RENESAS

Renesas IC Toolbox Software

This manual is intended to help users understand the key concepts of the Renesas IC Toolbox (RICBox) software.

Contents

1. Introduction			
	1.1	System Requirements	.4
2.	Insta	llation and Setup	.4
	2.1	RICBox Software Installation	.4
	2.2	Device Software Installation	.4
3.	Load	ing and Creating Configurations	.8
	3.1	Creating a New Configuration	.8
	3.2	Loading a Settings File	.9
4.	Side	Panel Buttons	10
	4.1	Control Panel View	11
	4.2	Wizard Setup	11
	4.3	Configuration View	12
	4.4	Register View	13
	4.5	Block Diagram View	13
5.	Devi	ce Connection	14
	5.1	Scanning for Connected Devices	15
	5.2	Connecting to the Emulator	16
6.	Error	s and Warnings	19
7.	Com	mand Line Interface (CLI)	19
8.	Statu	is Monitor	22
9.	Logg	ling	23
10.	Abou	ıt	24
11.	Powe	er Estimation View	25
12.	Clou	d Updating	26
13.	Regi	ster Write Export	27
14.	Multi	ple Configuration Support	28
15.	Revi	sion History	31

Figures

Figure 1. RICBox Installation Files	.4
Figure 2. Help > Updates Window	.5
Figure 3. RICBox Updates Page	.5

Figure 4. Available Plugins Window	6
Figure 5. Plugin Software Information Window	6
Figure 6. Selectable Versions Window	6
Figure 7. Plugin Install Process	7
Figure 8. Create New Project Window	8
Figure 9. Select Product Plugin	9
Figure 10. Open Existing Project Window	9
Figure 11. Select RBS File	10
Figure 12. Side Panel Buttons	10
Figure 13. Control Panel Overview Page	11
Figure 14. Wizard Page Navigation Window	12
Figure 15. Configuration View Tabs	12
Figure 16. Configuration View Search	13
Figure 17. Register Access Window	13
Figure 18. Block Diagram Pop-up Page	14
Figure 19. Not Connected Button	14
Figure 20. Connect Button	14
Figure 21. Connected Button Illuminated	14
Figure 22. Program Button	15
Figure 23. Not Connected Button	15
Figure 24. Configure Button	15
Figure 25. Refresh Connection	16
Figure 26. Connect Button in Connection Settings	16
Figure 27. Not Connected Button	16
Figure 28. Configure Button	17
Figure 29. Manual Connection Settings	17
Figure 30. Emulator Setup	18
Figure 31. Emulator Connect	18
Figure 32. Error and Warnings View	19
Figure 33. Tools Menu > CLI	19
Figure 34. CLI Help	20
Figure 35. CLI Connect Command Help	21
Figure 36. Tolls Menu > Status Monitor	22
Figure 37. Status Monitor Window	22
Figure 38. Help Menu > Log	23
Figure 39. Trace Button	23
Figure 40. Log Buttons	23
Figure 41. Help Menu > About	24
Figure 42. Current Device and Software Versions Window	24
Figure 43. Power Estimation View Button	25
Figure 44. Power Estimation View Window	25
Figure 45. Installed Software	26
Figure 46. Software Update Banner	26
Figure 47. Download 'rbpkg' Plugin	26
Figure 48. Manual Plugin Install	27
Figure 49. Register Write Export Button	27

Figure 50. Register Write Export Pop-up Page	
Figure 51. Export Register Writes	
Figure 52. Configuration Menu	
Figure 53. Configuration Selection	
Figure 54. Configuration Settings	29
Figure 55. Configurations Pop-up Menu	29
Figure 56. Current Active Configuration	29
Figure 57. Configurations Settings Overview	
Figure 58. Configurations in the CLI	

1. Introduction

The RICBox, or Renesas IC Toolbox, is a versatile Windows utility designed to streamline and enhance the development process when working with Renesas products. This tool is particularly useful for engineers and developers as they can configure and program ICs according to their requirements on various evaluation boards.

1.1 System Requirements

RICBox requires Windows 10. The setup program installs **.NET 5.0** alongside RICBox as a self-contained deployment, therefore, **.NET 5.0** does not have to be installed separately.

RICBox ships with an installable version of Python. The setup program will automatically install Python if an appropriate version of Python is not already installed or detected on the computer.

- USB 2.0 or USB 3.0 Interface
- Processor: Minimum 1GHz
- Memory: Minimum 512MB; recommended 1GB
- Available disk space: minimum 600MB (1.5GB 64-bit); recommended 1GB (2GB 64-bit)

2. Installation and Setup

There are two parts to installing and running the RICBox software. The first being the Renesas IC Toolbox and the second being the related device software plugin. Each device has a separate plugin interface available that will install in parallel to the RICBox software. This plugin can be installed through the RICBox update feature after the base software is installed.

2.1 **RICBox Software Installation**

- Download the RICBox software directly from the Renesas website.
 - This can normally be found on the webpage of the supported devices.
- The downloaded package will have two install files. Run the installer that pertains to the version of windows on the current system. For example:
 - The file labeled RICBox-version-x86.exe needs to be installed on a windows 32-bit system.
 - The file labeled RICBox-version-x64.exe needs to be installed on a windows 64-bit system.

System type Pen and Too	: 64-bit Operating System, x64-based pro ach: No Pen or Touch Input is available for th	ocessor his Display		
enesas-ic-too	lbox-ricbox-installer-v3.3.0(1).zip > Renesas IC Toolbo	x Installer (v3.3.0) 🗸	õ	, p s
^	Name	Туре		
	RICBox-3.3.0.75868-x64.exe	Application		
*	RICBox-3.3.0.75868-x86.exe	Application		

Figure 1. RICBox Installation Files

- Follow the on-screen prompts to finish the installation.
 - If any issues occur during the installation process, try to uninstall and reinstall the software. If the problem persists, then please contact Renesas support.

