

### Contents

1	Introduction	2
2	Initial Preparation	3
	Creating a New Project	
	The "Getting Started" Example	
	Related Documents	
6	Glossary	11
7	Document Revision History	12

## List of Figures

Figure 1.1	ZWIR4512 Development Kit	2
Figure 2.1	License Manager	3
Figure 3.1	New Project Window—Select a New Project Template	5
Figure 3.2	New Project Window - Select Files to Add to the Project	6
Figure 3.3	Project Explorer	7
Figure 3.4	Project Explorer after Compiling	8
Figure 3.5	CrossStudio™ in Debugging Mode	9
Figure 4.1	COM Port Settings	10

### 1 Introduction

This document is a step-by-step manual that describes how to install all required software and demonstrates creating a simple first project with the ZWIR4512 Development Kit (see Figure 1.1). The integrated development environment (IDE) CrossStudio<sup>™</sup> for ARM®\* from Rowley Associates is used. CrossStudio<sup>™</sup> is available for Microsoft® Windows, Linux, MAC OS, and Solaris computers. This document describes the installation of the Microsoft® Windows version of CrossStudio<sup>™</sup> that is included on the ZWIR4512 Development Kit DVD. If the DVD is unavailable or a different operating system is used, Rowley CrossStudio<sup>™</sup> can be downloaded from the Rowley Associates web page: <u>http://www.rowley.co.uk/arm/index.htm</u>.

Figure 1.1 ZWIR4512 Development Kit



<sup>\*</sup> CrossStudio™ is a trademark of Rowley Associates. ARM® is a trademark of ARM, Ltd.

© 2019 Renesas Electronics Corporation

### 2 Initial Preparation

During the following steps, the development environment will be installed with all packages. These steps are only done once to set up CrossStudio<sup>™</sup>.

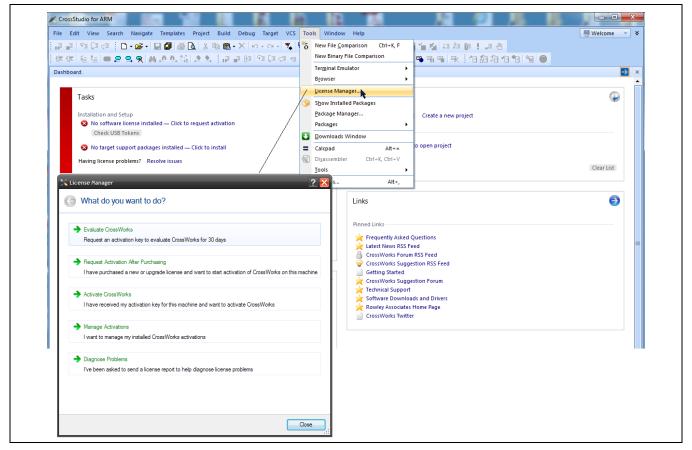
1. Install CrossStudio<sup>™</sup> for ARM from the DVD by activating the following executable:

```
install \verb|CrossStudio|arm_crossworks_2_3_5\_win\_x86\_setup.exe|
```

Alternately, download and install CrossStudio<sup>™</sup> for ARM from Rowley's website: <u>http://www.rowley.co.uk/arm/releases.htm</u>

- 2. A valid license is required before CrossStudio<sup>™</sup> can be used. For initial testing, it is possible to get a 30day evaluation license. Get the license by following these steps after installing CrossStudio<sup>™</sup> and starting the application:
  - a. Open the license manager by clicking on the menu bar *Tools → License Manager*. See Figure 2.1 for the resulting dialog box.

Figure 2.1 License Manager



- b. Click on the *Evaluate CrossWorks* button, and follow the steps.
- c. After receiving the activation key, click on the *Activate CrossWorks* button in the "License Manager" dialog box and enter the key.
- 3. To support the STM32 microcontroller and the ZWIR4512 modules, the corresponding support package must be installed from the DVD or downloaded from the web.

