RX72M Group

EtherCAT CiA402 Sample Program Firmware Information Technology

Summary
This application note describes a sample program which supports the CiA402 drive profile that is particularly for use in motor control through EtherCAT® communications. EtherCAT® is an Ethernet communications protocol for industrial applications.

This application note includes the sample code for the CiA402 drive profile which uses the EtherCAT FIT module.

Target Device
- RX72M Group

If you intend to use the program in a product, you need to evaluate it thoroughly in terms of suitability for the given environment.

When applying the program covered in this application note to another microcontroller, modify the program to suit the specifications of the target microcontroller and extensively evaluate the program after modification.
Table of Contents

1. Overview ..............................................................................................................................3
   1.1 About This Application Note ..........................................................................................3
   1.2 Operating Environment .................................................................................................3
   1.3 FIT Module Configuration .............................................................................................3
   1.4 The Projects ..................................................................................................................3

2. Obtaining a Development Environment ............................................................................4
   2.1 How to Obtain e² studio ..................................................................................................4
   2.2 How to Obtain the Compiler Package .............................................................................4

3. Building a Project ...............................................................................................................4
   3.1 Importing the EtherCAT Slave Stack Code to the Sample Program .........................4
   3.2 Importing a Project .........................................................................................................7
   3.3 Changed Information .......................................................................................................8
      3.3.1 Changing the File Configuration of the EtherCAT FIT Module .............................8
      3.3.2 Changing the Configurations ....................................................................................8

4. Checking the Operation .....................................................................................................9
   4.1 Preparation for Debugging ..............................................................................................9
   4.2 Building and Debugging the Project ..............................................................................9

5. Connection with TwinCAT ................................................................................................12
   5.1 Preparing the ESI File ....................................................................................................12
   5.2 Starting up TwinCAT .....................................................................................................12
   5.3 Scanning the Network ....................................................................................................12
   5.4 Writing to the SII-Manufactured EEPROM .................................................................13
   5.5 Rescanning the Device ..................................................................................................14
   5.6 Checking the Operation Mode .......................................................................................14
      5.6.1 CiA402 State Transition ..........................................................................................14
      5.6.2 csp Mode ................................................................................................................15
      5.6.3 csv Mode ................................................................................................................16

6. CiA402 Drive Profile .........................................................................................................18
   6.1 Operation Modes ...........................................................................................................18
   6.2 State Transition ..............................................................................................................19
   6.3 Object Dictionary ...........................................................................................................19
   6.4 Implementing the Motor Control Program ....................................................................21

7. Documents for Reference ....................................................................................................27
1. Overview

1.1 About This Application Note

This application note describes the procedure through to the evaluation of the sample program for the CiA402 drive profile and the FIT modules such as that for EtherCAT and the board support package (BSP) in combination.

The sample program covered in this application note runs on the Renesas Starter Kit+ for RX72M (hereafter "the RSK board") and the RX72M evaluation board from Tessera Technology (hereafter "the communications board").

1.2 Operating Environment

Table 1-1 Testing Environment

<table>
<thead>
<tr>
<th>Supported MCU</th>
<th>RX72M Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation board</td>
<td>Renesas Starter Kit+ for RX72M (product type name: RTK5572MNxCxxxxxBJ)</td>
</tr>
<tr>
<td></td>
<td>RX72M evaluation board TS-RX72M-COM from Tessera Technology</td>
</tr>
<tr>
<td>Integrated development environment (IDE)</td>
<td>e² studio V.7.5.0 from Renesas Electronics</td>
</tr>
<tr>
<td></td>
<td>IAR Embedded Workbench for Renesas RX 4.13.1</td>
</tr>
<tr>
<td>Cross tool</td>
<td>C/C++ Compiler Package for RX Family V3.01.00 from Renesas Electronics</td>
</tr>
<tr>
<td></td>
<td>GCC for Renesas RX 4.8.4.201803</td>
</tr>
<tr>
<td></td>
<td>IAR C/C++ Compiler for Renesas RX version 4.13.1</td>
</tr>
<tr>
<td>Emulator</td>
<td>E2 Lite</td>
</tr>
</tbody>
</table>

1.3 FIT Module Configuration

The sample program covered in this application note is configured with the use of the following FIT modules.

Table 1-2 FIT Module Configuration

<table>
<thead>
<tr>
<th>Type</th>
<th>Module Name</th>
<th>FIT Module Name</th>
<th>Rev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Board Support Package</td>
<td>Board support package (BSP)</td>
<td>r_bsp</td>
<td>5.20</td>
</tr>
<tr>
<td>Device Driver</td>
<td>Compare-match timer (CMT)</td>
<td>r_cmt_rx</td>
<td>3.40</td>
</tr>
<tr>
<td>Device Driver</td>
<td>Serial communications interface (SCI)</td>
<td>r_sci_rx</td>
<td>2.20</td>
</tr>
<tr>
<td>Middleware</td>
<td>Byte queue buffer (BYTEQ)</td>
<td>r_byteq</td>
<td>1.71</td>
</tr>
<tr>
<td>Middleware</td>
<td>System time module</td>
<td>r_sys_time_rx</td>
<td>1.00</td>
</tr>
<tr>
<td>Device Driver</td>
<td>EtherCAT</td>
<td>r_ecat_rx</td>
<td>1.10</td>
</tr>
</tbody>
</table>

1.4 The Projects

The projects covered in this application note are listed below.