2.2 Device Software Installation

The RICBox software must be installed before installing a specific device software.

1. Open RICBox and click on *Help > Updates* to open the RICBox *Updates* window.



Figure 2. Help > Updates Window

2. This will display the *RICBox Updates* pop-up page.



Figure 3. RICBox Updates Page

3. The available plugins will be displayed on the left side of the window.

RENESAS		
Search Installe	ed Aller Aller Aller	
Search		Q
RICBox-Design	er	
RICBox		
R_VersaClock7		
R_PCle		
R_FemtoClock	2	
R_FemtoClock	3	
R_VersaClock7	Auto	
R_ProXo2		
N_FIGX02		

Figure 4. Available Plugins Window

4. Click on one of the installation packages to view the software information.

RICBox-Designer RICBox	R_VersaClock7	,	5.0.0-dev117617	Install Download
R_VersaClock7	R_VersaClock7	R_VersaClock7		Download
R_PCle	The R_VersaClock7 pac	kage.		
R_FemtoClock2	Id	R_VersaClock7		
R_FemtoClock3	Version	5.0.0-dev117617		
R_VersaClock7Auto	Installed Version	5.0.0-dev117617		
R ProXo2	Architecture	хб4		
I I I I I I I I I I I I I I I I I I I	Date Published	2022-08-12		
	Tags	timing, versaclock7		
	Dependencies	RICBox ~7		

Figure 5. Plugin Software Information Window

5. Use the drop-down menu by the install button to select what version is to be installed.

D. Varan Clask	•	5.0.0-dev114945	• 🛨 Install
R_VersaClock7 G R_VersaClock7		5.0.0-dev114945	Download
		5.0.0-dev114883	
The R_VersaClock7 pac	kage.	5.0.0-dev114037	
Id	R_VersaClock7	5.0.0-dev113151	
Version	5.0.0-dev114945	5.0.0-dev113137	
Installed Version	5.0.0-dev113151	5.6.6 0 0 0 0 0 0 0 0	
Architecture	хб4	5.0.0-dev113100	
Date Published	2022-07-28	5.0.0-dev113012	
Tags	timing, versaclock7	5.0.0-dev112912	
Dependencies	RICBox ~7		

Figure 6. Selectable Versions Window

6. Click the install button to add that package to the install queue (more than one can be added).

7. Click the Install Now button in the upper-right corner of the window to initiate the installation process.

RENESAS		S RUCBox Updates			7		×
Search Installed				2 packages to i	nstall on exit	Insta	I Now
Search	٩	Prerelease			Manual I	nstall	-
RICBox-Designer RICBox R_VersaClock7 R_PCle		R_FemtoClock R_FemtoClock2 The R_FemtoClock2 pa	2 ckage.	5.0.0-dev114962	• 🛃	Install Downloa	sd
R_FemtoClock2		Id	R_FemtoClock2	/			
R_FemtoClock3 R_VersaClock7Auto		Version Installed Version Architecture	5.0.0-dev114962 Not installed				

Figure 7. Plugin Install Process

8. Once the installation process is complete, RICBox will restart and the new updates should now be accessible.

3. Loading and Creating Configurations

Settings files are used to save and distribute custom device configurations. They have the file extension '.rbs'. Each settings file contains all of the register settings for a given device, metadata for the RICBox to understand, and other information pertaining to that specific configuration. They can be opened like a text file but should not be edited from any other location than the RICBox software.

Note: Changing values in a settings file from a text editor will corrupt the file and it will not be usable with the RICBox software.

3.1 Creating a New Configuration

1. To create a new configuration, open the RICBox software and click on the *Create New* project button.

Renesas	RI
File Help	
RENESAS BIG IDEAS FOR EVERY SPACE	
Create new project	
Open project	

Figure 8. Create New Project Window

2. Then in the "Select Product Family" section, select the device family that is being used. From the "Select Product" section, select the working device. Click the *OK* button to open the new configuration.

Select a Product Family	Select a Product Variant
FemtoClock2	RC32504A
VersaClock7	RC32514A
	8P49N344 Specific Product
	RC22504A
	RC22514A
	RC32508A
ocked – the 'point-ot-use'. The combination of small size (4 C22504: Clock Generation C22514: Clock Generation with internal crystal C32504: Jitter Attenuation	x4mm), low power and excellent noise performance sets this family in a class by itself.
C32514: Jitter Attenuation with internal crystal	
	Select OK when done
	Select OK when done

Figure 9. Select Product Plugin

3.2 Loading a Settings File

Loading a settings file is similar to creating a new one. To load an existing settings file, click on the *Browse* button just after opening the RICBox software. This will take the user to a file browser.

Note: Recently used settings files are under the "Recent Files" section.



Figure 10. Open Existing Project Window

Navigate to the directory that stores the settings file and select it. RICBox settings files have the file type '.rbs'.

→ Y T 🚺 > This	PC → Desk	top > Rixbox_Settings_Files		5 V	Search Rixbo	ox_Settings	_Files	P
rganize 👻 New folder						== •		2
This PC	^	Name	Date modified	Туре	Size			
3D Objects		RC32504A_test_Config.rbs	4/7/2021 10:37 AM	RBS File]			
📃 Desktop								
Documents								
Downloads								
J Music								
Pictures								
Videos								

Figure 11. Select RBS File

4. Side Panel Buttons

The side panel consists of five separate buttons. Each button opens to a separate page. Each page has a unique view, enabling the configuration of the device from different perspectives.



Figure 12. Side Panel Buttons

- Control Panel displays the device overview
- Wizard displays the wizard view for initial device configuring
- Configuration displays the register settings in a readable text format with a search engine
- Registers shows a graphic of the registers in the device
- Block Diagram displays a configurable block diagram view

4.1 Control Panel View

The control panel view displays an overview page depicting the major settings for the device. This page can be used as an important reference for the overall device configuration. Each device will display different values in the overview section depending on what is the most important information pertaining to the configuration.