The ZWIR4512 Board Support Package can be downloaded\* from the ZWIR4512 product page on IDT's website: <u>http://www.IDT.com/ZWIR4512</u>.

The STM32 CPU Support Package can be downloaded from Rowley's website: <u>http://www.rowleydownload.co.uk/arm/packages/STM32.htm</u>.

Install the SMT32 CPU and ZWIR4512 Board Support Packages by double-clicking each of the following files in the package:

install\CrossStudio\CMSIS\_3.hzq install\CrossStudio\STM32.hzq install\CrossStudio\ZWIR4512.hzq

4. The connection between the microcontroller JTAG and UART interface and the development PC is via a USB virtual COM port interface, provided by a FTDI chip. This chip requires installation of a special VCP driver from the DVD or downloaded from the FTDI website: <u>http://www.ftdichip.com/Drivers/VCP.htm</u>

Install the VCP driver by clicking on the following file:

install\FTDI\CDM20830\_Setup.exe

 Connect the ZWIR4512 test board to the user's PC via the USB cable included in the kit. The PC operating system will automatically install the newly found USB device. An appropriate installation guide can be found under <u>http://www.ftdichip.com/Support/Documents/InstallGuides.htm</u>

During the installation, two new COM ports are created. The first one performs the communication between CrossStudio<sup>™</sup> and the microcontroller JTAG interface. The second one is connected to the microcontroller UART interface.

Access to on-line ZWIR45xx software downloads requires a free customer login account. See section 7. After login, click on "ZWIR4512 API RCS" under the "Software" heading.

### 3 Creating a New Project

After the successful initial preparation, a first simple application can demonstrate the development process with CrossStudio<sup>™</sup>.

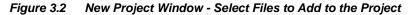
- 1. Start CrossStudio™.
- 2. In the menu bar, select *File → New Project*.
- 3. In the *New Project* window, complete the following steps:
  - a. Scroll to the ZMDI section as shown in Figure 3.1.
  - b. In the project template window, click on 6LoWPAN project template targeted at ZWIR4512AC1xx modules or 6LoWPAN project template targeted at ZWIR4512AC2xx modules depending on which module is used.
  - c. Enter a project name, and select a project path. Note: creating a separate folder for each project is recommended.
  - d. Click Next.

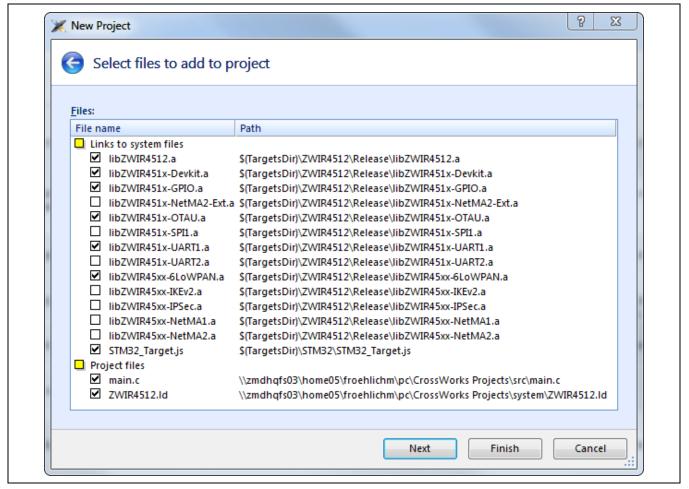
Figure 3.1 New Project Window—Select a New Project Template

Select new project template	Search Pro	ojects	
$(\mathbf{j})$ Don't see your device or board? Use Tools > Package Manager to install p	ackages.		
Description	Manufacturer	Board	•
ZMDI			
6LoWPAN project template targeted at ZWIR4512AC1xx modules 6LoWPAN project template targeted at ZWIR4512AC2xx modules	ZMDI ZMDI	ZWIR4512AC1 ZWIR4512AC2	
6LoWPAN project template targeted at ZWIR4512AC1xx modules		ZWIR4512AC2	₹ Prowse



