RX600 Series

CAN-UART Bridge

R01AN0721EU0210
Rev. 2.10
Sep 22, 2012

Introduction

The MCU firmware that can be downloaded with this application note demonstrates how the RX MCU can provide an easy solution to interfacing between asynchronous serial devices (UART) and CAN.

Renesas CAN-equipped MCUs, such as those in the RX600 Series have integrated asynchronous serial port (UART) peripherals. This application note together with its source code demonstrate using the RX MCU as a "bridge" for bidirectional communication between a PC RS232 connection and a CAN network.

Data entered by the user at the PC terminal will be sent to the RX MCU over the RS232 serial link. This data is interpreted by the RX MCU CAN-UART Bridge software and the data is encoded into CAN data packets which the RX MCU then transmits onto the CAN bus. Conversely, the RX MCU receives commands and data packets from the CAN bus, decodes them, and passes the data on to the terminal back over the RS232 serial link.

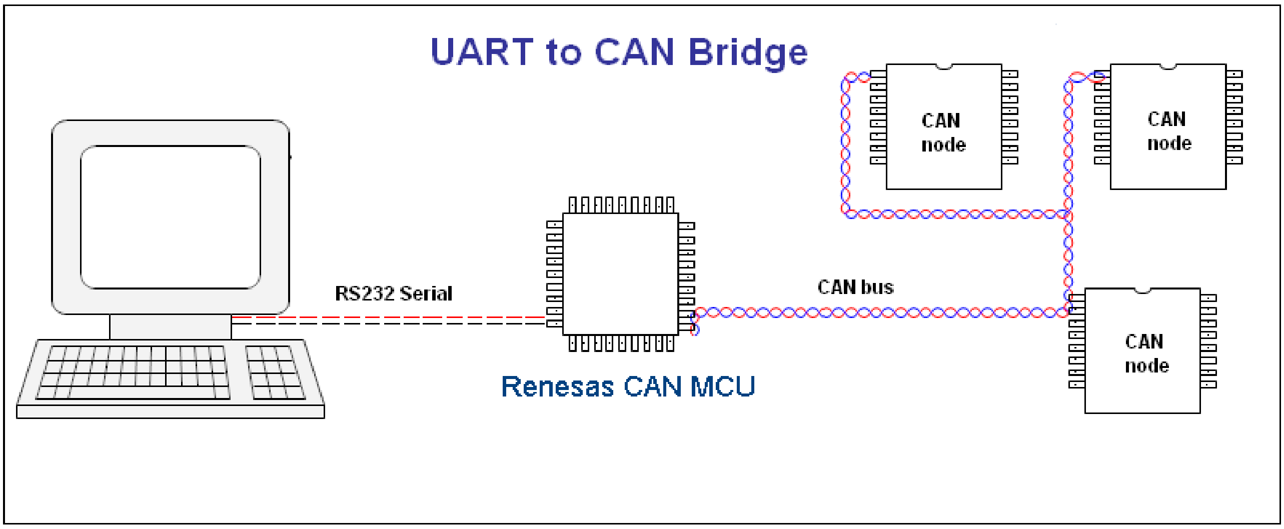

Figure 1 The RX MCU as a UART-CAN gateway between a PC and a CAN bus. Data can travel to and from the CAN bus and will all be shown on the PC terminal.

Target Devices

RX600 Series MCUs with CAN. Source code for most RX RSKs come with this application note.

Related Documents

Refer to the CAN API specification and application note R01AN0339EU for using the CAN driver.

See also schematics for the particular board you are using. To date, this code is ported to boards RSK-RX62N, YRDK62N, RSK-RX62T, RSK-RX630, RSK-RX630, and RSK-RX62G.

Contents

1. A CAN to UART Gateway

This application note describes the use of a Renesas CAN-equipped RX600 Series MCU as a serial UART to CAN bus "Bridge". A demonstration program serves as an effective example of how the RX MCU, with its UART peripheral and CAN module, becomes a communication gateway between the two interfaces.