The following sections describe the project for the RX72M communications board as an example. If you will be using the project for RSK board, read the statements referring to the communications board as the RSK board as required.

Table 1-3 List of Projects

<table>
<thead>
<tr>
<th>MCU</th>
<th>Evaluation Board Name</th>
<th>Project Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>RX72M</td>
<td>RSK board</td>
<td>ecat_cia402_demo_rskrx72m</td>
</tr>
<tr>
<td></td>
<td>Communications board</td>
<td>ecat_cia402_demo_comrx72m</td>
</tr>
</tbody>
</table>
2. Obtaining a Development Environment

2.1 How to Obtain e² studio
Access the following URL and download the e² studio.


This application note assumes that you will be using V7.5.0 or a later version of the e² studio. If you are using a version earlier than V7.5.0, some functions of the e² studio may not be available.

When downloading the e² studio, obtain the latest version on the website.

2.2 How to Obtain the Compiler Package
Access the following URL and download the RX family C/C++ Compiler Package.


3. Building a Project

3.1 Importing the EtherCAT Slave Stack Code to the Sample Program
This project does not include the EtherCAT Slave Stack Code.

* The EtherCAT Slave Stack Code (SSC) tool is required to generate the EtherCAT Slave Stack Code.
* The SSC tool can be obtained from EtherCAT Technology Group (ETG).

The sample program is provided in the zip format, "ecat_cia402_demo_comrx72m.zip". Unzip it in the desired folder in advance.

(1) Double-click on the SSC project file of the sample program to start the SSC tool.

   ecat_cia402_demo_comrx72m\src\smc_gen\r_ecat_rx\utilities\rx72m\ssc_config\RX72M EtherCAT CiA402.esp

(2) Click on [Project] → [Create new Slave Files]. In the [Current new Slave Files] dialog box, click on [Start].
(3) The source code is generated. If this is successful, the message "New files created successfully" will appear, so click on [OK].
(4) If the patch command is not installed on your PC, you will need to install ver. 2.5.9 or a later version of GNU patch.

If it is already installed, skip this step.

Download the patch command (currently ver. 2.5.9) from the following Web page and store "patch.exe" in a folder on a path that makes the file executable from the command prompt.

http://gnuwin32.sourceforge.net/packages/patch.htm

(5) Right-click on the apply_patch.bat file and select [Run as an administrator] → [Yes].

The patch file contains modifications to make the SSC source file suitable for the RX family.

ecat_cia402_demo_comrx72m\src\smc_gen\r_ecat_rx\utilities\rx72m\batch_files\apply_patch.bat

After execution of the patch, the modified source file is stored in the following folder.

ecat_cia402_demo_comrx72m\src\application\ecat

```
--- Move SCC Src folder ---
  1 dir(s) moved.
--- Patching process start ---
  patching file Src\cia402appl.c
  patching file Src\cia402appl.h
  patching file Src\ecatcoe.h
  patching file Src\mailbox.h
  patching file Src\sockserv.h
--- Patching process end ---
--- Move patched Src folder ---
  1 dir(s) moved.
Press any key to continue . . .
```
3.2 Importing a Project

(1) Click on [File] → [Import].

(2) In the [Select] dialog box, select [General] → [Existing Projects into Workspace] and click on [Next].

(3) In the [Import Projects] dialog box, select the [Select root directory] checkbox and click on [Browse].

(4) Select "ecat_cia402_demo_comrx72m" as a sample project file for the communications board and click on [Open].

(5) Check "ecat_cia402_demo_comrx72m" under the [Projects] label and click on [Next]. The project will be imported into the workspace.
3.3 Changed Information
In this project, the file configuration of the EtherCAT FIT module and the settings in the configuration files for a few other FIT modules have been changed to configure them for use with the sample program. The details are described below.

Refer to the changed information when building a new project. When using the imported project, proceed to section 4, Checking the Operation.

3.3.1 Changing the File Configuration of the EtherCAT FIT Module
The configuration of the files for the EtherCAT FIT module has been partly modified to create the sample program for the CiA402 drive profile.

The table below lists the files with configurations to be changed.

