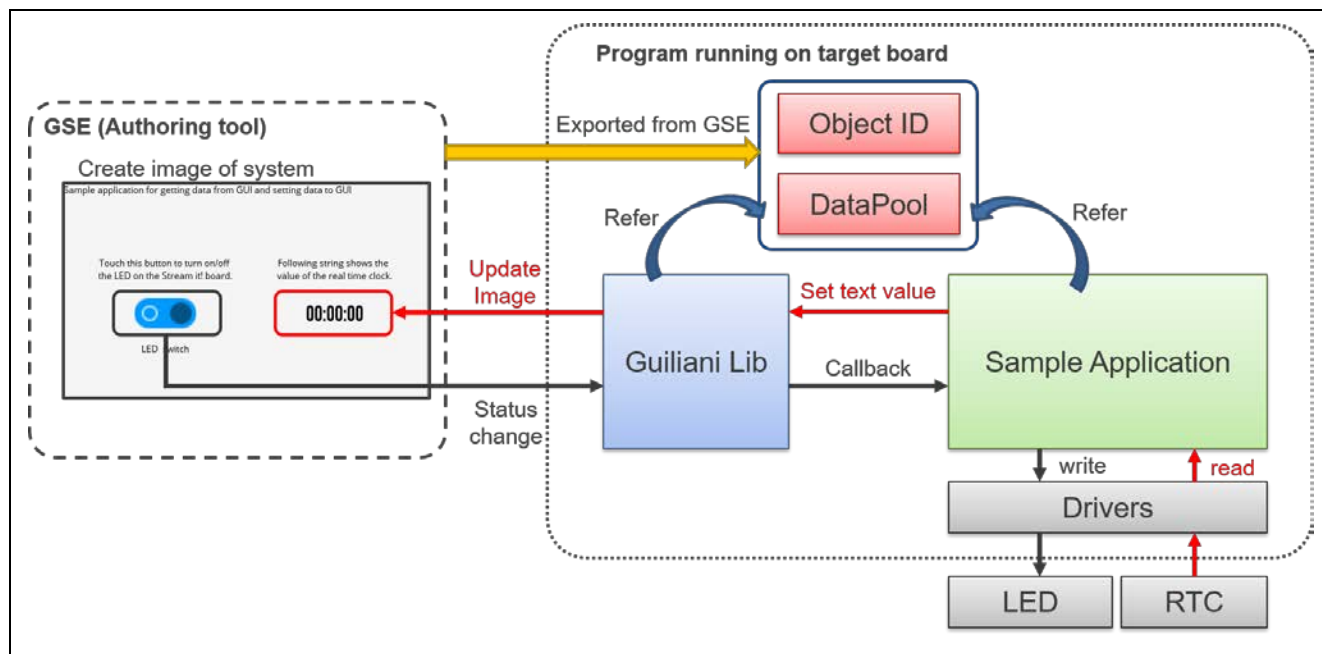


Figure 3-1 Overview of GUI Sample

3.2 How to implement Switch Function

By connecting an object for input and DataPool, when the value of the object is modified, the callback function registered by sample application will be called.

Following sequence shows the way LED button on GUI Sample Program is developed.

3.2.1 GUI Editor

1. Added object for input.
In this sample, CheckBox is added.
2. Named unique object ID.
In this sample, CheckBox is named “AID_CHECKBOX_1”.