The CAN-UART Bridge application will act as a CAN 'sniffer' by monitoring the CAN bus and transmitting all data over to the serial port (UART/RS232). The serial data can be viewed on a PC with the use of a generic terminal program such as "HyperTerminal". You can also send messages from the PC terminal program over the Bridge *to* the CAN bus.

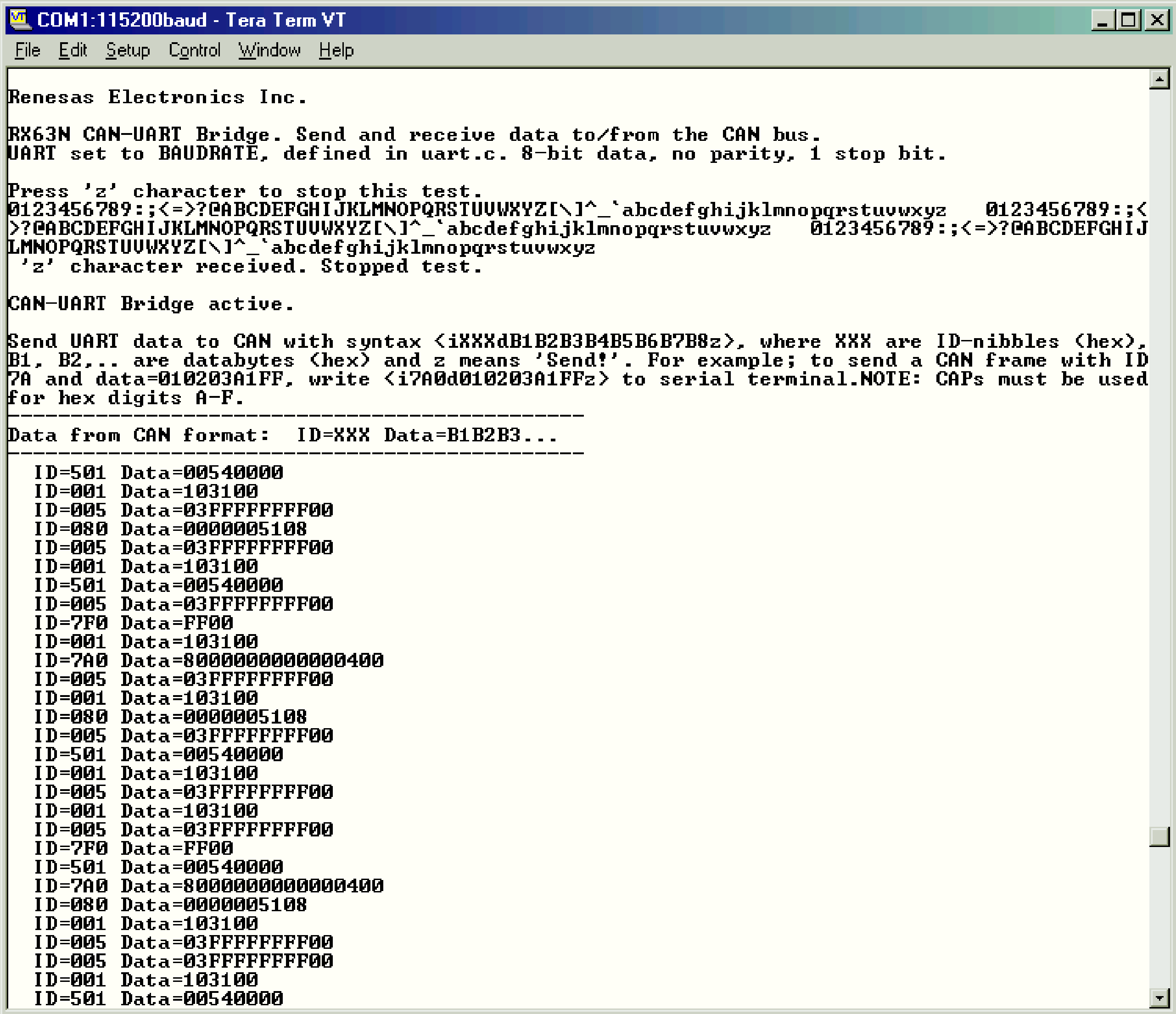

Figure 2. Screenshot of a PC terminal, to which the CAN-UART Bridge streams CAN data from a CAN bus attached to the RSK. A user can send data to the CAN bus from the terminal by entering data using syntax <iXXXdB1B2B3B4B5B6B7B8z>. XXX is the CAN ID (11-bit) and B1B2… data bytes. XXX and B1B2… are hexadecimal characters 0-F. Capital letters must be used for A-F.

1.1 Buffering of CAN Bus Data

Typically, the CAN bus will operate at a much higher baudrate than the UART. The CAN bus may be operating at 500kBaud while the UART is running at 115.2kBaud as in the example code, and CAN transmissions on a bus tend to occur in bursts rather than being continuous. This raises the possibility of the CAN data overrunning the capacity of the UART to forward them, resulting in lost dataframes. To handle the speed differences between the CAN bus and the UART baud rates, incoming CAN data is automatically buffered. Therefore, buffering of the CAN dataframes in RAM provides the UART time to send the data from the MCU. Should the buffer overflow, a message will appear in the serial data stream to increase the buffer for incoming CAN data.

2. Renesas Development Boards Supported

Table 1

Renesas MCU	Board	Development board model order#
RX62N	RSK-62N	R0K5562N0S000BE
RX62N	YRDK62N	YRDKRX62N
RX62T	RSK-62T	R0K5562T0S000BE
RX630	RSK-630	R0K505630C011BR
RX63N	RSK-63N	R0K50563NC00BR

This demonstration runs on any Renesas RX development board shown in Table 1.