Table 3-1 List of Changed Files for the EtherCAT FIT Module

<table>
<thead>
<tr>
<th>Folder Name</th>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>src\appl</td>
<td>cia402sample.c</td>
<td>The CiA402 sample program</td>
</tr>
<tr>
<td></td>
<td>cia402sample.h</td>
<td></td>
</tr>
<tr>
<td>utilities\rx72m\batch_files</td>
<td>apply_patch.bat</td>
<td>The patch file to apply modifications in the sample program to the slave stack source file generated by the SSC tool and the batch file to apply patches</td>
</tr>
<tr>
<td></td>
<td>RX72M_CiA402_xxxxxx.patch</td>
<td></td>
</tr>
<tr>
<td>utilities\rx72m\esi</td>
<td>RX72M EtherCAT CiA402.xml</td>
<td>The ESI file for the CiA402 sample program</td>
</tr>
<tr>
<td>utilities\rx72m\ssc_config</td>
<td>RX72M EtherCAT CiA402.esp</td>
<td>The SSC tool project file for the CiA402 sample program</td>
</tr>
</tbody>
</table>

3.3.2 Changing the Configurations
Change the configuration files for the other FIT modules where this is required to configure them for use with the sample program.

For the items and settings in the configuration files, refer to the manuals, etc. in the doc folder for each FIT module.

The table below lists the changes in configurations.

Table 3-2 List of Changes in Configurations

<table>
<thead>
<tr>
<th>Item to be Changed</th>
<th>FIT Module Name</th>
<th>Configuration File Name</th>
<th>Macro Name</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increase in heap size</td>
<td>BSP</td>
<td>r_bsp_config.h</td>
<td>BSP_CFG_HEAP_BYTES</td>
<td>0x8000</td>
</tr>
<tr>
<td>Enabling SCI CH6</td>
<td>SCI</td>
<td>r_sci_rx_config.h</td>
<td>SCI_CFG_CH6_INCLUDED</td>
<td>1</td>
</tr>
<tr>
<td>Enabling the transmission completed interrupt</td>
<td>SCI</td>
<td>r_sci_rx_config.h</td>
<td>SCI_CFG_TEI_INCLUDED</td>
<td>1</td>
</tr>
</tbody>
</table>
4. Checking the Operation

4.1 Preparation for Debugging
This section describes how to set up the evaluation board to run this sample program.

Table 4-1    Evaluation Board Settings

<table>
<thead>
<tr>
<th>Item for Setting</th>
<th>MCU</th>
<th>Evaluation Board Name</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAN Cable</td>
<td>RX72M</td>
<td>RSK board</td>
<td>Connect to &quot;CN2 IN&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Communications board</td>
<td>Connect to &quot;ECAT IN&quot;</td>
</tr>
</tbody>
</table>

(1) Connect the LAN cable as described in Table 4-1.

(2) Connect the E2 Emulator Lite to the JTAG connector on the evaluation board with the user interface cable.

(3) Connect the E2 Emulator Lite to a USB port of the PC you are using. The "ACT" LED on the E2 Emulator Lite starts to blink.

(4) The "Found new hardware" wizard appears. Follow the procedure described below to install the driver. Installation on a PC running Windows™ 7, 8, or 8.1 requires logging in with a user name that has administrator rights.
   Windows™ 7/8/8.1: A notice appears on the Windows taskbar when the installation is finished.
   Windows™ 10: A button for device configuration appears on the Windows taskbar and installation is automatic.

(5) Supply power to the evaluation board.

4.2 Building and Debugging the Project

(1) Left-click on the "ecat_cia402_demo_comrx72m" project in the Project Explorer view. Click on the arrow next to the build button (hammer icon) and select [Hardware Debug] from the drop-down menu.
(2) e² studio will build the project. After building is completed, click on the arrow next to the [Debug] button (bug icon). You can start debugging by selecting [Debug Configurations].

(3) Click on "ecat_cia402_demo_comrx72m Hardware Debug" and download the program to the target device. Press the [Debug] button to start debugging.
(4) If a firewall warning is displayed for 'e2-server-gdb.exe', check the checkbox for [Private networks, such as my home or work network] and click on [Allow access].

(5) The User Account Control (UAC) dialog box may appear. Enter the administrator's password and click on [Yes].

(6) If the Confirm Perspective Switch dialog box appears prompting you to switch the perspective, check the checkbox for [Remember my decision] and click on [Yes].

(7) The green "ACT" LED on the E2 Lite debugger will be continuously lit.

(8) After downloading the program, click on the [Resume] button to run the code to the first line of the main function main (). If you click on the [Resume] button again, the target will run the remaining program.
5. **Connection with TwinCAT**

This section describes the procedure for operating the sample program using TwinCAT3.

5.1 **Preparing the ESI File**

Before starting up TwinCAT, copy the ESI file included in the sample program to the specified location for TwinCAT (`\TwinCAT\3.x\Config\I/O\EtherCAT`).

```
ecat_cia402_demo_comrx72m\src\smc_gen\r_ecat_rx\utilities\rx72m\esi\RX72M EtherCAT CiA402.xml
```

5.2 **Starting up TwinCAT**

1. From the start menu, select `[Beckhoff] → [TwinCAT3] → [TwinCAT XAE (VS20xx)]`.

2. After starting the program, select `[File] → [New] → [Project]` to create a new project of the TwinCAT XAE Project type.