AL
50MHz
10MHz
10MHz
~227.2727MHz
156.25MHz [LVDS]
156.25MHz [LVDS]
100MHz [CMOS, Qx/nQx Opposite Phase]
100MHz [HCSL]
10GHz
100
~363.0624kHz
59.84 degrees
~11.0524MHz
yes
JAMODE (jitter attenuator mode)
10GHZ
1000 - 22 9202Hz (4 7190% from goal of 25Hz)
~23.6205HZ (-4.7 169% from goal of 25HZ)
~1 5542kHz (-37 8301% from goal of 2 5kHz)
~0 1804/dR
0.1034B

Figure 13. Control Panel Overview Page

4.2 Wizard Setup

When creating new configuration, the wizard page is the first screen to appear. The intention of the wizard is to allow the configuration of the main parts of a given device without having to navigate through the entire GUI to do so. It does this through a step process where navigation is performed through a set of pages; each page pertaining to a different section of the device that is necessary for overall configuring.

Configuring RC32504A		1 of 3 Inputs •
Operation Mode Synthesizer APLL Reference Selects XIN / R Crystal Frequency SOMHz Load Capacitance (pF) 8 Coverdrive CLKIN Clock Mode Frequency None Clock Mode Frequency None Clock Mode Clock Mode Frequency None Clock Mode Clock Mode Clo	Current Page Open specific wizard page	 Operation Mode: Synthesizer - The device is configured to use a single clock source as the input to the analog PLI (APLI). The APLI then uses that cock source to generate the outputs. The reference source is selected by using the APLI Reference mux and can come from either XIN or CLKIN. Jitter Attenuator - The device is configured to use the reference for the XIN pin as the input to the APLI and the clock source at the CLKIN pin as reference for the digtal PLI (DPLI). The DPLI settings will become available when this mode is selected. DFO - The device is setup as a digitally controlled oscillator, whereby the output frequency can be adjusted via a frequency control word (FCW). This turns off DPLL functionality and use dock sources similar to Synthesizer mode. Crystal - Configure the settings for the clock reference at the XIN pin. Set the signal type through the clock mode menu and adjust the frequency between TMHz and 800MHz.
	Wizard navigation buttons	
Cancel		Next Finish

Figure 14. Wizard Page Navigation Window

4.3 Configuration View

Configuration view enables easy navigation through register settings via the tabs at the top of the page. Each section has all of the critical registers and data fields listed to allow configuration of the device block.

File	Tools	Help									
<u> </u>	-0	Search	OSC	Inputs	APLL	DPLL	TDC APLL	Outputs	GPIO		
۲ ۲	e.	Searc	-								
//		Keywo	ord								
Ē.	_Э ,	act_mon_accept_ppm_clkin0									
œ		act_mo	n_accep	ot_ppm_c	lkin1						
- 40 40	₩ }	act_mon_enable_clkin0									
Z	_ ରା	act_mon_enable_clkin1									
\vdash	ি	act_mo	n_rejec	t_ppm_cl	kin0						
		act_mo	n_rejec	t_ppm_cll	kin1						

Figure 15. Configuration View Tabs

The search tab provides access to the configurable fields through a search bar. The search engine can be used to find any specific configurable field.

Search OSC Inputs APLL DPLL TDC APLL Outputs GPIO									
Search Keyword apl actual_apll_frequency apll_goal_frequency									
apll_pfd_input_frequency									
apll_ref_mux_frequency									
TOP.APLL.ANA_SPARE_CNFG.ana_spare_out									
TOP.APLL.ANA_SPARE_STS.ana_spare_sts									
TOP.APLL.APLL_CNFG.apll_dis									
TOP.APLL.APLL_CNFG.apll_ref_sel									
TOP.APLL.APLL_CNFG.cp_dis									

Figure 16. Configuration View Search

4.4 Register View

The register view shows a graphical diagram of the registers and allows the user to read or write any of the individual registers. By clicking on the individual register block, the given registers for that block will appear to the right. They can either be adjusted by writing directly to the diagram or entering values into the data fields on the right.

	00	01	02	03	04	05	06	07	Info		Operations
D0h	1F	0	0	0	0	0	0	0	Offset	103h	Write Read
D8h	0	0	0	0	1	0	0	0			
E0h	4A	1E	1	81	22	0	5C	8F	Bitsets		
E8h	0	0	4	0	0	0	0	0	0x103[7:6]	TOP.OUT[0].ODRV_MODE_CNFG.out_cmos_mode	OUTx, nOUTx a 🗡 🖬
F0h	В	1	0	0	44	0	0	0	0x103[5:4]	TOP.OUT[0].ODRV MODE CNFG.out lvds cm voltage	900mV ×
F8h	D	4D	*	0	0	0	0	0	0v103(3)	TOP OUTIOLODBY, MODE (NEG out host term en	
100h	69	0	В	6C	B4	3	0	0	0x105[5]		
108h	69	0	В	6C	B4	3	0	0	0x103[2]	TOP.OUT[0].ODRV_MODE_CNFG.en_out_bias	
110h	69	0	В	6C	B4	3	0	0	0x103[1:0]	TOP.OUT[0].ODRV_MODE_CNFG.out_mode	HCSL ×
118h	69	0	В	6C	B4	3	0	0			
120h	0	0	70	0	0	0	70	0			
128h	0	0	0	0	0	0	0	0			
130h	10	2F	0	2	0	2	0	0			
138h	0	0	0	0	0	0	0	0			
140h	21	6	44	9	5	0	0	0			
148h	0	0	0	0	0	0	0	0			
150h	0	0	0	0	64	0	0	5			



4.5 Block Diagram View

The block diagram view is intended to give a visual representation to the internal register settings. The diagram has the same block representation as in each datasheet. To navigate around the block diagram, click on any of the blocks and open the pop-up page. This leads to a more in-depth look as to what registers are available to configure each block.