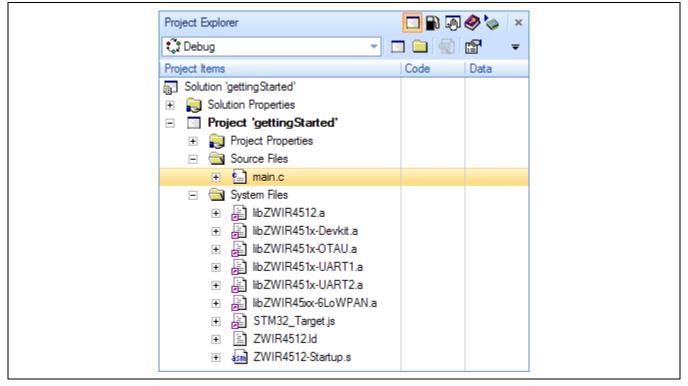
- 4. Select files to add to the project.
  - a. Select all files as shown in Figure 3.2.
  - b. Ensure that the libraries *libZWIR45xx-IKEv2.a* and *libZWIR45xx-IPSec.a* are NOT selected. Note: If the libraries *libZWIR45xx-IKEv2.a* and/or *libZWIR45xx-IPSec.a* are selected, communication over IPv6 is not possible without configuring the security module.
  - c. Click Finish.





5. At this point, the project has been created and all required files have been added to the project. The *Project Explorer* on the right side shows all project files. All libraries, the startup code, the linker script, and the reset script can be found in the *System Files* folder as shown in Figure 3.3. The application-specific code is in the *Source Files* folder. For this example, open the automatically created *main.c* file by double clicking it.





6. Write the program.

A first example is the "getting started". The code can be found on the DVD under

#### src/Examples/GettingStarted/src/gettingStarted.c.

It is also included in the IDT software zip file available for download\* on the ZWIR4512 product web page: <a href="https://www.IDT.com/ZWIR4512">www.IDT.com/ZWIR4512</a>.

To use this example, the "getting started" c-code (from the *gettingStarted.c*) must be copied into the current project *main.c* (Figure 3.3)

For more details about the "getting started" program, see section 4.

<sup>\*</sup> Access to on-line ZWIR45xx software downloads requires a free customer login account. See section 7. After login, the gettingStarted.c example code is available under the "Software" heading.

<sup>© 2019</sup> Renesas Electronics Corporation

7. Compile the program:

Right click on the project in the *Project Explorer* and then click on *Build* on the resulting drop-down menu. Figure 3.4 shows the "Project Explorer" after the program has been compiled for this example.

Figure 3.4 Project Explorer after Compiling

Pro	ject l	Explo	rer			) 🔊	🤣 🍖 🛛 🗴
ि	🕄 Debug 👻			💌 🖾 💼	I 😨 🛛	<b>8</b>	▼
Pro	ject l	tems			Code		Data
62	Solu	tion	'gettingStarted'				
-	3	Solu	tion Properties				
	+	۹	_debug				
	+	× .	_release				
Ξ	1	_	ject 'gettingStarted'		41	,188	15,772
	+		Project Properties				
	-	_	Source Files				
			📄 main.c			864	136
	-		System Files				
		+	📄 libZWIR4512.a				
		+	ibZWIR451x-Devkit.a				
		+	IbZWIR451x-OTAU.a				
		+	ibZWIR451x-UART1.a				
		+	IbZWIR451x-UART2.a				
		+	☐ libZWIR45∞-6LoWPA	N.a			
		+	STM32_Target.js				
		+	ZWIR4512.ld				
		+	Sm ZWIR4512-Startup.s			104	
	+	5	Output Files				

Note: The following steps must be done only at the project initialization.

- 8. Connect to the device:
  - a. On the main menu bar, click **Target**  $\rightarrow$  **Targets**.
  - b. Right click in the opened *Target* window.
  - c. Select New Target Interface → Generic FT2232 Device.
  - d. On the main menu bar, click *Target → Connect → Generic FT2232 Device*.