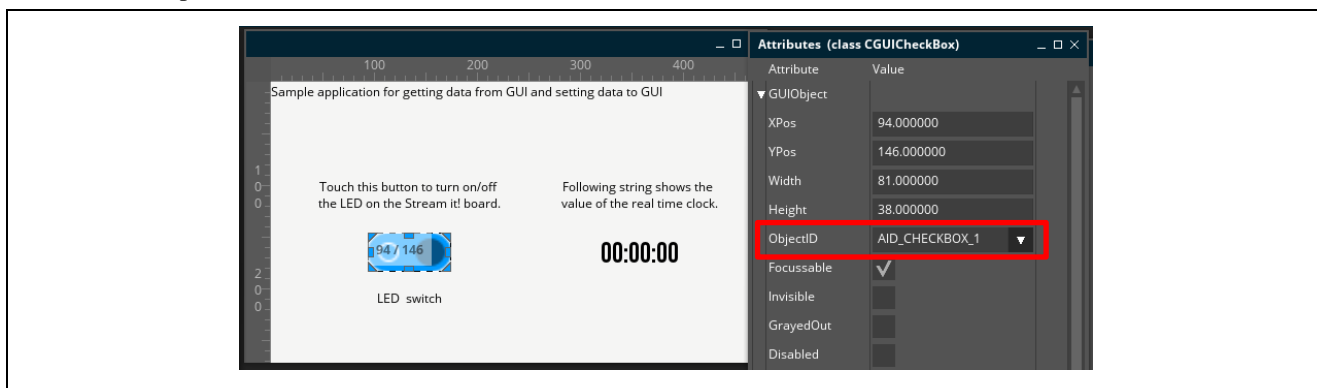


Figure 3-2 Setting ObjectID of CheckBox

3. Added DataPool by selecting “Resources” menu → “Manage” → “DataPool”. “Manage Datapool” dialogue will be appeared.
 - a. Pressed “Add new Entry”.
 - b. Named unique name of DataPool. In this sample, DataPool is named “DATAPOOL_LED”.
 - c. Pressed “▼” icon and select the object ID you named in the sequence 2. In this sample, AID_CHECKBOX_1 is chosen.
 - d. Pressed “Add as Observer”.
 - e. Pressed “Close”.

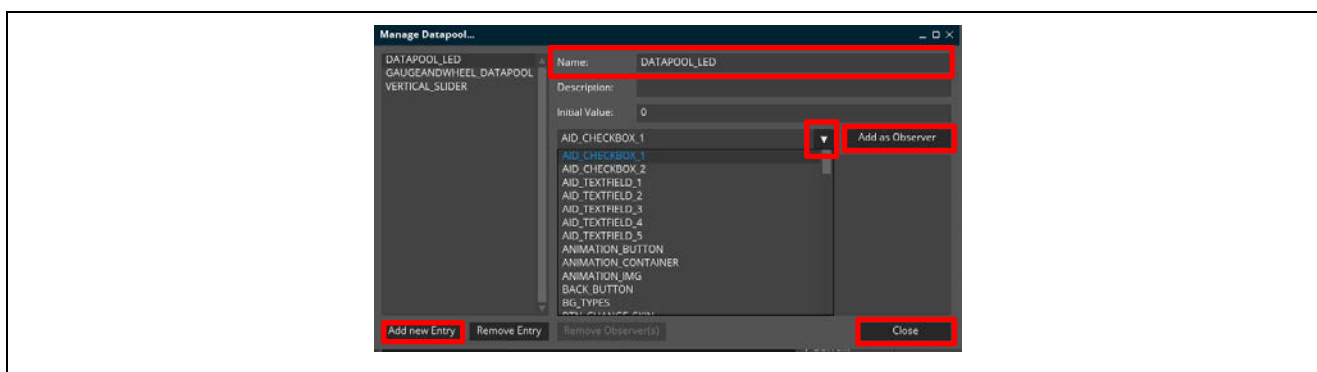


Figure 3-3 Setting DataPool parameters

Note that “GAUGEANDWHEEL_DATAPPOOL” and “VERTICAL_SLIDAR” are defined as DataPool for original TES Guiliani demonstration. this GUI Sample never use these DataPool definitions.

3.2.2 User Application

1. Register the callback function.

Please mind that callback function is registered during running Guiliani.

To register the callback function, use `CGUIDataPool::Register()` function.

Figure 3-4 shows the sample program to register the callback function to DataPool.

In this package, Figure 3-4 is described in “GUI_Sample()” function in:

"RZA1LU_Sample\src\tes\GUI_Sample\Source\MyGUI_SR.cpp"

```
/* register callback function */
CGUIDataPool::Register(DATAPPOOL_LED, &pvLedButtonCallback);
```

The ID of DataPool. callback function to register

Registering callback function

Figure 3-4 Registering callback function

2. Get the current switch value.

To get the value of DataPool, use `CGUIDataPool::Get()` function.