5.3 **Scanning the Network**

1. In the System Manager tree, right-click on `[I/O] → [Devices]` and select `[Scan]`.

2. In the `[HINT: Not all types of devices can be found automatically]` dialog box, click on `[OK]`.

3. In the `[new I/O devices found]` dialog box, select the checkbox for the Ethernet adapter to be scanned and click on `[OK]`.

4. In the `[Scan for Boxes]` dialog box, click on `[Yes]`.

5. The "EtherCAT drive(s) added. Append linked axis to NC-Configuration" dialog box will be displayed. Click on `[Yes]`.

6. The "Active Free Run" dialog box will be displayed. Click on `[Yes]`.

Scanning was successful if a box has been added, e.g. "Device 1" → "Box 1" under "I/O" → "Devices", in the System Manager tree.
5.4 Writing to the SII-Manufactured EEPROM

* The EEPROM on the communications board is blank at the time of shipment, so you will need to write to the board’s EEPROM.
* The procedure described in this section is not required if the EEPROM has already been programmed.

If the EEPROM is blank, Box 1 is displayed in a form like "Box1 (PFFFFFFFF RFFFFFFFF)" in the System Manager tree.

(1) If you double-click on [Box 1] in the System Manager tree, a panel will be displayed on the right side of the screen.

(2) Select the [EtherCAT] tab and click on the [Advanced Settings] button.

(3) In the hierarchy of the tree on the left side of the [Advanced Settings] dialog box, select [ESC Access] → [EEPROM] → [Hex Editor].

(4) In the [Hex Editor] dialog box, select "Download from list".

(5) In the [Write EEPROM] dialog box, select [Renesas Electronics Corp.] → [Renesas RX72M Group] → [RX72M EtherCAT CiA402] and click on [OK]. Data are written to the EEPROM.

(6) After programming, restart the communications board (by resupplying power or applying a reset) to make sure that the written data are reflected in the operation of the microcontroller.
5.5 Rescanning the Device

(1) Press the [Restart TwinCAT (Config Mode)] button.

(2) In the [Restart TwinCAT System in Config Mode] dialog box, click on [OK].

(3) In the [Load I/O Devices] dialog box, click on [Yes].

(4) In the [Active Free Run] dialog box, click on [Yes].
   This will be OK if "Box 1" in the System Manager tree has turned to "Box 1 (RX72M EtherCAT
   CiA402)".

5.6 Checking the Operation Mode

(1) If you double-click on [Box 1] in the System Manager tree, a panel will be displayed on the right side of
   the screen.

(2) Select the [Online] tab and check that "Current Status" has turned to "OP".

(3) In the System Manager tree, expand + on the left side of "Box 1".

5.6.1 CiA402 State Transition

To check the operation in csp and csv modes, the state must be changed to "Operation Enabled" in both
modes.

Change the state by setting a value in the "Control Word" object and check the state by confirming the value
of the "Status Word" object.

(1) Select "Outputs" → "Control Word" in the System Manager tree and then select the [Online] tab on
   the right-side panel. A value will then be displayed.
   Click on [Write] and set values from [7] to [15] in that order.

(2) Select "Inputs" → "Status Word" in the System Manager tree and then select the [Online] tab on the
   right-side panel. A value will then be displayed.
   If the value is [4663], the state is "Operation Enabled". Proceed to the next step.
   If the value is [4616], the state is "Fault" for some reason. Set "Control Word" to [128] once and
   return to step (1).
5.6.2 **csp Mode**

Check that "Module 1" is displayed as "Module 1 (csp-axis)".

(1) Select "Outputs" → "Target Position" in the System Manager tree and then select the [Online] tab on the right-side panel. A value will then be displayed.

  Click on [Write] and set a desired value.

  Here, set [100000] as an example.

(2) Select "Inputs" → "Actual Position" in the System Manager tree and then select the [Online] tab on the right-side panel. A value will then be displayed.

  Check that the value set in "Target Position" allows incrementation up to [100000].
5.6.3 csv Mode

(1) In the System Manager tree, select "Box 1 (RX72M EtherCAT CiA402)" and then select the [Slots] tab on the right-side panel.

(2) If you select "Axis 1" under "Slot" in the left frame of the tabbed page, modules which can be added will be displayed in the right frame of the page, so add "csv-axis".

(3) Check that "Module 2 (csv-axis)" has been added in the System Manager tree.

(4) Follow the procedure described in section 5.6.1 to change the state to "Operation Enabled".

(5) Select "Outputs" → "Target Velocity" in the System Manager tree and then select the [Online] tab on the right-side panel. A value will then be displayed.

   Click on [Write] and set a desired value.

   Here, set [100000] as an example.

(6) Select "Inputs" → "Actual Velocity" in the System Manager tree and then select the [Online] tab on the right-side panel. A value will then be displayed.