The Renesas CAN API (application programming interface in download package R01AN0339) , consisting of `r_can_api.h`, `r_can_api.c`, and `config_r_can_rapi.h` simplifies development of CAN applications by providing all the driver and protocol level routines needed to control the CAN hardware. See the CAN API application note in that download for full details on how to use the R_CAN API. Note that nothing in `r_can_api.c` should be changed, only make changes to `config_r_can_rapi.h`. You can always download the very latest Renesas CAN API driver package (R01AN0339) and replace these three files.

3. 'Bill of Materials' for Running the CAN-UART Bridge

Besides this application note and the source code, download HEW and the RX toolchain (compiler and linker) and debugger for E1 and E20 from the Renesas website.

Here is a complete list of items to obtain:

- High-performance Embedded Workshop (HEW) programming environment, toolchain and debugger
- CAN_UART_Bridge_RX6xx project HEW workspace package
- CAN API application note: R01AN0339EU0110
- RX6xx MCU development board (Table 1)
- E1 or E20 emulator
- RS232 cable with DB9 connectors
- Systec CAN bus analyzer GW-002, 32040xx, or similar (optional)
- PC Terminal application (e.g. HyperTerminal or TeraTerm)

4. Building the Project

4.1 Tools

Refer to the installation guide for your Renesas RX600 Series development board for details on connecting the board with the E1 or E20 emulator.

To build the UART- CAN Bridge target executable, open the HEW workspace project for your particular board by double-clicking on the file CAN_UART_Bridge_RX6xx.hws.

For help with HEW software, refer to the High-performance Embedded Workshop User's Manual .pdf file.

4.2 Configuring the CAN API

Table 2 shows files typically contained in the CAN UART Bridge project.

Table 2

Source files	Header files:
can_uart_bridge.c	**config_r_can_rapi.h**
r_can_api.c	can_uart_bridge.h
dbsct.c	r_can_api.h
hwsetup.c	iodefine.h
intprg.c	lcd.h
lcd.c	rskRX6xxdef.h
resetprg.c	sbrk.h
sbrk.c	stacksct.h
uart.c	typedefine.h
vecttbl.c	uart.h
	vect.h

The *can_uart_bridge.c* source file contains the main application routines.