Figure 3-5 shows the sample program to get the data from GUI.

In this package, Figure 3-5 is described in: "RZA1LU_Sample\src\tes\GUI_Sample\Source\MyGUI_SR.cpp"

```
void CMyGUI::pvLedButtonCallback(CDataPoolEntry& data)
{
    CGUIValue value;
    uint16_t led = LED0;
    int_t led_handle = (-1);

    /* get the value of datapool for LED checkbox */
    CGUIDataPool::Get(DATAPPOOL_LED, value);

    /* open LED driver */
    led_handle = open( DEVICE_IDENTIFIYER "led", O_RDWR);

    /* check the value of datapool for LED checkbox */
    if (value.ToInt() == 0)
    {
        /* LED OFF */
        control(led_handle, CTL_SET_LED_OFF, &led);
    }
    else
    {
        /* LED ON */
        control(led_handle, CTL_SET_LED_ON, &led);
    }
    close(led_handle);
}
```

The value of DataPool and object is copied to the variable "value".

The ID of DataPool.

Checking the value of DataPool and object. value "0" means "off", and value "1" means "on".

Figure 3-5 Sample of callback function

3.3 How to implement Real Time Clock Function

Following sequence shows the way real time clock on GUI Sample Program is developed.

Note that the way to update the text and to update numeric value is different.

3.3.1 GUI Editor

1. Added object for output.

In this sample, TextField is added.

2. Named unique object ID.

In this sample, TextField is named "AID_TEXTFIELD_2".

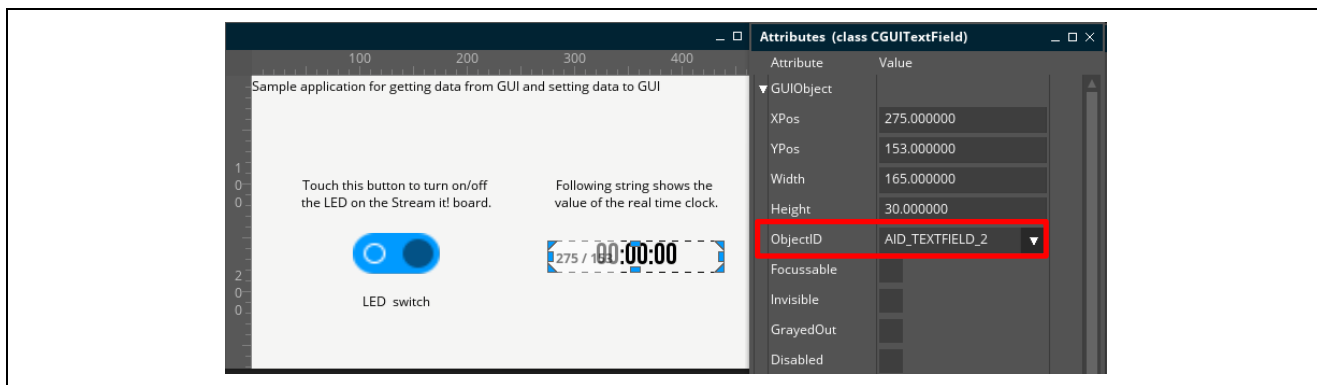


Figure 3-6 Setting ObjectID of CheckBox

3.3.2 User Application

1. Add callback for polling Real Time Clock.

"RZA1LU_Sample\src\tes\GUI_Sample\Source\MyGUI_SR.cpp"

```
// add callback for polling RTC
GETTIMER.AddAnimationCallback(100, this);
```

ms order add DoAnimate member of this
class as callback.

Figure 3-7 Sample of adding callback.

2. Update the current time as text by using SetLabel() function in the callback.

Figure 3-8 shows the sample program to update screen.

