

# Renesas Microcomputer

R20AN0076EJ0101

Rev.1.01

## M3S-GUI-BUILDER: GUI Builder Introduction Guide

Jun 20, 2011

### Introduction

This document explains how to use M3S-GUI-BUILDER (GUI Builder) and describes the included sample program files.

GUI Builder is a tool that simplifies the creation of GUI applications for embedded devices. It can be used in combination with Visual C# 2010 Express and Visual C++ 2010 Express from Microsoft Corporation to create and verify the operation of the GUI design on a PC.

C language programs generated automatically by GUI Builder can also run on a variety of microcomputers (MCUs). This requires a separate M3S-GUI-LIB (GUI Library) designed for the MCU to be used. See the Renesas Electronics Web site for a list of the supported MCUs.

### Target Device

PC running Microsoft Windows XP SP3

### Contents

1. Product Configuration .....	2
2. Development Sequence .....	3
3. Installing the Development Environment .....	4
4. Using the Sample Program .....	5
5. Program Description.....	7
6. Notes .....	15

## 1. Product Configuration

GUI Builder comprises the following elements.

- M3S-GUI-BUILDER V.2.01 Release 00
- M3S-GUI-BUILDER V.2.01 Release 00 Installation Guide (this document: r20an0076ej0101\_gui.pdf)
- Product number: R0MZZZZZGL0020RXC

To install GUI Builder, run **setup.exe**. Note that it is necessary to agree to the terms of the software license agreement during the installation process.

The following files are copied during the installation process.

**Table 1.1 Product Configuration**

Installer(setup.exe)	For Windows installer. Installer will show the GUI product agreement. if user admits this agreement, installer will copy the GUI file to the path below. C:\Renesas\an_r20an0076ej_gui_v201r00
Documentation(doc)	
r20an0076ej0101_gui.pdf	Introduction Guide
r20uw0069ej0101_gui.pdf	User's Manual
Visual C# data(mgtctl)	
MGTCcontrols.dll	Custom controls
MGTProject.zip	Custom project templates
MGTForm.zip	Custom form templates
Visual C++ data(lib)	
MGT.lib, WGP16.lib, LCD.lib, SDL.lib, SDLmain.lib font_sample_win32.lib	Library for Visual C++
WGP16.dll, LCD.dll, SDL.dll	DLL for Visual C++
Visual C++ header(include)	
mgt.h, lcd.h, ximg.h font_sample_mgt.h	header files
PC tools(bin)	
guigen.exe , guiextract_img.exe	GUI program generator tool
guicov.exe	GUI test coverage measurement tool
Sample data(sample)	
MAKE.bat, etc.	Visual C# project data, Visual C++ make environment

Notes: Visual C# and Visual C++ are available for download from the Microsoft Corporation Web site.

The version of the tools and the library included in this product.

**Table 1.2 Versions**

Tools and Library	Version
Custom control (MGTCcontrols.dll)	1.1.0.0
GUI program generation tool (guigen.exe)	1.1.1
GUI program generation tool (guiextract_img.exe)	1.00
GUI test coverage measurement tool (guicov.exe)	1.1.0
Run-time library (MGT.lib)	1.1.1

## 2. Development Sequence

The GUI development sequence comprises the following steps.

- Creation of the design of the GUI elements in Visual C#
- Creation of main function and user-defined functions
- Automatic generation of C language source code from Visual C# data by using GUI program generator tool (guigen.exe)
- Building of C language source code and PC libraries by using Visual C++
- Operation verification on PC

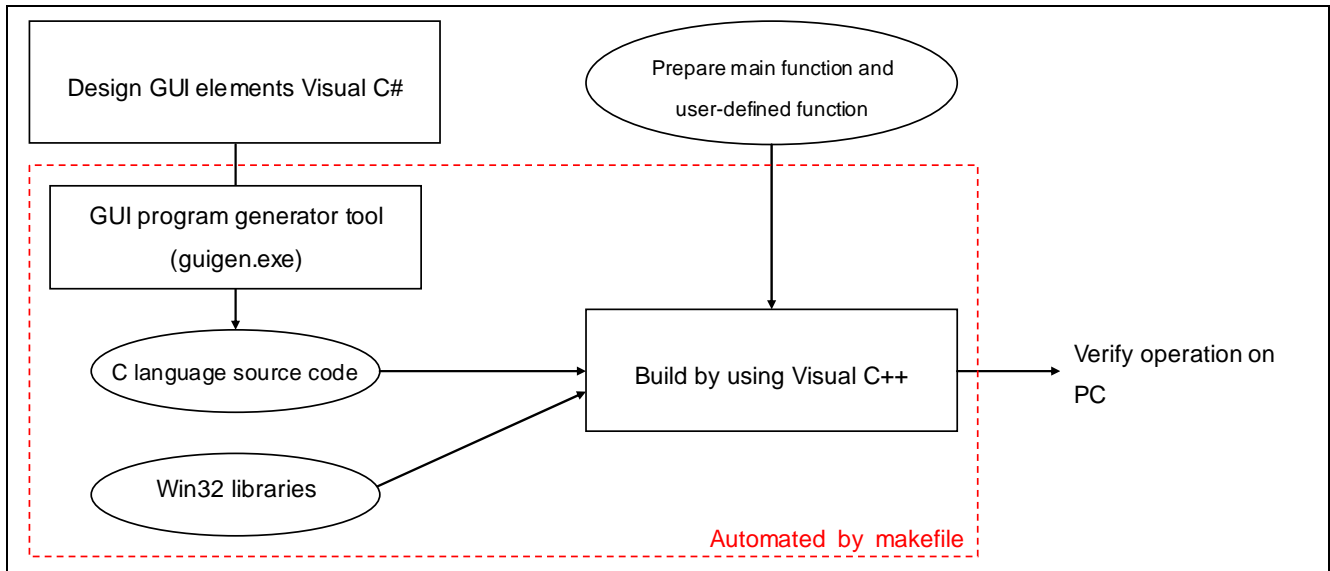


Figure 2.1 Development Sequence

### 3. Installing the Development Environment

#### 3.1 Installing Visual C# and Visual C++

Obtain Visual C# 2010 Express and Visual C++ 2010 from the Microsoft Corporation Web site and install them on the PC.

<http://www.microsoft.com/express/Downloads/>

Installation of the following optional items may be omitted.