- 9. Start debugging:
  - a. On the main menu bar, click **Debug**  $\rightarrow$  **Go** (or press F5).
  - b. Wait until the application was loaded successfully at the microcontroller. Note: if uploading errors occur, try the uploading again.
- 10. Start the program by pressing the *Continue Execution* button 10 on the main menu or press F5.

Figure 3.5 CrossStudio™ in Debugging Mode

🖌 gettingStarted - CrossStudio for ARM (Stopped)										
Ele Edit View Search Project Build Debug Target VCS Browser Tools Window Help Debug										
⇔ ⇒  ↓ ⊉  ☜ [⊈ [⊈ ]: D + 😰 + 🖳 🕼 🙆 🔏 🐁 🛍 🕲 ×   い + ↔ → 🌾 💭 + 🚰 + 🥬 ! 🛄 getingStated 💦 👘 🏠  ↓ 💷 🕼  ↓ ↓ → 🔶										
揮 律 ! 智 性 🖸 🗢 鸟 🗙 🏚 🛎 🍇 🎍 🖢 🚛 運 匪 1 智 匣 禪 律 理 [禪 傳 [韓 ] 🎆 🍕 🏂 🛛 🕼 / 尋 🔞 + : [(Uhknown function)??? 👘 🔹 🔍 🔍 🔍 👒 👒 👒 😪										
<u>রি বি 'ব রি 'বা 'বা 'বা 'বা 'বা 'বা 'বা 'বা 'বা 'বা</u>										
assembly		Sec. X		/ main.c			Breakpoints			<u> </u>
n	17 B	I <sup>6</sup> ∃ (∃ c∃ ¢] ▼	🔁 😼 🐼		D 🖓 📲 🧕	🗊 🖅 🗇 🖓	🐚 • 🕅 🛷 🔮	2 🚷 🚳 🎨 🖄		
08001D7A	480B	ldr r3, 0x08001DA 🔺	#include <sto< td=""><td>fio.h&gt; ER45xx-6LOWPAN.h"</td><td></td><td><b>^</b></td><td>📄 Breakpoir</td><td></td><td></td><td></td></sto<>	fio.h> ER45xx-6LOWPAN.h"		<b>^</b>	📄 Breakpoir			
08001D7C 08001D7E	781A 3A01	ldrb r2, [r3] subs r2, #1		IR45XX-BLOWPAN.N IR45XX-UART1.h"			🕀 🛃 CM3 Excep	tions		
08001D7E	B2D2	uxtb r2, r2		IR451x-Devkit.h"						
08001D82	701A	strb r2, [r3]								
08001D84	781A	ldrb r2, [r3]	ZWIR_UART1_PF	RINTF		//enable print				
08001D86	B94A	cbnz r2 0x08001D9	THITE TRUCADA	ress_t addrAll = { IPV6_ALLNODES };		//all node ada				
08001D88 08001D8A	8672 4A08	cpsid i ldr r2, 0x08001DA		ress t addrUnsp = { IPV6_ALLHODES },	D 3:	//unspecified				
08001D8A	8811	ldrh r1, [r2]	10		,					
08001D8E	B289	uxth r1, r1	char buffer[3			//receive buff				
08001D90	F0410180	orr r1, r1, #0x80	char bufferPt	t = 0;		//receive poin	Watch 1		😼 💭 😂	20 🗇
08001D94	8011	strh r1, [r2]	void SerialRe	eceiveEvent ( void ) {		//UART CALL ba	Never	- x <sub>2</sub> x <sub>8</sub> x <sub>10</sub> x <sub>16</sub> ±x	< 😒 🖪 🗍 🏭 💆	an Ka
08001D96 08001D98	B662 220A	cpsie i movs r2, #10		UART1_ReadByte ( &buffer[bufferPt]	) ){	//read Byte fr	Name	Value		0 00
08001D9A	701A	strb r2, [r3]	bufferPt	++;		··· · · ·	bufferPt	0x00		
08001D9C	E8BD4001	pop.w {r0, lr}		ffer[bufferPt-1] == '\r')    (buffer			DUTTERPL	0000		
08001DA0	4685	mov sp, r0	ZWIR_Se     buffer	endUDP2 ( buffer, bufferPt, &addrAll	, 1000 );	//send receive				
08001DA2 08001DA4	4770 2055	bx lr		tLED1(1);		//switch LED 1				