The file *config_r_can_rapi.h* is intended to keep the user from having to change anything in the CAN driver files *r_can_api.h* and *r_can_api.c*. When using the CAN API for your own hardware, there are other settings that you may want to change in *config_r_can_rapi.h*. For example, change the CAN interrupt priority, the transceiver pins used if porting to another board, or changing the CAN baudrate settings. The CAN bus data-rate is, by default, configured for: 500 kbit/s.

To be able to run the CAN-UART Bridge the demo-application must be configured to use CAN interrupts.

See the Renesas CAN API application note (R01AN0339EU) for details on using the CAN API.

5. Operation

The CAN-UART Bridge will send CAN data via the target MCU over to the UART so that it shows up on for example a PC terminal program. You can also send data to the CAN bus from the PC terminal. Here follows some checklists to get the CAN-UART Bridge running.

5.1 Board connections

1. Connect the E1 or E20 debugger/emulator to the board and download the application. If you have an RSK, the Quick Start Guide shows you how to open a HEW project and download/debug code.
2. Connect a UART cable from your PC's RS232 COM port to the MCU running CAN-UART Bridge. If your CAN-UART Bridge board does not have a DSUB-9 connector installed, refer to the board schematic to find the UART RX and TX pins to connect directly.
3. Connect the CAN-UART Bridge board's CAN bus to a CAN network. Remember that the CAN bus should have a 120 ohm resistor at each end. Your total resistance value as measured between the two twisted wires will be 60 ohms. (Ideally there should not be resistors at the nodes, but this applies to large networks of longer distances larger than an office cubicle or test bench).

5.2 Serial Port Settings

1. In the default firmware settings the UART configuration is: 8 data bits, 1 stop bit, no parity, no flow control.
2. Check the BAUDRATE for actual speed. This will be defined in the `uart.c` file and on some model boards will also be displayed on the LCD at startup.
3. Set the serial communication port settings for your terminal application to match the CAN-UART Bridge program settings.

5.3 Serial Terminal Command Syntax

When you get a correct serial connection you will see a stream of characters in the PC terminal. Send 'z' with the PC keyboard to stop this 'loop' and to enter CAN-UART-Bridge mode.

The syntax to send data via the bridge to the CAN bus is

iXXXdB1B2B3B4B5B6B7B8z

Where: **i** signifies that a three hex-digit standard CAN ID to follow ('i' must be entered in this position).

d signifies that a zero to eight byte hex-digit payload data is to follow ('d' must be entered in this position).

z = means 'send now'.

Note: When entering hex values A-F, use only CAPITALS.

For example, to send a CAN message with CAN ID 0x07A0 and data bytes 0x09 0x0A 0x0B 0xCC, enter **i7A0d090A0BCCz** into the UART terminal.

5.4 Board LED Indication

During operation, the program will light various LEDs to provide some indication of status. For RSK boards, the LEDs can be interpreted as follows, though the actual LED used may vary between boards:

- LED0 (Green most boards): UART received data.
- LED1 (Orange most boards): UART transmitted data.
- LED2 (Red most boards): CAN received a dataframe.
- LED3 (Red most boards): CAN is transmitted a dataframe.
- All LEDs On: CAN is in an error state.