- Microsoft Silverlight
- Microsoft SQLServer 2008 Express Service Pack 1 (x86)

#### 3.2 Installing the Runtime Component

Download **Microsoft Visual C++ 2008 SP1 Redistributable Package (x86)** from the Microsoft Corporation Web site and install it on the PC. This package installs runtime components, not Visual C++ 2008 itself.

There may be additional DLLs that need to be installed, depending on the PC setup. Confirm which DLLs are needed and install them in addition to the runtime components downloaded from the Microsoft Corporation Web site.

#### 3.3 Installing Custom Controls in Visual C#

Simply installing Visual C# is not sufficient in order to use GUI Builder. It is also necessary to register custom controls in Visual C# and to install custom project templates and custom form templates. See the GUI Builder User's Manual for installation instructions.

## 4. Using the Sample Program

A sample program is included to illustrate the basic operation of GUI Builder and provide setting examples.

### 4.1 File Structure

The **sample** directory contains the following files.

**Table 4.1 File Structure**

File Name	Description
main.c	main function: initial settings, etc.
user.c	User-defined function: program related to GUI operation
MAKE.BAT	Batch file for generating the application program source code
Makefile	Makefile referenced by Visual C++
SampleProject.zip	Compressed archive containing Visual C# data (must be extracted before use)

In addition to the **sample** directory, the following relative directory paths are referenced. Do not change the directory structure.

- bin
- include
- lib

### 4.2 Building the Program

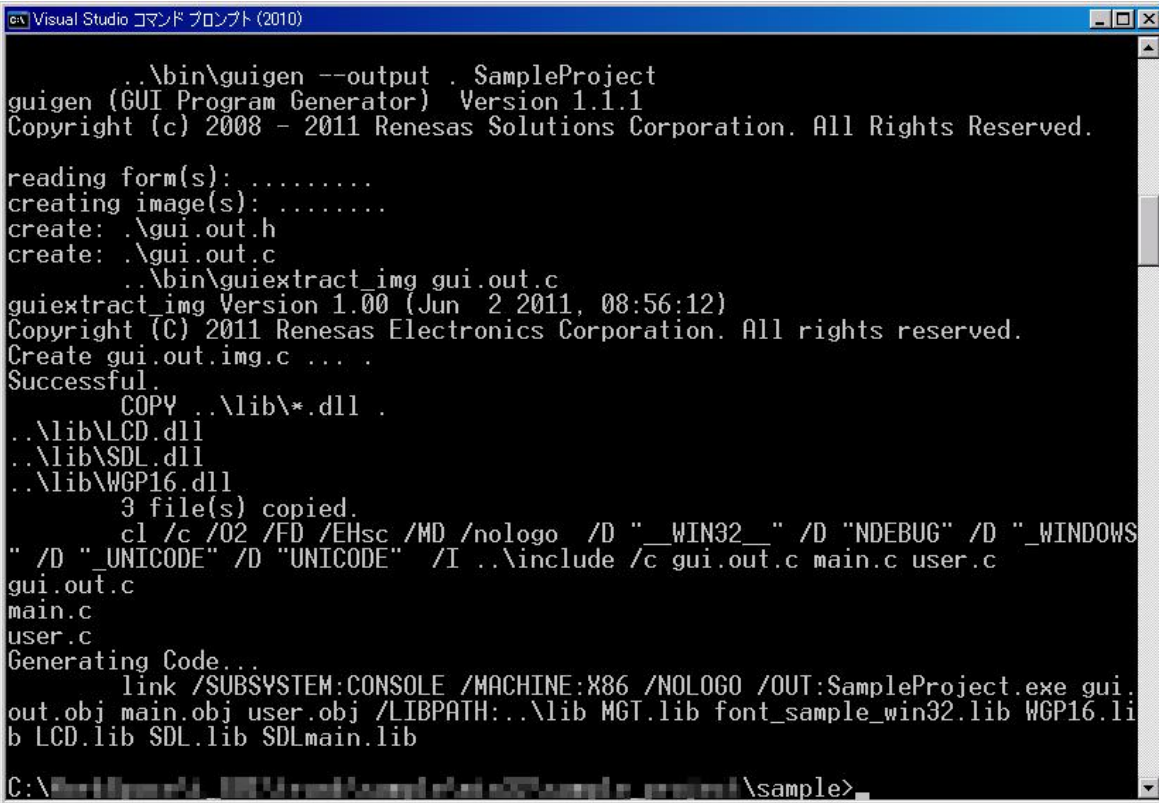
Follow the steps below to generate the application program (SampleProject.exe).

- Extract the contents of **SampleProject.zip** (located in the **sample** directory) to the **sample** directory. Depending on the software used to extract the contents of the archive, the directory structure may be altered. Make sure that a directory called **SampleProject** is created in **sample** and that it contains **SampleProject.suo** and all the other files, etc.
- Launch **Visual Studio Command Prompt (2010)** (located in Start menu > All Programs > Microsoft Visual Studio 2010 Express).
- From the command prompt, change the current directory to **sample**.
- Run **make.bat**.

An error will occur during the build process if any of the required files are located in a directory with a path containing Japanese characters.

When the build process finishes successfully, the message shown in figure 4.1 is displayed, indicating that the following files have been created (or copied).

- C language source code files (**gui.out.c**, **gui.out.h**, **gui.out.img0001**, ... **gui.out.img0008**)
- PC library files (**LCD.dll**, **SDL.dll**, and **WGP16.dll** are copied from the **lib** directory.)
- Intermediate generated files (\*.obj)
- Application program files (**SampleProject.exe** and **SampleProject.exe.manifest**)



```
ex\Visual Studio コマンド プロンプト (2010)
.. \bin\guigen --output . SampleProject
guigen (GUI Program Generator) Version 1.1.1
Copyright (c) 2008 - 2011 Renesas Solutions Corporation. All Rights Reserved.

reading form(s): .....
creating image(s): .....
create: .\gui.out.h
create: .\gui.out.c
.. \bin\guiextract_img gui.out.c
guiextract_img Version 1.00 (Jun 2 2011, 08:56:12)
Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
Create gui.out.img.c ... .
Successful.
COPY ..\lib\*.dll .
.. \lib\LCD.dll
.. \lib\SDL.dll
.. \lib\WGP16.dll
3 file(s) copied.
cl /c /O2 /FD /EHsc /MD /nologo /D "_WIN32_" /D "NDEBUG" /D "_WINDOWS"
/D "_UNICODE" /D "UNICODE" /I ..\include /c gui.out.c main.c user.c
gui.out.c
main.c
user.c
Generating Code...
link /SUBSYSTEM:CONSOLE /MACHINE:X86 /NOLOGO /OUT:SampleProject.exe gui.
out.obj main.obj user.obj /LIBPATH:..\lib MGT.lib font_sample_win32.lib WGP16.li
b LCD.lib SDL.lib SDLmain.lib
C:\>
```

Figure 4.1 Build Screen

Run the generated executable file, **SampleProject.exe**, to verify the operation of the GUI on the PC. Two windows are displayed, one showing a command prompt and one showing the GUI.