08001DA4	0800	movs r0, #0x55 lsrs r0, r0, #0	<ul> <li>return</li> </ul>							
08001DA8	2874	cmp r3, #0x74	> }							
08001DAA	2000	movs r0, #0	)							
08001DAC	012C	lsls r4, r5, #4	• 3							
08001DAE	2000 main>	movs r0, #0	void DataCall	Lback (unsigned char* data, unsigned	<pre>short size) {</pre>	//UDP Socket c				
08001DB0	8510	push {r4, 1r}	<ul> <li>int a;</li> </ul>				Registers 1	<b>1</b>	<b>X</b>	
08001DB2	B672	cpsid i		ddress_t *sender = ZWIR_ <mark>Get</mark> PacketSen	derAddress();	//get packet s	-			
08001DB4 08001DB6	4820 781A	ldr r3, 0x08001E3		:eived data from "); 0; a < 16; ++ a )		//print sender	<ul> <li>Register Group</li> </ul>	s x <sub>2</sub> x <sub>8</sub> x <sub>10</sub> x <sub>16</sub> ±	< '×' → 😳 🜆 🗞	· 💼
08001DB5	F00202FC	ldrb r2, [r3] and r2, r2, #0xFC		"%02x:", sender -> u8 [ a ] );		//prene senser	Name	Value		
08001DBC	F36F0282	bfc r2, #2, #1	ZWIR_SetLE			//switch LED 3 🚽	▼ CPU			
08001DC0	701A	strb r2, [r3]	4	-85.		•				
08001DC2	2200	movs r2, #0					r0	0×00000000		
08001DC4 08001DC6	605A 4B1D	str r2, [r3, #4] ldr r3, 0x08001E3	Output	4	> 🖸 🗿 💭 🛄		r1 r2	0x00000000 0x08001db1		
08001DC8	4798	blx r3	Show: Transcript	- 🍢 🐾 Tasks -		-	8 12	0x00000001		
08001DCA	481D	ldr r3, 0x08001E4	v Completed				r3 r4	0x00000000		
08001DCC	4798	blx r3	🎲 Downloading "Lo	ader_rpc.elf <sup>-</sup> to Generic FT2232 Dev	1.9 KB in 0.4s	01	r5	0×000000000		
08001DCE 08001DD0	481D 4798	ldr r3, 0x08001E4 blx r3	Completed		4.3 KB/s	ок	n6	0×00000000		
08001DD0	4798 481D	ldr r3, 0x08001E4	🕆 Verifying "Loade	r_rpc.elf" on Generic FT2232 Device	1.9 KB in 0.0s		r7	0×00000000		
08001DD4	4798	blx r3	Verifying "Loader Completed		24.5 KB/s	ок	n8	0×00000000		
08001DD6	481D	ldr r3, 0x08001E4	Prasing Coetting	Started.elf to Generic FT2232 Device			🗌 r9	0×00000000		
08001DD8 08001DDA	4798 481D	blx r3 ldr r3, 0x08001E5	Completed	Autorion to denote i izzdz Device	20.8 KB/s	ок	n10	0×00000000		
08001DDA 08001DDC	481D 4604	mov r4, r0		ttingStarted.elf" to Generic FT2232 D			n11	0×00000000		
08001DDC	B103	cbz r3 0x08001DE2	Completed	aungstated.en to Generic F12232 D	40.5 KB in 2.2s	ОК	n12	0×2000c000		
08001DE0	4798	blx r3		C			sp(r13)	0x2000c000		
08001DE2	4B1C	ldr r3, 0x08001E5	Completed	Started.elf on Generic FT2232 Devic	40.9 KB in 0.9s 41.6 KB/s	ОК	lr(r14)	0x0800020b		
08001DE4 08001DE6	B10B 4620	cbz r3 0x08001DEA mov r0, r4		e			pc(r15)	0x08001db0 0x61000000		
03001050	4020	10V 10, 14 V	Completed	for user application	Completed in 0.1	s ок 🖵		0x61000000 0x2000c000		
			<ul> <li>Completed</li> </ul>			· · · · · ·	or_main	0720000000		