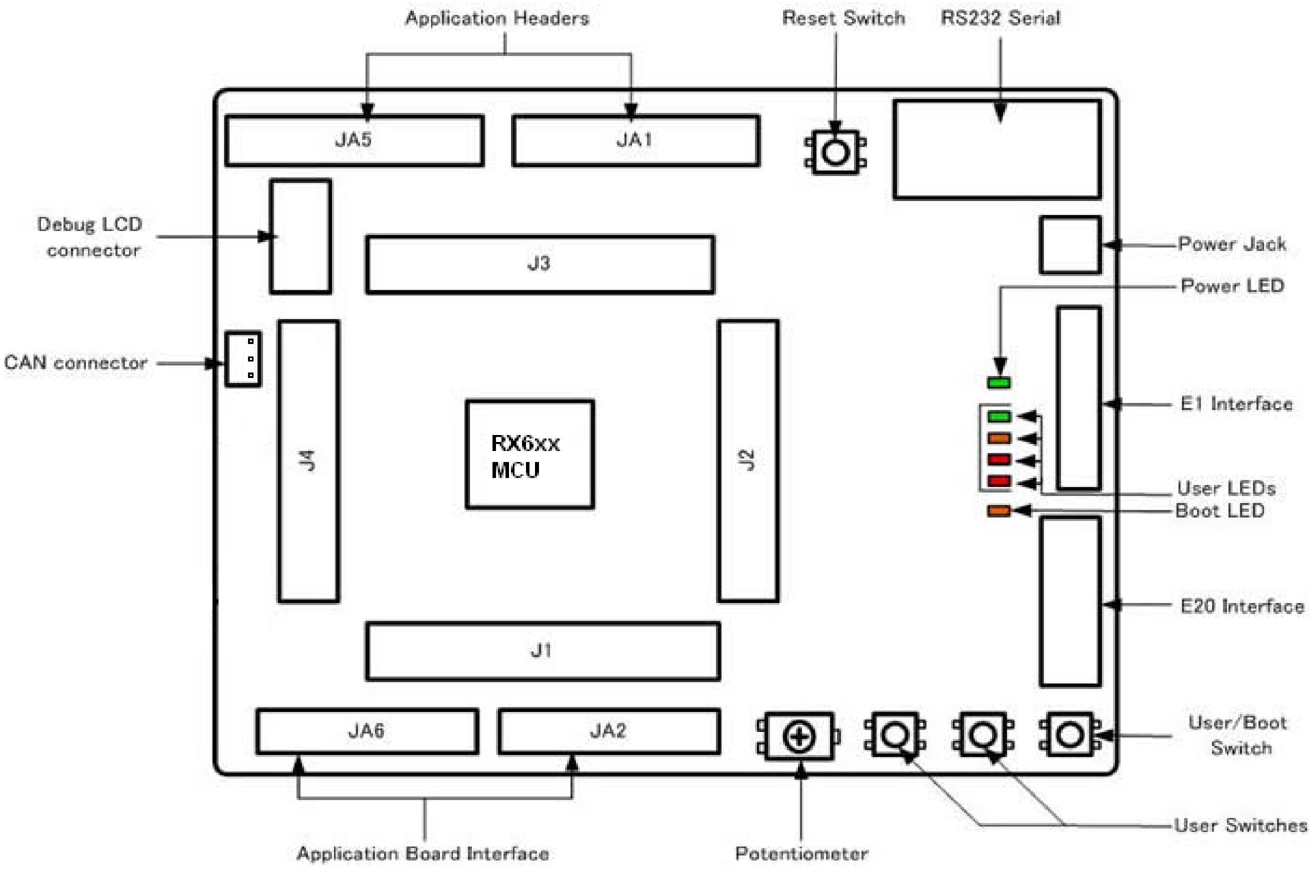

Figure 3 Typical layout of RX600 Series RSK development board, with the LED indicators to the right.

6. Using the Systec CAN Bus Analyzer

A convenient way to demonstrate the UART- CAN Bridge is by using a CAN bus analyzer such as the Systec CANmodul. This is the Systec "CAN sniffer" that comes with the Renesas CAN development Kits from Renesas, and which is used for the Flash over CAN application note and firmware (R01AN0235EU_RX).

The Systec CANmodul can be ordered separately from Systec-Electronics, Germany. The sniffer connects to the PC USB host and to a CAN network. On the CAN side the Systec acts as a node on the CAN bus. On the PC side, a GUI monitors received data and can also send CAN dataframes onto the CAN bus through the Systec module.