   Check that the value set in "Target Velocity" allows incrementation up to [100000].
Box 1 (RX72M EtherCAT CiA402)
- Module 1 (csp_axis)
- Module 2 (csp_axis)

Inputs
- Error Code
- Status Word
- ActualPosition

(6) Actual Velocity
- Actual Torque
- Touch Probe Status
- Touch Probe Position 1 Positive Value
- Touch Probe Position 2 Positive Value
- Digital Inputs

Outputs
- Control Word
- Max Torque
- Velocity Offset
- Torque Offset
- Touch Probe Function
- Positive Torque Limit Value
- Negative Torque Limit Value
- Physical Outputs

(5) Target Velocity
6. CiA402 Drive Profile

The CiA402 drive profile is a device profile for driving motors and motion control and mainly defines functional operations for servo drives, sine-wave inverters and stepping motor controllers. In this profile, the multiple operation modes and corresponding parameters are defined as an object dictionary. Also, Finite State Automaton (FSA) to define the internal and external behavior in every state is included. When changing the state, the result after transition is reflected in the status word object that shows the current state by specifying the state through the control word object. The control word and various command values (such as speed) are assigned to RxPDO, and the status word and various real values (such as position) are assigned to TxPDO. Please see the contents of the CiA402 standard for more details.

![CiA402 Communication Flow](image)

Figure 6-1   CiA402 Communication Flow

6.1 Operation Modes

In the application note, the following modes are supported from among the operation modes defined in the CiA402 standard.

<table>
<thead>
<tr>
<th>Operation Mode</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile position mode</td>
<td>No</td>
</tr>
<tr>
<td>Velocity mode (frequency converter)</td>
<td>No</td>
</tr>
<tr>
<td>Profile velocity mode</td>
<td>No</td>
</tr>
<tr>
<td>Profile torque mode</td>
<td>No</td>
</tr>
<tr>
<td>Homing mode</td>
<td>No</td>
</tr>
<tr>
<td>Interpolated position mode</td>
<td>No</td>
</tr>
<tr>
<td>Cyclic synchronous position mode</td>
<td>Yes</td>
</tr>
<tr>
<td>Cyclic synchronous velocity mode</td>
<td>Yes</td>
</tr>
<tr>
<td>Cyclic synchronous torque mode</td>
<td>No</td>
</tr>
<tr>
<td>Cyclic synchronous torque mode with commutation angle</td>
<td>No</td>
</tr>
<tr>
<td>Manufacturer specific mode</td>
<td>No</td>
</tr>
</tbody>
</table>
6.2 State Transition

In this application note, the following is supported as FSA defined in the CiA402 standard.

![CiA402 State Transition Diagram](image)

Figure 6-2 CiA402 State Transition Diagram

6.3 Object Dictionary

The following is the list of the object dictionaries supported in this application note.