It is possible to debug the uploaded program in CrossStudio<sup>™</sup>. All debug features are explained in CrossStudio's Help function accessible via the main menu.

### 4 The "Getting Started" Example

The "getting started" example provides a basic introduction to the ZWIR4512-I-Development Kit. This program can be found on the DVD under src\examples\gettingstarted. For this example, at least two devices must be pre-programmed with the "getting started" code.

This example shows how to receive and send UDP packets. It also shows how data can be transmitted over the UART Interface.

During module initialization, two UDP sockets (port 1000 to receive data and port 1001 to receive blink packets) are opened and the UART receiver is enabled to receive user input.

Every 5 seconds, the program sends one blink packet to all nodes. The sending device indicates packet transmission with LED 2 while the receiving device indicates packet reception with LED 4. The receiver writes a message containing the sender's address to the serial interface.

All characters received by the UART interface are saved in a buffer until a return character is received or the buffer runs out of free space. In this case, the device broadcasts the buffer contents to all other nodes and indicates transmission with LED 1. Receiving devices will write the transmitted buffer contents to their serial interface and indicate packet reception with LED 3.

To use this example, connect at least one device with a PC and power the other devices either with an external power supply or also via the PC.

Open a UART terminal (e.g., Microsoft® HyperTerminal) on the PC and establish a connection to the *second* virtual COM port with the parameters set as in Figure 4.1.

COM17 Properties Port Settings		
<u>B</u> its per second:	115200	
<u>D</u> ata bits:	8	
Parity:	None	
<u>S</u> top bits:	1 🗸	
Elow control:	None	
	<u>R</u> estore Defaults	

Figure 4.1 COM Port Settings



If any other "getting started" device is powered, the terminal should print out the received blink packets as

Received blink from fe80:0000:0000:0211:7d00:002f:0003: 5232

To test the data transfer function, connect a second device and open a second terminal. Now it is possible to enter a message (e.g., "hello world" in the example below) in one of the terminals and transmit it to the other one by pressing "return." The second terminal should display the received message as follows (note that the sender address will differ from this example):

```
Received data from fe80:0000:0000:0211:7d00:002f:0003: hello world
```

#### 5 Related Documents

Document
ZWIR4512 Data Sheet
ZWIR451x Programming Guide*

Visit the ZWIR4512 product page <u>www.IDT.com/ZWIR4512</u> or contact your nearest sales office for the latest version of these documents.

\*Note: Documents marked with an asterisk (\*) require a free customer login account for access.

#### 6 Glossary

Term	Description			
СОМ	COM Name of the serial port interface			
JTAG Joint Test Action Group				
UART Universal Asynchronous Receiver/Transmitter				
UDP User Datagram Protocol				
VCP	Virtual COM Port			

Revision	Date	Description
1.00	October 25, 2010	First release.
1.10	March 25, 2011	Improved "getting started" example.
1.20	May 21, 2011	Improved specification of library setup. Added figure explaining COM port settings.
1.30	June 18, 2012	Updated installation instructions.
1.31	July 23, 2012	Minor edits for clarity. Minor corrections.
1.32	August 26, 2013	Changed example images to match new CrossWorks™ version.
1.40	October 2, 2014	Changed example images to match new libraries.
	April 15, 2016	Changed to IDT branding.

# 7 Document Revision History

#### 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 Jan 2024)

#### **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>.