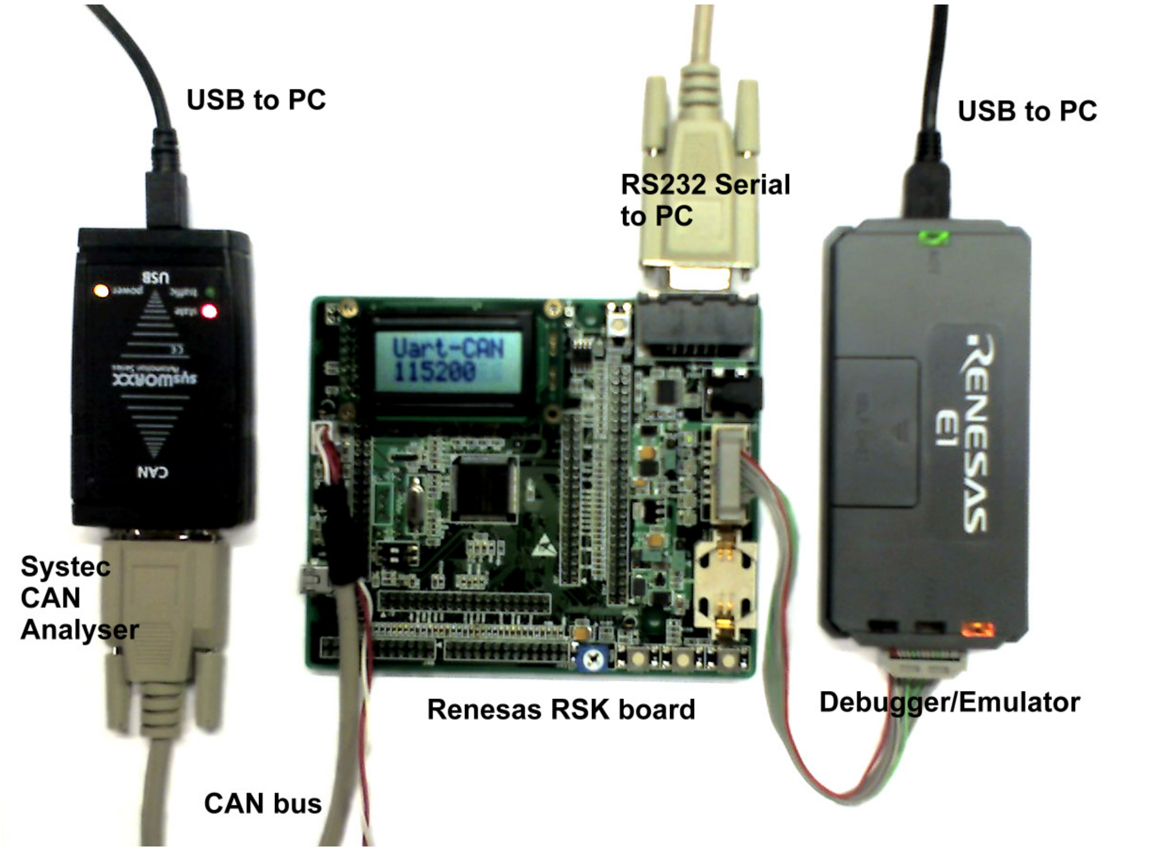

Figure 4 RSK RX630 in CAN UART Bridge test setup

7. Support

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Jun 30, 2010	—	First edition issued
2.00	Jan 15, 2012	1	Figure of PC terminal added. Workspace for 63N added.
2.10	Sep 22, 2012	1	Workspace for 62G added.

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 document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

   Handle unused pins in 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 a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

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: +44-1628-651-700, Fax: +44-1628-651-804

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

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

Renesas Electronics Malaysia Sdn.Bhd.
Unit 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 Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141