<table>
<thead>
<tr>
<th>Operation Mode</th>
<th>OBJECT Name</th>
<th>INDEX</th>
<th>Category</th>
<th>Access</th>
<th>Data Type</th>
<th>PDO Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclic synchronous position mode</td>
<td>Position actual value</td>
<td>0x6064</td>
<td>Mandatory</td>
<td>ro</td>
<td>INT32</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Following error window</td>
<td>0x6065</td>
<td>Optional</td>
<td>rw</td>
<td>UINT32</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Velocity actual value</td>
<td>0x606C</td>
<td>Conditional</td>
<td>ro</td>
<td>INT32</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Max torque</td>
<td>0x6072</td>
<td>Optional</td>
<td>rw</td>
<td>UINT16</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Torque actual value</td>
<td>0x6077</td>
<td>Conditional</td>
<td>ro</td>
<td>INT16</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Target position</td>
<td>0x607A</td>
<td>Optional</td>
<td>rw</td>
<td>INT32</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Software position limit</td>
<td>0x607D</td>
<td>Optional</td>
<td>c,rw</td>
<td>INT32</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Target position</td>
<td>0x607A</td>
<td>Optional</td>
<td>rw</td>
<td>INT32</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Software position limit</td>
<td>0x607D</td>
<td>Optional</td>
<td>c,rw</td>
<td>INT32</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Position offset</td>
<td>0x60B0</td>
<td>Optional</td>
<td>rw</td>
<td>INT32</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Velocity offset</td>
<td>0x60B1</td>
<td>Optional</td>
<td>rw</td>
<td>INT32</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Torque offset</td>
<td>0x60B2</td>
<td>Optional</td>
<td>rw</td>
<td>INT32</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Following error actual value</td>
<td>0x60F4</td>
<td>Optional</td>
<td>ro</td>
<td>INT32</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Target velocity</td>
<td>0x60FF</td>
<td>Conditional</td>
<td>rw</td>
<td>INT32</td>
<td>Yes</td>
</tr>
<tr>
<td>Function Group</td>
<td>OBJECT Name</td>
<td>INDEX</td>
<td>Category</td>
<td>Access</td>
<td>Data Type</td>
<td>PDO Mapping</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------------------</td>
<td>-------</td>
<td>----------</td>
<td>--------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Torque Limiting</td>
<td>Positive torque limit value</td>
<td>0x60E0</td>
<td>Conditional</td>
<td>rw</td>
<td>UINT16</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Negative torque limit value</td>
<td>0x60E1</td>
<td>Conditional</td>
<td>rw</td>
<td>UINT16</td>
<td>Yes</td>
</tr>
<tr>
<td>Homing</td>
<td>Home Offset</td>
<td>0x607C</td>
<td>Optional</td>
<td>rw</td>
<td>INT32</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Homing speeds</td>
<td>0x6099</td>
<td>Conditional</td>
<td>c,rw</td>
<td>UINT32</td>
<td>No</td>
</tr>
<tr>
<td>Touch Probe</td>
<td>Touch probe function</td>
<td>0x60B8</td>
<td>Optional</td>
<td>rw</td>
<td>UINT16</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Touch probe status</td>
<td>0x60B9</td>
<td>Optional</td>
<td>ro</td>
<td>UINT16</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Touch probe pos 1 pos value</td>
<td>0x60BA</td>
<td>Optional</td>
<td>ro</td>
<td>INT32</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Touch probe pos 2 pos value</td>
<td>0x60BC</td>
<td>Optional</td>
<td>ro</td>
<td>INT32</td>
<td>Yes</td>
</tr>
<tr>
<td>Gear ratio</td>
<td>Gear ratio</td>
<td>0x6091</td>
<td>Optional</td>
<td>c,rw</td>
<td>UINT32</td>
<td>No</td>
</tr>
<tr>
<td>Other object</td>
<td>Error code</td>
<td>0x603F</td>
<td>Optional</td>
<td>ro</td>
<td>UINT16</td>
<td>Yes</td>
</tr>
<tr>
<td>Controlling the power</td>
<td>Controlword</td>
<td>0x6040</td>
<td>Mandatory</td>
<td>rw</td>
<td>UINT16</td>
<td>Yes</td>
</tr>
<tr>
<td>drive system</td>
<td>Statusword</td>
<td>0x6041</td>
<td>Mandatory</td>
<td>ro</td>
<td>UINT16</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Quick stop option code</td>
<td>0x605A</td>
<td>Optional</td>
<td>rw</td>
<td>INT16</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Shutdown option code</td>
<td>0x605B</td>
<td>Optional</td>
<td>rw</td>
<td>INT16</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Disable operation option code</td>
<td>0x605C</td>
<td>Optional</td>
<td>rw</td>
<td>INT16</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Halt option code</td>
<td>0x605D</td>
<td>Optional</td>
<td>rw</td>
<td>INT16</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Fault reaction option code</td>
<td>0x605E</td>
<td>Optional</td>
<td>rw</td>
<td>INT16</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Modes of operation</td>
<td>0x6060</td>
<td>Optional</td>
<td>rw</td>
<td>INT8</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Modes of operation disp</td>
<td>0x6061</td>
<td>Optional</td>
<td>ro</td>
<td>INT8</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Supported drive modes</td>
<td>0x6502</td>
<td>Mandatory</td>
<td>ro</td>
<td>INT32</td>
<td>No</td>
</tr>
<tr>
<td>General object</td>
<td>Motor type</td>
<td>0x6402</td>
<td>Optional</td>
<td>rw</td>
<td>INT16</td>
<td>No</td>
</tr>
<tr>
<td>Position control</td>
<td>Position demand value</td>
<td>0x6062</td>
<td>Optional</td>
<td>ro</td>
<td>INT32</td>
<td>No</td>
</tr>
<tr>
<td>function</td>
<td>Position actual internal value</td>
<td>0x6063</td>
<td>Optional</td>
<td>ro</td>
<td>INT32</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Position window</td>
<td>0x6067</td>
<td>Optional</td>
<td>rw</td>
<td>UINT32</td>
<td>No</td>
</tr>
<tr>
<td>Optional application</td>
<td>Digital inputs</td>
<td>0x60FD</td>
<td>Optional</td>
<td>ro</td>
<td>UINT32</td>
<td>Yes</td>
</tr>
<tr>
<td>FE</td>
<td>Digital outputs</td>
<td>0x60FE</td>
<td>Optional</td>
<td>c,rw</td>
<td>UINT32</td>
<td>No,Yes</td>
</tr>
</tbody>
</table>
6.4 Implementing the Motor Control Program

According to the CiA402 standard from the list of CiA402 protocol stack I/F functions in Table 6-3, implement the motor control application. Each function links the number of each state transition of CiA402 FSA shown in Figure 6-2 and the corresponding function is called in case of state transition. In each function, describe the processing that calls the motor control program or the relevant processing of the main CPU.