In the package, text is updated in “CMyGUI::DoAnimate” function in:

"RZA1LU_Sample\src\tes\GUI_Sample\Source\MyGUI_SR.cpp"

```
DATE last_date;

void CMyGUI::DoAnimate(const eC_Value &vTimes)
{
    /* polling real time clock */
    {
        char date_str[32];
        DATE date;
        /* open real time clock */
        int_t rtc_handle = open(DEVICE_IDENTIFIER "rtc", O_RDWR);
        if (control(rtc_handle, CTL_GET_DATE, &date) == 0)
        {
            if( date.Field.Second != last_date.Field.Second )
            {
                /* create text for time */
                sprintf(date_str,"%02d:%02d:%02d", (int_t) date.Field.Hour, (int_t) date.Field.Minute,
(int_t) date.Field.Second);
                /* get the object for AID_TEXTFIELD_2 */
                CGUITextField* pkTextField =
static_cast<CGUITextField*>(GETGUI.GetObjectByID(AID_TEXTFIELD_2));
                /* set the new label for AID_TEXTFIELD_2 */
                pkTextField->SetLabel(date_str);
            }
            last_date = date;
        }
        close(rtc_handle);
    }
}
```

Set current time value.
Guiliani will update
screen image.

Figure 3-8 Sample of updating screen.

4. Tips for developing GUI application

This section shows the way to develop GUI that is not used in the sample program. Here, some crucial functions are picked up.

For more detail, please refer to the Guiliani SDK of TES solutions.

guiliani.de : <https://www.guiliani.de/mediawiki/index.php?title=Downloads:EvalKits>

Guiliani 2.2 SDK including GSE and GuilianiDemo for Renesas RZ/A (StreamIt) with eGML (FreeRTOS10 for e² studio 6.2)

Guiliani 2.2 SDK including GSE and GuilianiDemo for Renesas RZ/A (StreamIt) with eGML (FreeRTOS10 for EWARM 8.30.1)

4.1 Adding new screen image (new dialog)

You can add new screen image into your system. The new screen image is managed as dialog in case of GSE. To add new dialogue, which means new screen, does not require any modifying the user sample application.

4.1.1 GUI Editor

1. Press “+” icon left-bottom of Dialoglist window. “Create new dialog” window will be appeared.
2. Name the unique name of the new dialogue.
3. Specify the width and the height of the dialogue.
4. Press “OK”.