Figure 18. Block Diagram Pop-up Page

5. Device Connection

Device connection is handled through the button on the lower right-hand corner of the screen. Before connecting to a device, ensure that the device is properly powered. If using a Renesas evaluation board, the RICBox software can automatically detect the connection protocol type. To directly connect to a device:

1. Click on the *Not Connected* button in the lower right-hand corner.





2. Click the *Connect* button right next to the program button (this may take a few seconds to establish a proper connection).



Figure 20. Connect Button

3. Once fully connected to a device, the connection button will be illuminated green.

Errors	Warnings	RC32504A	Connected

Figure 21. Connected Button Illuminated

4. Once a device is fully connected, click on the *Program* button to write all register values from the configuration to the device. After the program button has been clicked the first time, the device will follow all configuration updates without having to re-write the registers. For example, if a user programs a configuration and then changes the output frequency afterwards, the device will reflect the new output frequency.



Figure 22. Program Button

5.1 Scanning for Connected Devices

To use the device scanning tool:

1. Click on the *Not Connected* button in the lower right-hand corner of the screen.



Figure 23. Not Connected Button

2. Click the *Configure* button next to the read button.



Figure 24. Configure Button

3. Under the "Auto" tab, click the *Refresh* button.



Figure 25. Refresh Connection

4. A list of connected devices should be displayed. Select the desired connected device, then click on Connect.

Connection Settings	×
Auto Manual	
📀 ftdi (0) I2C 0x09	0
🕑 ftdi (0) SPI 🗡	0
Refresh	Connect

Figure 26. Connect Button in Connection Settings

5.2 Connecting to the Emulator

The RICBox software comes equipped with an emulator to simulate the connection to a real device. To establish a connection to the emulator:

1. Click the Not Connected button in the lower right-hand corner of the screen.



Figure 27. Not Connected Button

2. Click the *Configure* button next to the Read button.



Figure 28. Configure Button

3. Click on the "Manual" tab.

RENESAS	RICBox Connection	—		×
Auto Manual				
Adapter	Emulator			•
Protocol	I2C			-
Address (7-bit)			0x0	00 🍣
Clock Rate (Hz)			10000	00 🗢
			Conn	ect

Figure 29. Manual Connection Settings

4. Change the Adapter settings to "Emulator".



Figure 30. Emulator Setup

5. Select the desired connection protocol and click *Connect*.

RENESAS	RICBox Connection		×
Auto Manual			
Adapter Protocol Address (7-bit) Clock Rate (Hz)	Emulator 12C	0x1 10000	• 00 🔄 00 🗢

Figure 31. Emulator Connect

6. Errors and Warnings

Error and warning messages indicate that one or more of the register settings are not properly set. An error/warning message will be displayed on the tool bar at the bottom of the screen. To view the current errors, click on the Red highlighted errors button. If there is none, then no errors are occurring. The warnings will be displayed with a yellow button indicator. All Errors should be handled before programming a device.



Figure 32. Error and Warnings View

7. Command Line Interface (CLI)

The CLI is a power tool for configuring and testing devices. It gives users quick access to everything the GUI has to offer, but in a terminal command line.

1. Start by clicking on the "Tools" drop-down menu off the main menu. Click on CLI.



Figure 33. Tools Menu > CLI

2. When the CLI window opens, type "help" for a list of all available commands



Figure 34. CLI Help

Every command has documentation on how to use it. Choose a command and type "-h" after it to view the available options.

RENESAS	RICBox CLI	—	×
> connect -h			
usage: connect [-h] [-c	CHANNEL] [-i INDEX] [-p PORT NUMBER] [-a ADDRESS]		
[offse	t-size OFFSET_SIZE] [spi-mode {0,1,2,3}]		
[spi-b	it-order {msbfirst,lsbfirst}] [spi-3wire]		
[baud	BAUD] [ssc-interface {i2c,owi}]		
[rsens	e-interface {i2c,owi}] [device DEVICE]		
[{ftdi,a	ardvark,linuxi2c,linuxspi,ssc,rsense,emulator}]		
[{custom	,i2c,spi,owi,ssc,smbuslike,rs232,rsense}]		
Connect to a device so	read/write commands can be issued.		
positional arguments:			
{ftdi,aardvark,linuxi	2c,linuxspi,ssc,rsense,emulator}		
	Type of the adapter providing connectivity to the		
	device.		
{custom,i2c,spi,owi,s	sc,smbuslike,rs232,rsense}		
	Serial protocol for the connection (e.g., i2c, spi,		
	ssc, rsense)		
optional arguments:			
-h,help	show this help message and exit		
-c CHANNEL,channel	CHANNEL		
	FTDI channel / serial number (e.g., A or B for		
i TNDEV index TND	F 12232HQ)		
-I INDEX,INDEX IND	EA ETDI index (e.g. 0 on 1 for ET233200)		
-n PORT NUMBERnon	-number PART NUMBER		
-p Fort_Nonbert,por	Aardvark port number (e.g., 0 or 1) for T2C/SPT. COM		
	port number for SSC and RSense (e.g. 2 for COM2).		
-a ADDRESS,address	ADDRESS		
	I2C: 7-bit hex address of slave. Not used for other		
	protocols.		
offset-size OFFSET_	SIZE		
	offset size, in bytes		
spi-mode {0,1,2,3}	SPI mode.		
spi-bit-order {msbf	irst,lsbfirst}		
ani Zuina	SPI bit order. Detault is MSB first.		
spi-swire	Use SPI in three Wire mode.		
ssc-interface (i)c	badd fate in bps for K3232 and 330 connections.		
	Intra-interface for SSC connections		
rsense-interface {i			
	Intra-interface for RSense connections		
device DEVICE	The device id of a (sub)device within a composite		
	device.		
A			
1			
connect -h			

Figure 35. CLI Connect Command Help

8. Status Monitor

The RICBox software has a feature to monitor all status bits of any device. To open this up, click the "Tools" menu button at the top of the screen and click on *Status Monitor*. Once open, click the *Start* button to start polling the registers in the device.