To quit the application, click the × button in the upper right corner of either window.

### 4.3 Operation

There are two ways to manipulate the GUI on the PC.

- Mouse
- Keyboard

To manipulate the GUI with the mouse, left-click on GUI controls (buttons, etc.). Pressing multiple points at the same time (multi-touch) and moving the contact point in the pressed state (dragging) are not supported.

To manipulate the GUI with the keyboard, use the left and right direction keys (arrow keys), the Tab key, and the SHFT + Tab key combination to move the focus. Then press the Enter key or the Spacebar to click the GUI control that has the focus.

To perform a click operation, it is necessary to release the mouse button (or keyboard key) over the same control over which it was pressed. The operation has no effect if the button (key) is released after moving the focus to a different location.

## 4.4 Changing the Design and Operation of the GUI

To make changes to the design of the GUI, double-click **SampleProject.sln**, located in the **SampleProject** directory created when the contents of the archive were extracted, to launch Visual C#. After making changes to the design, save the project. It is not necessary to run the build process in Visual C#.

To make changes to the operation of the GUI, edit the **user.c** file. This file contains code specifying user-defined functions, the processing that occurs when buttons are pressed, and the like. Do not edit C language source code that was generated automatically.

After making changes, run **make.bat** to update **SampleProject.exe**, etc.

If the changes to the design or program code are not built correctly, execute the command **make.bat clean** from the command prompt to delete the intermediate generated files. Then execute the **make.bat** command again.

## 5. Program Description

The configuration of the sample GUI dialog boxes (forms) is as follows.

**Table 5.1 Sample Dialog Box Configuration**

Dialog Box	Description
main	The dialog box displayed initially. The other dialog boxes are accessed from this dialog box.
MGTLabel	Dialog box related to label display and settings.
MGTButton	Dialog box related to button display and settings.
MGTIconButton	Dialog box related to icon button display and settings.
MGTCheckBox	Dialog box related to checkbox display and settings.
MGTListBox	Dialog box related to list box display and settings.
MGTPictureBox	Dialog box related to picture box display and settings.
other	Dialog box for other setting items.
resource	Dialog box for preparing image data. Not used for display.

See the sample C language source code for example settings for the various GUI controls.

## 5.1 main

This dialog box is displayed initially.

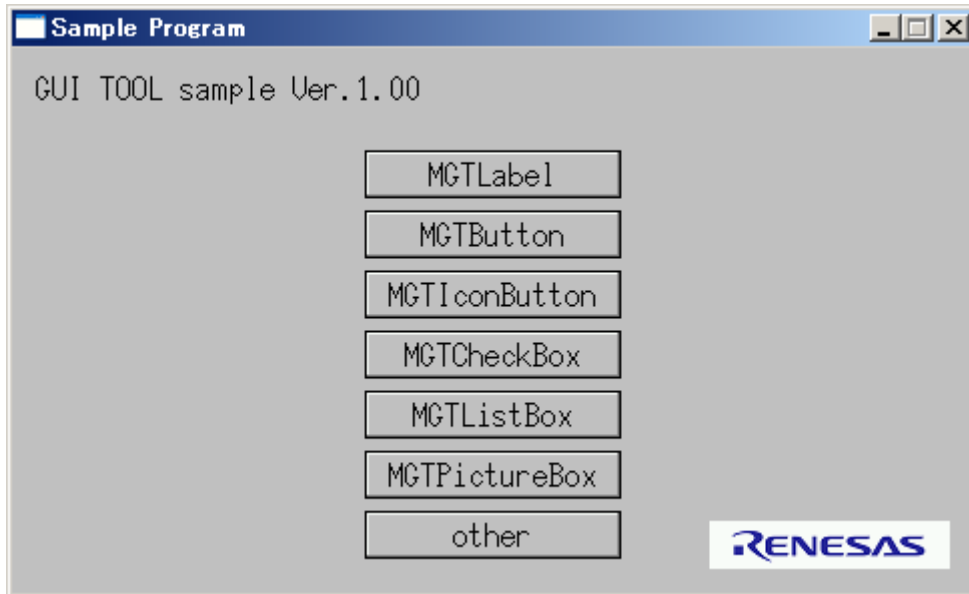


Figure 5.1 main Dialog Box

Click on one of the buttons in the center of the dialog box to access the corresponding dialog box. Changing dialog boxes uses the **OnClickGoto** property, so there is no need to supply program code to implement dialog box transitions.

The logo in the lower right corner of the dialog box is implemented by setting an image as a label, and it is displayed automatically.

## 5.2 MGTLabel

A label can display a character string or an image.