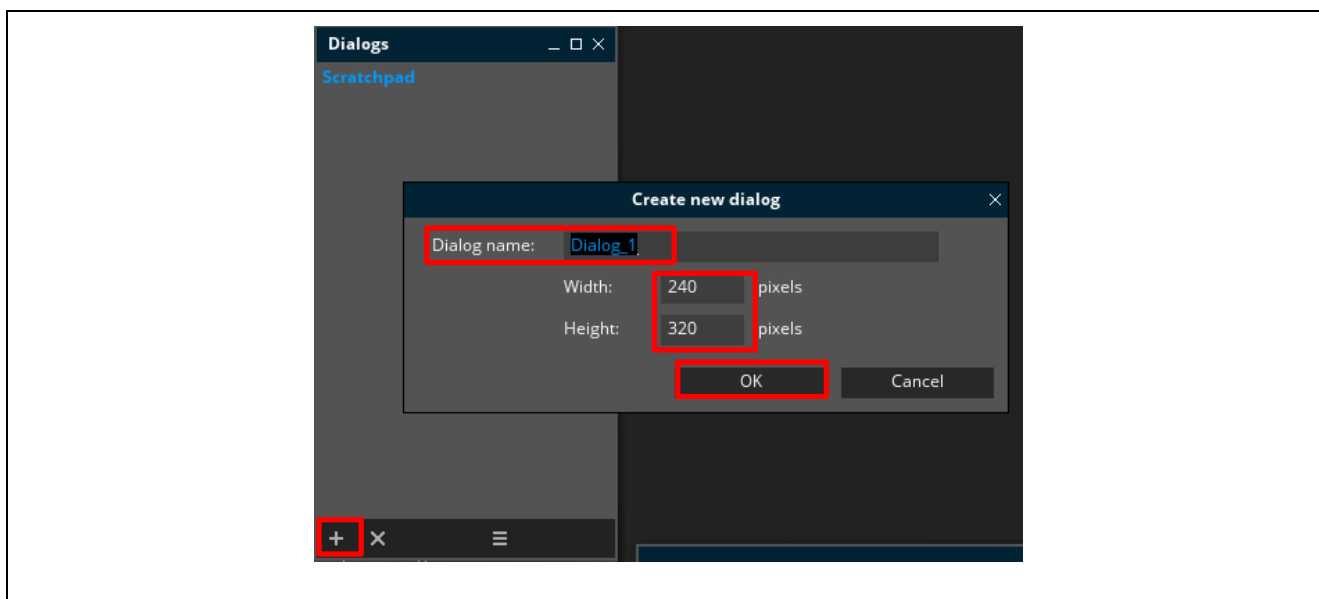


Figure 4-1 Adding new dialogue

4.2 Adding new screen image transition

After creating new dialog, you can add the screen image transition function. Also, this function does not require any modifying the user sample application.

4.2.1 GUI Editor

1. Add object such as switch, button, and so on, for moving the dialogue. (Here, describe by switch object of GUI Sample)
2. Modify the “Command Class ID” parameter to CMD_DIALOG_TRANSITION.
3. Press “▼” icon on the right of the “DestDialogFileName”. Select the dialog ID to transit.
4. Press “▼” icon on the right of the “Source Object ID”. Select the dialog ID on which the object exists.

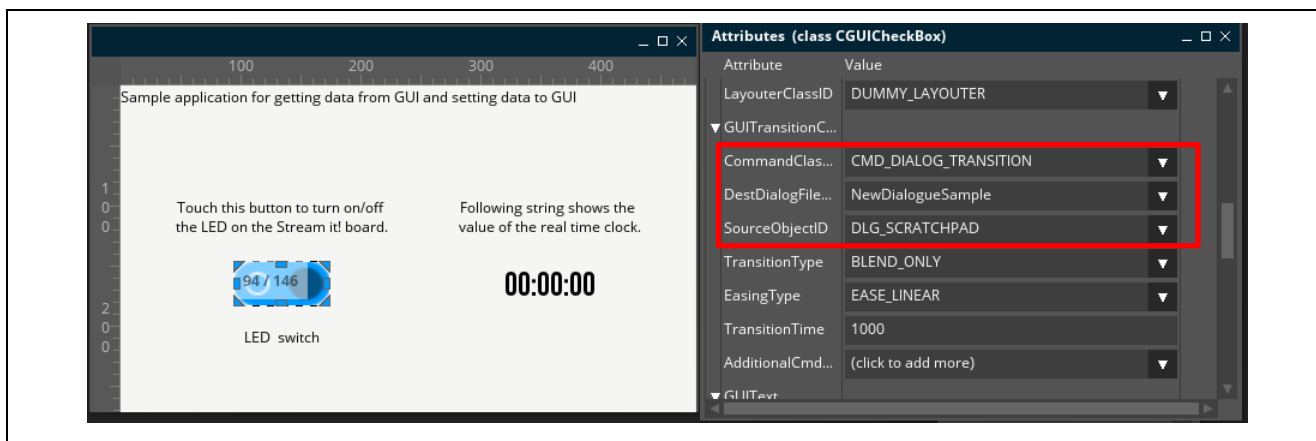


Figure 4-2 Modifying the attribute to transit dialogue.

4.3 Sharing the value between objects

You can share the value between some objects by using DataPool. For example, in case of user system require that a lamp object on GUI co-work with switch object.

4.3.1 GUI Editor

DataPool can be added by selecting “Resources” menu → “Manage” → “DataPool”. “Manage Datapool” dialogue will be appeared.

- Press “Add new Entry”.
- Name unique name of DataPool. In this sample, DataPool is named “DATAPOOL_LED”.
- Press “▼” icon and select the object ID you want to share the value. In this sample, AID_CHECKBOX_1 is chosen.
- Press “Add as Observer”.
- Repeat the sequence c. and d.
- Press “Close”.