Figure 36. Tolls Menu > Status Monitor

New registers can be added to the list by typing in the register name in the field in the upper right corner of the screen and clicking *Add*.

RENESAS	RICBox Status Monitor	– – ×
Start Status monitor is not running.	Add a custom field	Add
GLOBAL	GPIO 🔶	LOSMON
TOP.GLOBAL.DEVICE_STS.startup_seq_sts TOP.GLOBAL.DEVICE_STS.osc_fallback TOP.GLOBAL.DEVICE_STS.device_ready TOP.GLOBAL.DEVICE_STS.config_loaded	TOP.GPIO.STARTUP_STS.gpio_at_startup TOP.GPIO.GPIO_STS.lock_o TOP.GPIO.GPIO_STS.oe_i Add more regis to the list	TOP.LOSMON[0].LOSMON_STS.los_sts TOP.LOSMON[1].LOSMON_STS.los_sts TOP.LOSMON[0].LOSMON_EVENT.los_evt TOP.LOSMON[1].LOSMON_EVENT.los_evt sters
DPLL	BIAS	APLL
TOP.DPLL.DPLL_STS.dpll_state_sts TOP.DPLL.DPLL_STS.dpll_lock_sts TOP.DPLL.DPLL_LOL_CNT.dpll_lol_cnt	TOP.BIAS.BIAS_STS.bias_cal_fail TOP.BIAS.BIAS_STS.bias_cal_done	TOP.APLL.APLL_STS.apll_lock_sts TOP.APLL.VCO_CAL_STS.vco_cal_fail TOP.APLL.VCO_CAL_STS.vco_cal_done TOP.APLL.VCO_CAL_STS.vco_cap

Figure 37. Status Monitor Window

Note: Click *Stop* when finished with monitoring as the register polling may interfere with other feature of the software.

Note: Currently the status monitor only shows the register value and not what the value means. Users have to search up the register name in the configuration view and read the tooltip for more information.

9. Logging

The logging page shows communication transactions as well as debugging information. This a great tool for understanding the communication needed to properly program a device.

1. Go to the "Help" menu at the top of the page and click on Log.



Figure 38. Help Menu > Log

- 2. Deselect the Debug check box to show the registers written to the device in real-time. When a device is connected, clicking on the program button or changing register values will show up in the log.
- 3. Check the *Trace* check box to show the errors that are handled by RICBox.

Figure 39. Trace Button

The buttons in the lower right corner of the page are as follows:

- The file icon button takes the user to the directory that stores the log file.
- The "X" icon button clears the contents of the log.
- The arrow icon button takes the user to the bottom of the log view.

Debug 🗸 Trace

Figure 40. Log Buttons

E X

10. About

The "About" page depicts important information about the current running software.

To see the "About" page, go to the main menu at the top of any page, go to the "Help" tab, then click About.



Figure 41. Help Menu > About

This shows the versions of all currently installed individual device software and RICBox. This is handy when updating the current software. It is important to always use the most up to date software version as new features are frequently added.





Figure 42. Current Device and Software Versions Window

11. Power Estimation View

Device plugins that support power estimation view will have an additional side-panel button with a lightning bolt icon.



Figure 43. Power Estimation View Button

Click this button to open a window displaying the estimated power consumption of the device given the current configuration settings. Results are shown in terms of minimum, typical, and maximum power consumption for the configuration.

	Power Estimate			
4	estimate type	minimum	typical	maximum
	[Total]	286.2mW	320.4mW	365.4mW
·p:	- VDDA	217.8mW	241.2mW	268.2mW
	- VDDD	43.2mW	46.8mW	59.4mW
	- VDD00	0uW	0uW	Ould
[]	- VDD01 - VDD02 - VDD03 - VDDREF	OuW OuW OuW	OuW OuW 10.8mW	0ulv 0ulv 12.6mlv
ட்கு	- VDDXO	16.2mW 217.8mW	21.6mW 241.2mW	25.2mW 268.2mW
	- VDDA	217.8mW	241.2mW	268.2mW
	[XIN/REFIN]	16.2mW	21.6mW	25.2mW
7	- VDDXO	16.2mW 9mW	21.6mW 10.8mW	25.2mW 12.6mW 12.6mW
	[Digital] - VDDD	43.2mW 43.2mW	46.8mW 46.8mW	59.4mW
	[OUT0]	0uW	0uW	0uW
	- VDD00	0uW	0uW	0uW
	[OUT1]	0uW	0uW	0u₩
	- VDDO1	0uW	0uW	0u₩
	[OUT2]	0u₩	0uW	0uM
	- VDDO2	0u₩	0uW	GuM
	[OUT3]	0u₩	0uW	0uW
	- VDDO3	0u₩	0uW	0uW

Figure 44. Power Estimation View Window

RENESAS

12. Cloud Updating

In addition to installing device plugins, the cloud update feature can help maintain software versions, display plugin related information, view currently installed plugins, and receive notifications of new software. To access these features, access the updates page in the RICBox GUI by going to "Help > Updates" while on the main page. View the currently installed plugins by clicking on the "Installed" tab.

s	RICBox Updates
alled	비배는 사람이 이번 관계 문제
7594	
ock7 7617	
esigner 7594	
lock2 7593	
	S alled 7594 2617 2617 2594 2594 2593

Figure 45. Installed Software

When new release software version is available, RICBox will notify users with a banner at the top of the main page when the software is initially opened.

RENESAS	RICBox	- (x c
File Help			
Info Updates Available R_VersaClock7 update is available.		View	Later
BIG IDEAS FOR EVERY SPACE			
<u></u>			
Create new project			

Figure 46. Software Update Banner

Download an external plugin package for manual PC installation by clicking on the *Download* button for an *rbpkg* file.