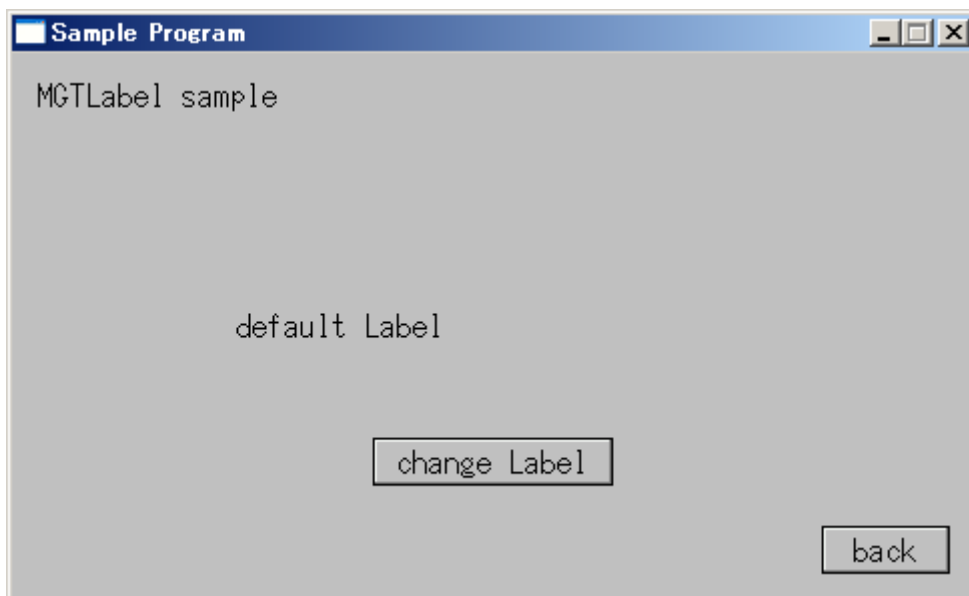


Figure 5.2 Label Sample

Click the **change Label** button in the center of the dialog box to change the character string of the label displayed near the center of the dialog box. The API function `mgt_set_label_text()` can be used to change the character string. The initial state (the original character string) is restored when you switch to a different dialog box.

It is possible to display an image instead of a character string by setting an image as the label's **Image** property. This is used to display an image on the main dialog box, but this image cannot be changed from within the sample program.

Click the **back** button in the lower right corner of the dialog box to return to the main dialog box. The **back** button on each of the dialog boxes described below functions the same way, so no additional explanation is provided.

#### Additional Information

- It is not possible to change the font, text color, or image from within the sample program.
- The character string may not contain newline characters.
- The character string is displayed in the upper left of the label area by default, and any portion that exceeds the available area is not displayed.

### 5.3 MGTButton

You can press the buttons and specify a character string or image to be displayed on them.

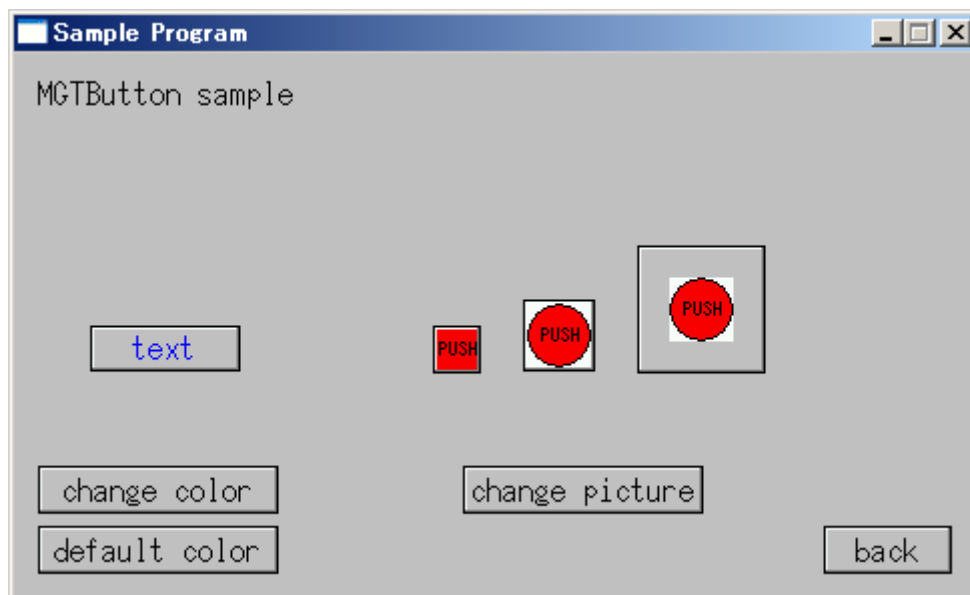


Figure 5.3 Button Sample

Click the **change color** button in the lower left corner of the dialog box to change the color setting of the button. The API function `mgt_set_rdb()` is used to set the color. After the setting is made, the new color takes effect when the dialog box is redrawn. The buttons on all the dialog boxes use the same setting, and it is not possible to make different settings for individual buttons. The setting made by the API function `mgt_set_rdb()` is retained until GUI Builder is initialized or a new setting is entered. Clicking the **default color** button restores the original setting.

Click the **change picture** button in the lower center of the dialog box to change the image specified for the button. The API function `mgt_set_button_image()` is used to make the setting. The initial state (the original image) is restored when you switch to a different dialog box.

The text or image is displayed in the center of the button. If the size of the specified image differs from the available display area, the peripheral portions are cut off or a blank border is added. The display area is four pixels smaller than the size of the button (because two pixels at the top, bottom, left, and right edges are taken up by the button design). When the focus is applied to the image and button, the focus color may not be visible if there is no blank border.

#### Additional Information

- It is not possible to change the button character string from within the sample program.

### 5.4 MGTIconButton

Unlike an ordinary button, an icon button displays only the specified image. In addition, any image may be specified for the “pressed state” and “focused state.”

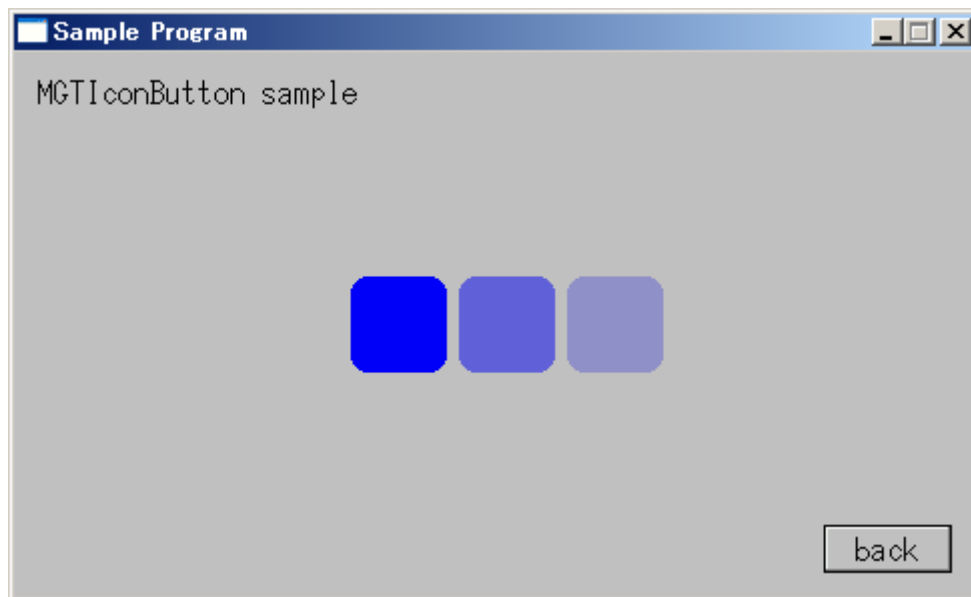





Figure 5.4 Icon Button Sample

The **Image**, **Image1**, and **Image2** properties specify, respectively, the image in the ordinary state, the image in the focused state, and the image in the pressed state. In like manner, the **Alpha**, **Alpha1**, and **Alpha2** properties specify the alpha value (transparency) in each of the above states.