| CiA402_StateTransition1 | Description | This function is used when state transition 1 has occurred.  
 Describe the operation in the case of the state transition. |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Usage</td>
<td>#include &quot;cia402appl.h&quot;</td>
</tr>
<tr>
<td></td>
<td>Parameters</td>
<td>TCiA402Axis *pCiA402Axis</td>
</tr>
</tbody>
</table>
|                         | Return Value| 0 Normal end  
 1 Error                                                        |
|                         | Remark      | In the case of error occurrence during processing, exit the function by setting  
 the appropriate values for each object in accordance with the CiA402 standard.  
 If 1 is set to return value, state transition does not occur. |

| CiA402_StateTransition2 | Description | This function is used when state transition 2 has occurred.  
 Describe the operation in the case of the state transition. |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Usage</td>
<td>#include &quot;cia402appl.h&quot;</td>
</tr>
<tr>
<td></td>
<td>Parameters</td>
<td>TCiA402Axis *pCiA402Axis</td>
</tr>
</tbody>
</table>
|                         | Return Value| 0 Normal end  
 1 Error                                                        |
|                         | Remark      | In the case of error occurrence during processing, exit the function by setting  
 the appropriate values for each object in accordance with the CiA402 standard.  
 If 1 is set to return value, state transition does not occur. |
### CiA402_StateTransition3

<table>
<thead>
<tr>
<th>Description</th>
<th>This function is used when state transition 3 has occurred. Describe the operation in the case of the state transition.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usage</td>
<td><code>#include &quot;cia402app1h&quot;</code></td>
</tr>
<tr>
<td>Parameters</td>
<td><code>TCIA402Axis *pCiA402Axis</code></td>
</tr>
<tr>
<td>Return Value</td>
<td>0 Normal end</td>
</tr>
<tr>
<td></td>
<td>1 Error</td>
</tr>
<tr>
<td>Remark</td>
<td>In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.</td>
</tr>
</tbody>
</table>

### CiA402_StateTransition4

<table>
<thead>
<tr>
<th>Description</th>
<th>This function is used when state transition 4 has occurred. Describe the operation in the case of the state transition.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usage</td>
<td><code>#include &quot;cia402app1h&quot;</code></td>
</tr>
<tr>
<td>Parameters</td>
<td><code>TCIA402Axis *pCiA402Axis</code></td>
</tr>
<tr>
<td>Return Value</td>
<td>0 Normal end</td>
</tr>
<tr>
<td></td>
<td>1 Error</td>
</tr>
<tr>
<td>Remark</td>
<td>In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.</td>
</tr>
</tbody>
</table>

### CiA402_StateTransition5

<table>
<thead>
<tr>
<th>Description</th>
<th>This function is used when state transition 5 has occurred. Describe the operation in the case of the state transition.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usage</td>
<td><code>#include &quot;cia402app1h&quot;</code></td>
</tr>
<tr>
<td>Parameters</td>
<td><code>TCIA402Axis *pCiA402Axis</code></td>
</tr>
<tr>
<td>Return Value</td>
<td>0 Normal end</td>
</tr>
<tr>
<td></td>
<td>1 Error</td>
</tr>
<tr>
<td>Remark</td>
<td>In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.</td>
</tr>
</tbody>
</table>
### CiA402_StateTransition6

**Description**
This function is used when state transition 6 has occurred. Describe the operation in the case of the state transition.

**Usage**
#include "cia402appl.h"

**Parameters**
TCA402Axis *pCiA402Axis

**Return Value**
0 Normal end
1 Error

**Remark**
In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.

### CiA402_StateTransition7

**Description**
This function is used when state transition 7 has occurred. Describe the operation in the case of the state transition.

**Usage**
#include "cia402appl.h"

**Parameters**
TCA402Axis *pCiA402Axis

**Return Value**
0 Normal end
1 Error

**Remark**
In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.

### CiA402_StateTransition8

**Description**
This function is used when state transition 8 has occurred. Describe the operation in the case of the state transition.

**Usage**
#include "cia402appl.h"

**Parameters**
TCA402Axis *pCiA402Axis

**Return Value**
0 Normal end
1 Error

**Remark**
In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.
## CiA402_StateTransition9

<table>
<thead>
<tr>
<th>Description</th>
<th>This function is used when state transition 9 has occurred. Describe the operation in the case of the state transition.</th>
</tr>
</thead>
</table>

### Usage
#include "cia402applh"

### Parameters
TCiA402Axis *pCiA402Axis

### Return Value
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Normal end</td>
</tr>
<tr>
<td>1</td>
<td>Error</td>
</tr>
</tbody>
</table>

### Remark
In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.

## CiA402_StateTransition10

<table>
<thead>
<tr>
<th>Description</th>
<th>This function is used when state transition 10 has occurred. Describe the operation in the case of the state transition.</th>
</tr>
</thead>
</table>

### Usage
#include "cia402applh"

### Parameters
TCiA402Axis *pCiA402Axis

### Return Value
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Normal end</td>
</tr>
<tr>
<td>1</td>
<td>Error</td>
</tr>
</tbody>
</table>

### Remark
In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.

## CiA402_StateTransition11

<table>
<thead>
<tr>
<th>Description</th>
<th>This function is used when state transition 11 has occurred. Describe the operation in the case of the state transition.</th>
</tr>
</thead>
</table>

### Usage
#include "cia402applh"

### Parameters
TCiA402Axis *pCiA402Axis

### Return Value
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Normal end</td>
</tr>
<tr>
<td>1</td>
<td>Error</td>
</tr>
</tbody>
</table>

### Remark
In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.
**CiA402_StateTransition12**

**Description**  
This function is used when state transition 12 has occurred.  
Describe the operation in the case of the state transition.