RICBox-Designer RICBox R_VersaClock7 R_FCle R_FemtoClock2 R_FemtoClock3 R_VersaClock7Auto R_ProXo2 R_ForXo2 R_ForXo2 R_ForXo2 R_FORXO	R_VersaClock7 R_VersaClock7 The R_VersaClock7 package. Id R_VersaClock7 Version 5.0.0-dev117617 Installed Version 5.0.0-dev117617 Architecture x64 Date Published 2022-08-12 Tags timing.versaclock7 Dependencies RICBox7	0-dev117617 • <u>install</u> Download	Name R_VersaClock7-x64-5.0.0-dev117617+03a7da4f.rbpkg R_ProXo2-x64-1.0.0-dev116026+2accec02.rbpkg
--	--	--	---

Figure 47. Download 'rbpkg' Plugin

Manually install a plugin by clicking on the Manual Install button and navigating to an 'rbpkg' file.



Figure 48. Manual Plugin Install

13. Register Write Export

Device plugins that support register write exporting will have the option to select it in the Tools menu. Click on *Register Write Export*.

Re	NES	AS	
File	Tools	Help	
~	С	u	
=	St	tatus Monitor	
	R	egister Write Export	
	چ ا	XIN_REFIN	
Ē	ମ୍ବ	CLKINO	

Figure 49. Register Write Export Button

A pop-up page will be displayed after clicking the *Register Write Export* button. The page displays the configurable settings for the exported file. Changing settings, like the Protocol, will change the transaction type in the exported file. Each setting corresponds to the format of the output file.

ř
6
ne 🖬
ERIC 🖌
4

Figure 50. Register Write Export Pop-up Page

Once the settings are selected, click the *Export* button and save the file as a text file. The exported file will be formatted like an algorithm, with each line being another serial transaction in order.