The configuration of the sample GUI IconButton is as follows.

Table 5.2 Sample IconButton Configuration

property	Meaning	Setting of sample(		
		Left Icon,	Center Icon,	Right Icon
Image	the image in the ordinary state			
			ICON1.bmp	
Image2	the image in the focused state			
			ICON2.bmp	
Image3	the image in the pressed state			
			ICON3.bmp	
Alpha	the alpha value in the ordinary state	255 (nontransparent),	128,	64
Alpha2	the alpha value in the focused state	255 (nontransparent),	128,	64
Alpha3	the alpha value in the pressed state	255 (nontransparent),	128,	64

None of the setting items for this dialog box can be entered dynamically. The images and alpha values set in the design cannot be changed from within the sample program.

In addition, it is not possible to specify transparent colors, so it is necessary to prepare icon button images in which the peripheral portion matches the background color or background image around the button in order to create icon buttons that are not square in shape. For example, if an image of a round button surrounded by solid black is used, the solid black portion is displayed as well. It is therefore necessary to modify the image beforehand by painting over the black portion of the image with a color, etc., that matches the area around the button.

**Additional Information**

- The **Image3** and **Alpha3** properties are not used.

## 5.5 MGTCheckBox

A square portion indicating the checked/unchecked state and a character string are displayed.

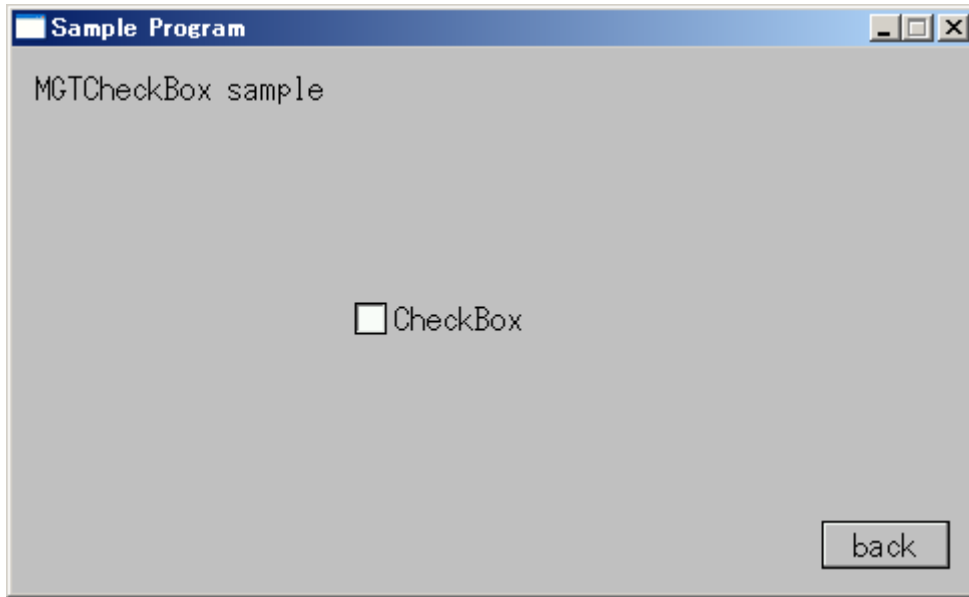


Figure 5.5 Checkbox Sample

The checkbox state is read by the API function **mgt\_get\_checkbox\_value()** and set by the API function **mgt\_set\_checkbox\_value()**.

The checked/unchecked state is retained only while the dialog box containing the control is displayed. The initial state (unchecked) is restored when you switch to a different dialog box.

In the sample program, the state is stored in RAM when the MGTCheckBox dialog box is closed, and the checkbox state is restored when you return to the MGTCheckBox dialog box.

Clicking anywhere within the checkbox display area, even outside the square checkbox, toggles the checked/unchecked state. An × mark is displayed inside the square checkbox when in the checked state.

### Additional Information

- It is not possible to change the checkbox character string.
- It is not possible to change the character or color used to display the × mark when in the checked state.

## 5.6 MGTListBox

Multiple characters strings are displayed in the list box area.

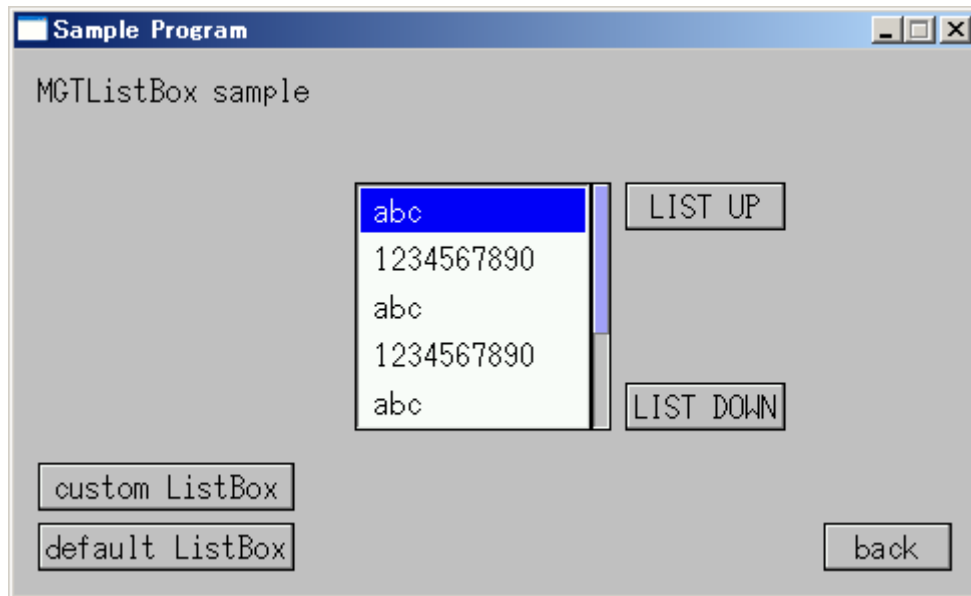


Figure 5.6 Listbox Sample

The list information (character strings displayed, selection position, display start position) is managed by the user and is specified by using the API function `mgt_set_listbox_data()`.