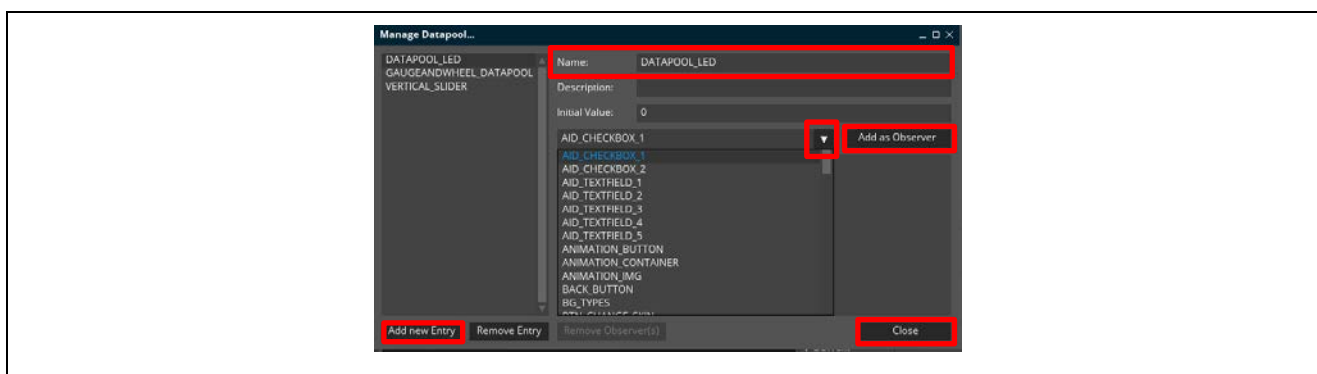


Figure 4-3 Setting DataPool parameters

4.3.2 User Application

The shared value by the DataPool can be updated by using CGUIDataPool::Set() function.

```
/* set the value of datapool for LED checkbox */
CGUIDataPool::Set(DATAPOOL_LED, (eC_Int)50);
```

The ID of DataPool.

Figure 4-4 Setting DataPool value

4.4 Chagin images and Adding new images

4.4.1 GUI Editor

You can change the image of the object by modifying the attribute that is set to “IMG_***”. By Clicking “IMG_***”, “Images” dialogue will be appeared. You can choose the image using this dialogue.

You can also add your own image by clicking “Insert new image” button.

Images bundled in the package are stored in the following directory.

- RZA1LU_Sample\src\tes\GSE\Resources

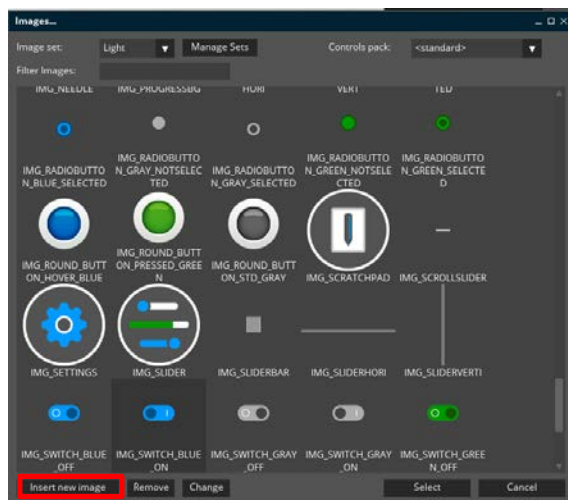


Figure 4-5 “Images” dialogue

Website and Support

Renesas Electronics Website

<http://www.renesas.com/>

Inquiries

<http://www.renesas.com/contact/>

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jun 29, 2018	all	First version issued

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.

- 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 a product with a different part number, confirm that the change will not lead to problems.

- 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.

Notice

1. 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.
2. 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.

(Rev.4.0-1 November 2017)



SALES OFFICES

Renesas Electronics Corporation

<http://www.renesas.com>

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

Renesas Electronics America Inc.

1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited

9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH

Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.

Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.

Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited

Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.

13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.

80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.

Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.

No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.

17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338