<pre># VersaClock7 (0.0.0.dev117604) # RC21008A # 2022-08-15T12:06:40 # protocol: I2C # offset size: 1 # mtu: 16 # All values are in hex # # WRITE <7-bit I2C address>, <length>, <offset>, <byte including="" offset="" stream=""> # I2C address is included because some offsets may require targeting a different address. # Length is the number of bytes in the stream, including the register offset. This could # be inferred by examining the byte stream but is included for convenience. # WAIT <time_ms> ####################################</time_ms></byte></offset></length></pre>	***************************************
<pre># RC21008A # 2022-08-15T12:06:40 # protocol: I2C # offset size: 1 # mtu: 16 # All values are in hex # # WRITE <7-bit I2C address>, <length>, <offset>, <byte including="" offset="" stream=""> # I2C address is included because some offsets may require targeting a different address. # Length is the number of bytes in the stream, including the register offset. This could # be inferred by examining the byte stream but is included for convenience. # WAIT <time_ms> ####################################</time_ms></byte></offset></length></pre>	# VersaClock7 (0.0.0.dev117604)
<pre># 2022-08-15T12:06:40 # protocol: I2C # offset size: 1 # mtu: 16 # All values are in hex # # WRITE <7-bit I2C address>, <length>, <offset>, <byte including="" offset="" stream=""> # I2C address is included because some offsets may require targeting a different address. # Length is the number of bytes in the stream, including the register offset. This could # be inferred by examining the byte stream but is included for convenience. # Offset is embedded in the byte stream but is listed separately for convenience. # WAIT <time_ms> ####################################</time_ms></byte></offset></length></pre>	# RC21008A
<pre># protocol: I2C # offset size: 1 # mtu: 16 # All values are in hex # # WRITE <7-bit I2C address>, <length>, <offset>, <byte including="" offset="" stream=""> # I2C address is included because some offsets may require targeting a different address. # Length is the number of bytes in the stream, including the register offset. This could # be inferred by examining the byte stream but is included for convenience. # Offset is embedded in the byte stream but is listed separately for convenience. # WAIT <time_ms> ####################################</time_ms></byte></offset></length></pre>	# 2022-08-15T12:06:40
<pre># offset size: 1 # mtu: 16 # All values are in hex # # WRITE <7-bit I2C address>, <length>, <offset>, <byte including="" offset="" stream=""> # I2C address is included because some offsets may require targeting a different address. # Length is the number of bytes in the stream, including the register offset. This could # be inferred by examining the byte stream but is included for convenience. # WAIT <time_ms> ####################################</time_ms></byte></offset></length></pre>	# protocol: I2C
<pre># mtu: 16 # All values are in hex # WRITE <7-bit I2C address>, <length>, <offset>, <byte including="" offset="" stream=""> # I2C address is included because some offsets may require targeting a different address. # Length is the number of bytes in the stream, including the register offset. This could # be inferred by examining the byte stream but is included for convenience. # Offset is embedded in the byte stream but is listed separately for convenience. # WAIT <time_ms> ####################################</time_ms></byte></offset></length></pre>	# offset size: 1
<pre># All values are in hex # # WRITE <7-bit I2C address>, <length>, <offset>, <byte including="" offset="" stream=""> # I2C address is included because some offsets may require targeting a different address. # Length is the number of bytes in the stream, including the register offset. This could # be inferred by examining the byte stream but is included for convenience. # WAIT <time_ms> ####################################</time_ms></byte></offset></length></pre>	# mtu: 16
<pre># # WRITE <7-bit I2C address>, <length>, <offset>, <byte including="" offset="" stream=""> # I2C address is included because some offsets may require targeting a different address. # Length is the number of bytes in the stream, including the register offset. This could # be inferred by examining the byte stream but is included for convenience. # UAIT <time_ms> ####################################</time_ms></byte></offset></length></pre>	# All values are in hex
<pre># WRITE <7-bit I2C address>, <length>, <offset>, <byte including="" offset="" stream=""> # I2C address is included because some offsets may require targeting a different address. # Length is the number of bytes in the stream, including the register offset. This could # be inferred by examining the byte stream but is included for convenience. # WAIT <time_ms> ####################################</time_ms></byte></offset></length></pre>	#
<pre># 12C address is included because some offsets may require targeting a different address. # Length is the number of bytes in the stream, including the register offset. This could # be inferred by examining the byte stream but is included for convenience. # 0ffset is embedded in the byte stream but is listed separately for convenience. # WAIT <time_ms> ####################################</time_ms></pre>	<pre># WRITE <7-bit I2C address>, <length>, <offset>, <byte including="" offset="" stream=""></byte></offset></length></pre>
<pre># Length is the number of bytes in the stream, including the register offset. This could # be inferred by examining the byte stream but is included for convenience. # Offset is embedded in the byte stream but is listed separately for convenience. # WAIT <time_ms> ####################################</time_ms></pre>	# I2C address is included because some offsets may require targeting a different address.
<pre># be inferred by examining the byte stream but is included for convenience. # Offset is embedded in the byte stream but is listed separately for convenience. # WAIT < time_ms> ####################################</pre>	# Length is the number of bytes in the stream, including the register offset. This could
<pre># Offset is embedded in the byte stream but is listed separately for convenience. # WAIT <time_ms> # wAIT <time_ms> # program all registers # write 00, 00, 00 to offset 0xFC WRITE, 00, 05, FC, FC 00 00 00 00 # write 33, 10, 00 to offset 0x00 WRITE, 00, 11, 00, 00 33 10 00 00 22 01 00 00 2A F0 12 01 00 00 01 00 # write 00, 30, 00 to offset 0x10 WRITE, 00, 11, 10, 10 00 30 00 00 00 05 80 80 00 8C 03 00 00 00 00 # write 00, 03, 22 to offset 0x20 WRITE, 00, 07, 20, 20 00 03 22 09 08 79 # write 00, 82, 00 to offset 0x27 WRITE, 00, 11, 27, 27 00 B2 00 00 04 52 86 82 001 80 00 00 180 00 # write 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 50 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 50 00 02 00 50 WRITE, 00, 50 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 50 00 02 00 5</time_ms></time_ms></pre>	# be inferred by examining the byte stream but is included for convenience.
<pre># WAII <time_ms> ####################################</time_ms></pre>	# Offset is embedded in the byte stream but is listed separately for convenience.
<pre># program all registers # program all registers # write 00, 00, 00 to offset 0xFC WRITE, 00, 05, FC, FC 00 00 00 00 # write 33, 10, 00 to offset 0x00 WRITE, 00, 11, 00, 00 33 10 00 00 22 01 00 00 2A F0 12 01 00 00 01 00 # write 00, 30, 00 to offset 0x10 WRITE, 00, 11, 10, 10 00 30 00 00 00 00 88 88 00 8C 03 00 00 00 00 # write 00, 03, 22 to offset 0x20 WRITE, 00, 07, 20, 20 00 03 22 09 08 79 # write 00, 82, 00 to offset 0x27 WRITE, 00, 11, 27, 27 00 82 00 00 04 52 86 82 01 80 00 00 180 00 # write 00, 02, 00 to offset 0x37 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50</pre>	# WAI1 <time_ms></time_ms>
<pre># program all registers # write 00, 00, 00 to offset 0xFC WRITE, 00, 05, FC, FC 00 00 00 00 # write 33, 10, 00 to offset 0x00 WRITE, 00, 11, 00, 00 33 10 00 00 22 01 00 00 2A F0 12 01 00 00 01 00 # write 00, 30, 00 to offset 0x10 WRITE, 00, 11, 10, 10 00 30 00 00 00 00 88 88 00 8C 03 00 00 00 00 # write 00, 03, 22 to offset 0x20 WRITE, 00, 07, 20, 20 00 03 22 09 08 79 # write 00, 82, 00 to offset 0x27 WRITE, 00, 11, 27, 27 00 B2 00 00 04 52 86 82 00 180 00 00 01 80 00 # write 00, 02, 00 to offset 0x37 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50</pre>	
<pre># write 00, 00, 00, 00, 10, 10 Offset 0xPC WRITE, 00, 05, FC, FC 00 00 00 00 # write 33, 10, 00, to offset 0x00 WRITE, 00, 11, 00, 00 33 10 00 00 22 01 00 00 2A F0 12 01 00 00 01 00 # write 00, 30, 00, to offset 0x10 WRITE, 00, 11, 10, 10 00 30 00 00 00 00 05 80 80 00 8C 03 00 00 00 00 # write 00, 03, 22, to offset 0x20 WRITE, 00, 07, 20, 20 00 03 22 09 08 79 # write 00, 82, 00, to offset 0x27 WRITE, 00, 11, 27, 27 00 82 00 00 04 52 86 82 00 180 00 00 01 80 00 # write 00, 02, 00, to offset 0x37 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50</pre>	# program all registers
<pre>While, 00, 00, FC, FC, 00 00 00 00 # write, 00, 00, 11, 00, 00 33 10 00 00 22 01 00 00 2A F0 12 01 00 00 01 00 # write 00, 30, 00 to offset 0x10 WRITE, 00, 11, 10, 10 00 30 00 00 00 00 05 80 80 00 8C 03 00 00 00 00 # write 00, 03, 22 to offset 0x20 WRITE, 00, 07, 20, 20 00 03 22 09 08 79 # write 00, 82, 00 to offset 0x27 WRITE, 00, 11, 27, 27 00 B2 00 00 04 52 86 82 00 180 00 00 180 00 # write 00, 02, 00 to offset 0x37 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50</pre>	# Write 00, 00, 00 to ottset 0xrc
<pre># write 03, 10, 00, 00 33 10 00 00 22 01 00 00 2A F0 12 01 00 00 01 00 # write 00, 30, 00 to offset 0x10 WRITE, 00, 11, 10, 10 00 30 00 00 00 00 05 80 80 00 8C 03 00 00 00 00 # write 00, 03, 22 to offset 0x20 WRITE, 00, 01, 20, 20 00 03 22 09 08 79 # write 00, 82, 00 to offset 0x27 WRITE, 00, 11, 27, 27 00 B2 00 00 04 52 86 82 00 180 00 00 180 00 # write 00, 02, 00 to offset 0x37 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50</pre>	Mulic, 00, 05, FC, FC 00 00 00 00
<pre># write 00, 30, 00 to offset 0x10 # write 00, 30, 00 to offset 0x10 # write 00, 30, 00 to offset 0x20 WRITE, 00, 11, 10, 10 00 30 00 00 00 05 80 80 00 8C 03 00 00 00 00 # write 00, 03, 22 to offset 0x20 WRITE, 00, 07, 20, 20 00 03 22 09 08 79 # write 00, 82, 00 to offset 0x27 WRITE, 00, 11, 27, 27 00 82 00 00 00 45 28 68 20 01 80 00 00 180 00 # write 00, 02, 00 to offset 0x37 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 20 50</pre>	π write 55, 10, 00 to oriset oxed
<pre>W Write 00, 30, 00 to offset 0.10 WRITE, 00, 11, 10, 10 00 30 00 00 00 00 80 80 80 00 8C 03 00 00 00 00 # write 00, 03, 22 to offset 0.20 WRITE, 00, 07, 20, 20 00 03 22 09 08 79 # write 00, 82, 00 to offset 0.27 WRITE, 00, 11, 27, 27 00 82 00 00 00 45 28 68 20 01 80 00 00 180 00 # write 00, 02, 00 to offset 0.37 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50</pre>	Mulic, 00, 11, 00, 00 55 10 00 02 20 100 00 24 F0 12 01 00 00 01 00
<pre>WRITE, 00, 01, 12, 10 00 00 00 00 00 00 00 00 00 00 00 00</pre>	
<pre>WRITE, 00, 07, 20, 20 00 03 22 09 08 79 # write 00, 82, 00 to offset 0x27 WRITE, 00, 11, 27, 27 00 B2 00 00 04 528 68 20 01 80 00 00 180 00 # write 00, 02, 00 to offset 0x37 WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50</pre>	$\mu_{\rm min}$ (1, 10, 11, 10, 10, 10, 10, 10, 10, 10, 1
# write 00, 82, 00 to offset 0x27 WRITE, 00, 11, 27, 27 00 B2 00 00 045 28 68 20 01 80 00 00 180 00 # write 00, 02, 00 to offset 0x37 WRITE. 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50	WRITE 00, 07, 20, 20 00 03 22 09 08 79
WRITE, 00, 11, 27, 27 00 B2 00 00 00 45 28 68 20 01 80 00 00 180 00 # write 00, 02, 00 to offset 0x37 WRITE. 00. 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50	# write 00, B2, 00, to offset 0x27
Write 00, 02, 00 to offset 0x37 WRITE, 00, 11, 37, 37,00,02,00,50,00,02,00,50,00,02,00,50,00,02,00,50	WRITE, 00, 11, 27, 27, 00, 82, 00, 00, 00, 45, 28, 68, 20, 01, 80, 00, 00, 01, 80, 00
WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50	# write 00, 02, 00 to offset 0x37
	WRITE, 00, 11, 37, 37 00 02 00 50 00 02 00 50 00 02 00 50 00 02 00 50