The list box cannot be manipulated by clicking directly within the list box. In addition, though a vertical scroll bar may be displayed automatically next to the list box depending on the number of display items in the list, it cannot be manipulated by clicking it.

Portions of character strings that extend beyond the display range are not displayed, and no horizontal scroll bar is displayed.

The list display position can be changed from within the program by clicking the **LIST UP** and **LIST DOWN** buttons at the right of the list. The **MGT\_HOOK\_ONPUSH** event is used to trigger processing so that the buttons respond when pushed rather than when clicked. It is also acceptable to have the **MGT\_HOOK\_ONCLICK** event trigger processing.

Click the **custom ListBox** button in the lower left corner of the dialog box to change the color of the list box selection line or the line spacing. The API function `mgt_set_rdb()` is used to make this setting. After the setting is made, the new color takes effect when the dialog box is redrawn. The list boxes on all the dialog boxes use the same setting, and it is not possible to make different settings for individual list boxes. The setting is retained until GUI Builder is initialized or a new setting is entered. Clicking the **default ListBox** button restores the original setting.

### Additional Information

- In the initial state only the frame is displayed.

## 5.7 MGTPictureBox

A picture box is an area in which a user-defined image can be displayed.

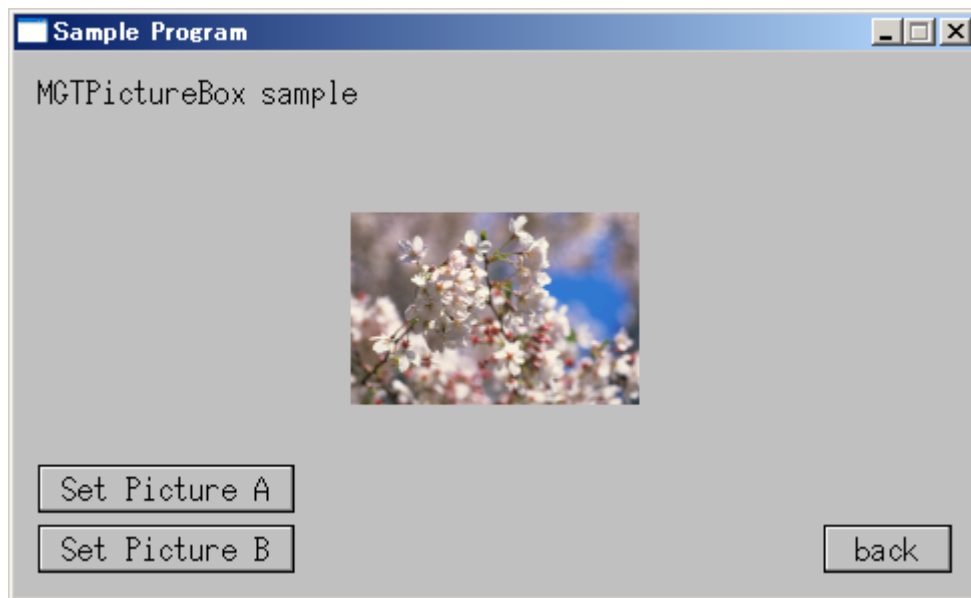


Figure 5.7 Picture Box Sample

The image to be displayed is specified by using the API function `mgt_set_picturebox_image()`. In contrast to a label, simply setting the image in the **Image** property does not cause the image to be displayed.

Click the **Set Picture A** or **Set Picture B** buttons in the lower left corner of the dialog box to paste either of two different images into the picture box. The initial state (no image) is restored when you switch to a different dialog box.

### Additional Information

- In the initial state nothing is displayed.
- The upper left corner of the display area is used as the origin point of the display. Portions that extend beyond the edges of the display area are not displayed.

## 5.8 other

This dialog box provides examples using other API functions.

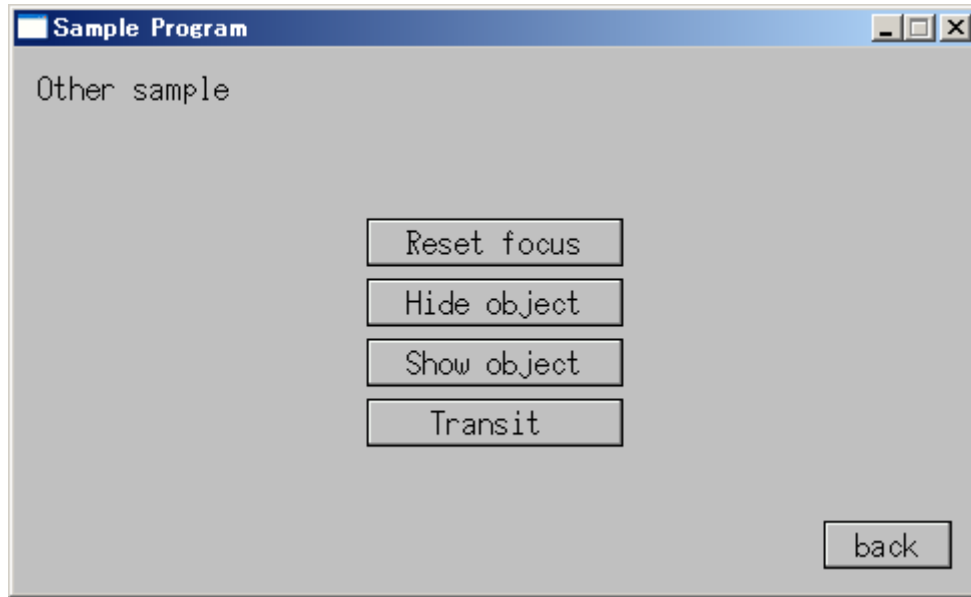


Figure 5.8 Other Samples

Clicking the **Reset focus** button cancels the focused state. The API function `mgt_reset_focus()` is used to cancel the focused state.