Figure 51. Export Register Writes

14. Multiple Configuration Support

More than one device configuration can be created at a time. Different configurations can hold differing device settings but still be saved to the same RICBox settings file (.rbs). In the lower right corner of a plugin GUI, click on the part number box to view the configuration selection drop-down menu.



Figure 52. Configuration Menu

Use this menu to change the currently active configuration. The initial configuration is always labeled default.



Figure 53. Configuration Selection

Click on the settings button to view the configurations pop-up menu.



Figure 54. Configuration Settings

	Configurations	×
1	default	Ū
2	config2	Ū
3	config3	Ū
	+ New Config	

Figure 55. Configurations Pop-up Menu

Click the numbers to change the currently active configuration. Green is the active configuration.

Configurations	×
1 default	Ū
2 config2	Ū
3 Config3	Ū
+ New Config	

Figure 56. Current Active Configuration

Change the name of any configuration by clicking on the name box and entering a new one. The trash icon button deletes the corresponding configuration. The "New Config" button creates a new configuration.

	Configurations	×	
1	default	Ū	Rename Configuration
2	RC21008_test_configuration	1	
3	config3	Ū	
			Delete Configuration
			Create New Configuration
	+ New Config		

Figure 57. Configurations Settings Overview

The CLI offers additional configuration support features. Opening the CLI and typing 'config –h' will reveal the available configuration functions.

RENESAS	RICBox CLI	-	×
> config -h usage: config [-h] [-l [copy- [name]] [-s SELECT] [-c] [-d DELETE] [-r RENAME] to COPY]		
Manage the list of con-	rigurations.		
positional arguments:			
name	The name for a new or renamed config		
optional arguments: -h,help -l,list -s SELECT,select ! -c,create -d DELETE,delete I -r RENAME,rename I copy-to COPY	show this help message and exit Display the list of configs SELECT Make the specified config active (Id, Index or Name) Create a new config DELETE Delete the specified config (Id, Index or Name) RENAME Rename the specified config (Id, Index or Name) Copy the current configuration settings (data fields, registers) to the specified config (Id, Index or Name)		

Figure 58. Configurations in the CLI

15. Revision History

Revision	Date	Description
1.02	May 22, 2024	Updated System Requirements section.
1.01	Aug 23, 2022	Updated screen shots and added new features.
1.00	Jul 2, 2021	Initial release.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES ("RENESAS") PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit <u>www.renesas.com/contact-us/</u>.