Clicking the **Hide object** button causes the **Hide object** button to disappear. Click the **Show object** button to restore it. It is also possible to hide controls other than buttons. The API functions used are `mgt_hide_control()` and `mgt_show_control()`.

Clicking the **Transit** button causes **SET** to be displayed to the right of the button and the main dialog box to reappear after about three seconds have elapsed. When the button's **OnClickGoto** property is used, the transition between dialog boxes occurs automatically when the **Transit** button is clicked. This allows the user to perform a transition conditionally. The API function `mgt_transit()` is used.

## 5.9 resource

In this dialog box the images to be used are registered. It is not used for display.

Since the purpose is to obtain image data, controls other than buttons may be used to specify images.

## 6. Notes

### 6.1 Creating a New Project in Visual C#

To create a new project, select MGTPProject.



Figure 6.1 Project Created by Visual C#

When the project is first saved, the save dialog box includes a **Create solution directory** checkbox. Make sure this is not checked to ensure that the included sample files and directory structure remain properly matched. Allowing Visual C# to create a solution directory will add an additional level to the directory structure, making it necessary to modify the paths of the Makefile input files.

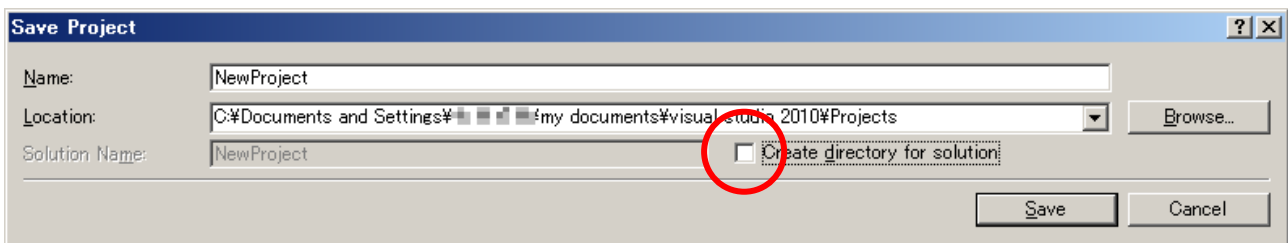


Figure 6.2 Point to Remember when Saving Project

## 6.2 Debugging the Application

The batch file **make.bat**, included in the sample, uses **guigen.exe** to perform conversion to C language source code and handles all steps through building with Visual C++. It is not possible to debug the application in this state.

To build the application in the Visual C++ integrated development environment (IDE), select **Win32 console application** and then check **Empty project** from the additional options in the Win32 application window.

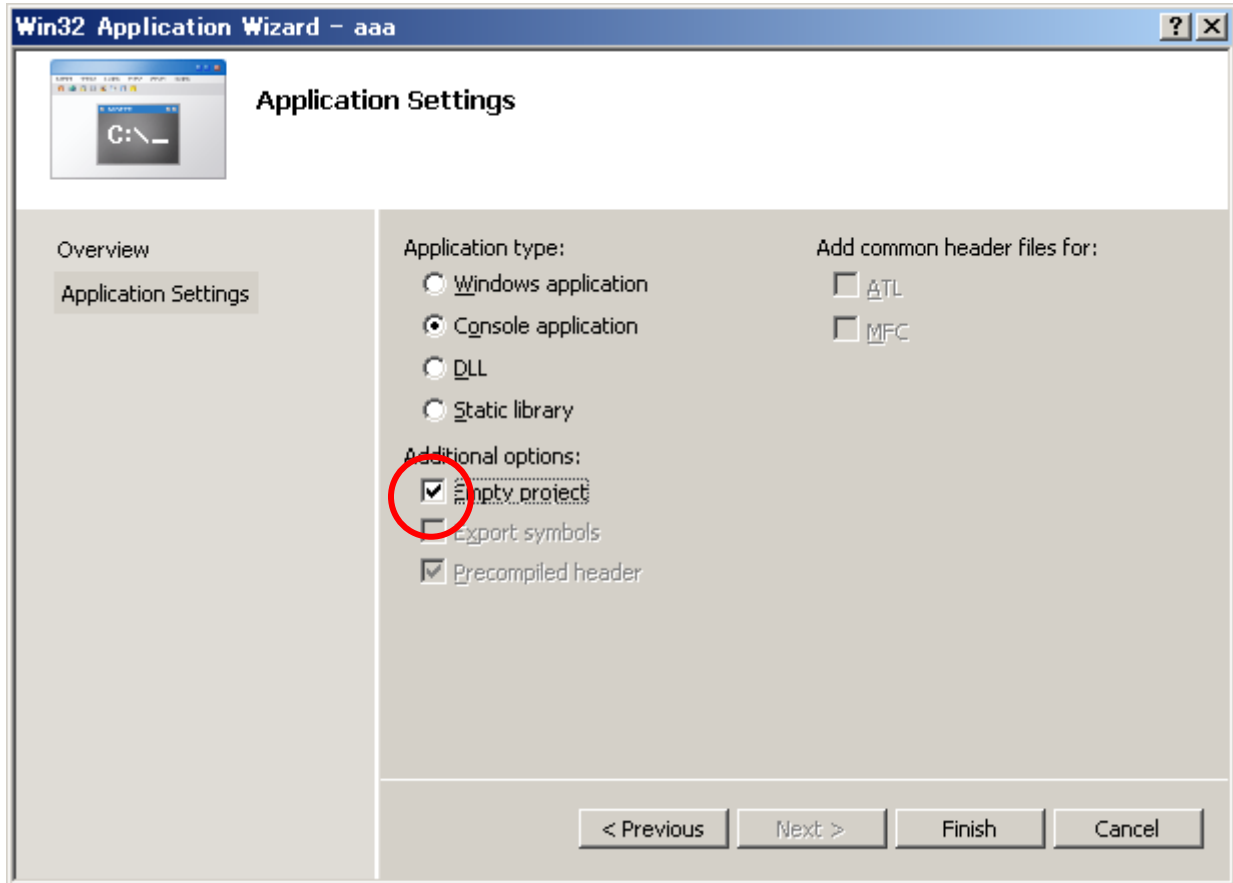


Figure 6.3 Creating a Project in Visual C++

Next, register the necessary C language source files and libraries.

Attempting to build the application with the default settings causes the following error message to be displayed.

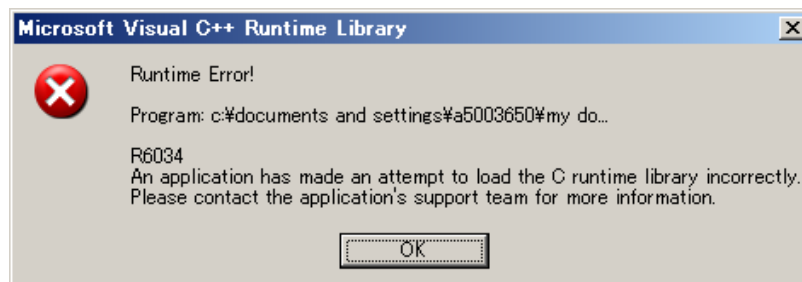


Figure 6.4 Error Message

To eliminate this error, add `__WIN32__` to the **Preprocessor definition** on the project properties page. The other settings may be left at their default values.

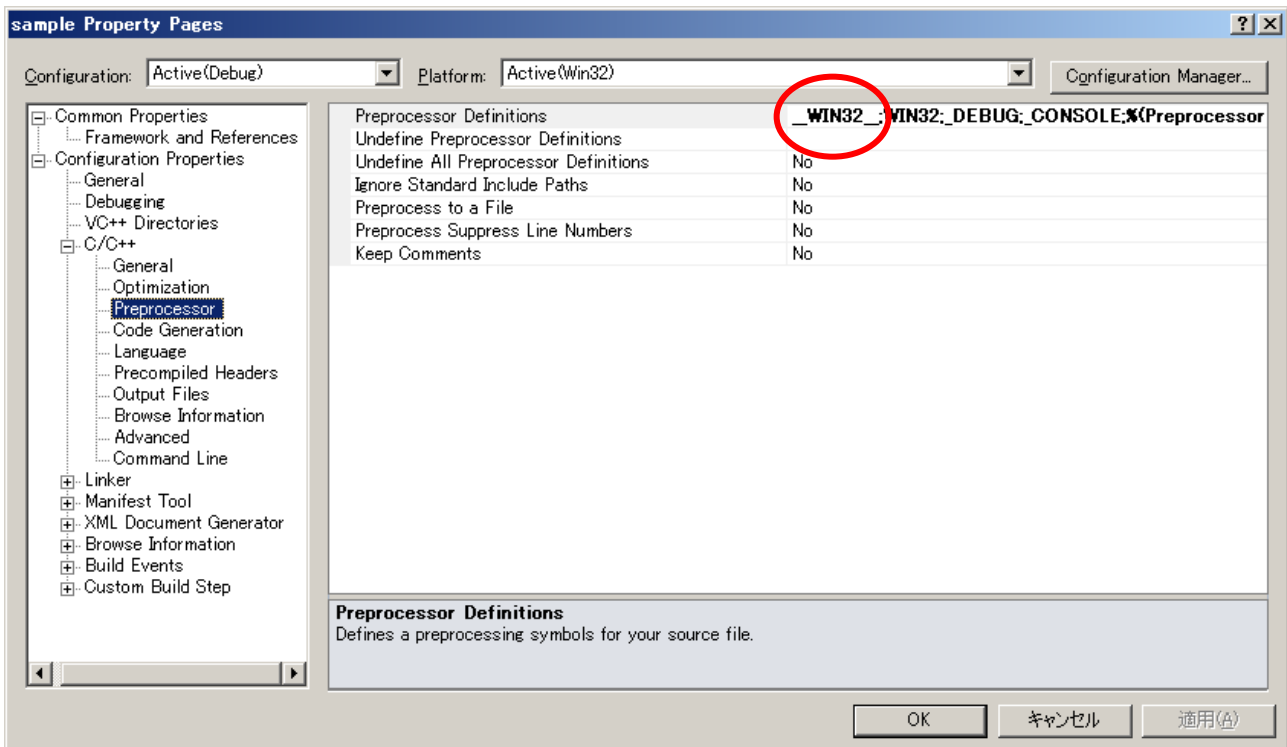


Figure 6.5 Project Property Settings

Copy the DLL files to the same directory to which the compiled application was output, then run the application.

For information on using Visual C++, settings, etc., see the help file or Web sites related to Visual C++.

Website and Support

Renesas Electronics Website

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

Inquiries

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

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

## Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Apr.20.11	—	First edition issued
1.01	Jun.20.11	—	Release with GUI Builder Ver.2.01

## General Precautions in the Handling of MPU/MCU Products

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

### 1. Handling of Unused Pins

Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

- The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

### 2. Processing at Power-on

The state of the product is undefined at the moment when power is supplied.

- The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

### 3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

- The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

### 4. Clock Signals

After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

- When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

### 5. Differences between Products

Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.

- The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

## 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.



### 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.

2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.  
Tel: +1-408-588-6000, Fax: +1-408-588-6130

#### Renesas Electronics Canada Limited

1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada  
Tel: +1-905-898-5441, Fax: +1-905-898-3220

#### Renesas Electronics Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K  
Tel: +44-1628-585-100, Fax: +44-1628-585-900

#### Renesas Electronics Europe GmbH

Arcadiastrasse 10, 40472 Düsseldorf, Germany  
Tel: +49-211-65030, Fax: +49-211-6503-1327

#### Renesas Electronics (China) Co., Ltd.

7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China  
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

#### Renesas Electronics (Shanghai) Co., Ltd.

Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China  
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

#### Renesas Electronics Hong Kong Limited

Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong  
Tel: +852-2886-9318; Fax: +852 2886-9022/9044

#### Renesas Electronics Taiwan Co., Ltd.

7F, No. 363 Fu Shing North Road Taipei, Taiwan  
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

#### Renesas Electronics Singapore Pte. Ltd.

1 HarbourFront Avenue, #06-10, Keppel Bay Tower, Singapore 098632  
Tel: +65-6213-0200, Fax: +65-6278-8001

#### Renesas Electronics Malaysia Sdn.Bhd.

Unit 906, 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 Korea Co., Ltd.

11F., Samik Laviel' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea  
Tel: +82-2-558-3737, Fax: +82-2-558-5141