**Usage**
```
#include "cia402appl.h"
```

**Parameters**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCiA402Axis *pCiA402Axis</td>
<td></td>
</tr>
</tbody>
</table>

**Return Value**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Normal end</td>
</tr>
<tr>
<td>1</td>
<td>Error</td>
</tr>
</tbody>
</table>

**Remark**

In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.

---

**CiA402_LocalError**

**Description**  
This function is used when state transition 13 has occurred.  
Describe the operation in the case of the state transition.

**Usage**
```
#include "cia402appl.h"
```

**Parameters**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UINT16 ErrorCode</td>
<td></td>
</tr>
</tbody>
</table>

**Return Value**

None

**Remark**

If the error corresponding to state transition 13 occurs, call this function after processing required and saving data at error location.

---

**CiA402_StateTransition14**

**Description**  
This function is used when state transition 14 has occurred.  
Describe the operation in the case of the state transition.

**Usage**
```
#include "cia402appl.h"
```

**Parameters**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCiA402Axis *pCiA402Axis</td>
<td></td>
</tr>
</tbody>
</table>

**Return Value**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Normal end</td>
</tr>
<tr>
<td>1</td>
<td>Error</td>
</tr>
</tbody>
</table>

**Remark**

In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.
### CiA402_StateTransition15

**Description**
This function is used when state transition 15 has occurred. Describe the operation in the case of the state transition.

**Usage**
```
#include "cia402appl.h"
```

**Parameters**
- `TCiA402Axis *pCiA402Axis`

**Return Value**
- 0: Normal end
- 1: Error

**Remark**
In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.

### CiA402_StateTransition16

**Description**
This function is used when state transition 16 has occurred. Describe the operation in the case of the state transition.

**Usage**
```
#include "cia402appl.h"
```

**Parameters**
- `TCiA402Axis *pCiA402Axis`

**Return Value**
- 0: Normal end
- 1: Error

**Remark**
In the case of error occurrence during processing, exit the function by setting the appropriate values for each object in accordance with the CiA402 standard. If 1 is set to return value, state transition does not occur.

### APPL_MOTOR_MotionControl_Main

**Description**
Implement the motion control code when the state of CiA402 FSA is "Operation enabled". Describe the process for each mode of operation.

**Usage**
```
#include "cia402appl.h"
```

**Parameters**
- `TCiA402Axis *pCiA402Axis`

**Return Value**
- 0: Normal end
- 1: Error

**Remark**
At the initial state, this function is described in "main.c" and calls "CiA402_DummyMotionControl" function for reference.
Documents for Reference

User’s Manual: Hardware
- RX72M Group User’s Manual: Hardware (Document No. R01UH0804)
- Renesas Starter Kit+ for RX72M User’s Manual (Document No. R20UT4383)
- RX72M Group Communications Board Hardware Manual (Document No. R01AN4661)

Download the latest version from Renesas Electronics website.

Startup Manual
- RX72M Group RSK Board EtherCAT Startup Manual (Document No. R01AN4689)
- RX72M Group Communications Board EtherCAT Startup Manual (Document No. R01AN4672)

Download the latest version from Renesas Electronics website.

Application Note
- RX Family EtherCAT Module Firmware Integration Technology (Document No. R01AN4881)

Technical Updates/Technical News
- Download the latest version from Renesas Electronics website.

User’s Manual: Development Environment
- RX Family C/C++ Compiler, Assembler, Optimizing Linkage Editor Compiler Package (R20UT0570)

Download the latest version from Renesas Electronics website.

CiA402 Standards:
- IEC 61800-7-201 Edition 1.0
  Adjustable speed electrical power drive systems Part 7-201: Generic interface and use of profiles for power drive systems Profile type 1 specification

- IEC 61800-7-301 Edition 1.0
  Adjustable speed electrical power drive systems Part 7-301: Generic interface and use of profiles for power drive systems Mapping of profile type 1 to network technologies
## Revision History

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Page</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Aug. 31, 2020</td>
<td>—</td>
<td>First edition issued</td>
</tr>
</tbody>
</table>
General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostat Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity.

2. Processing at power-on
   The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the 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.

5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is 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. Additionally, 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.

6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between $V_{IL}$ (Max.) and $V_{IH}$ (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between $V_{IL}$ (Max.) and $V_{IH}$ (Min.).

7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of 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.

---

*Arm® and Cortex® are registered trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.
Ethernet is a registered trademark of Fuji Xerox Co., Ltd.
IEEE is a registered trademark of the Institute of Electrical and Electronics Engineers Inc.
TRON is an acronym for "The Real-time Operation system Nucleus".
ITRON is an acronym for "Industrial TRON".
μITRON is an acronym for "Micro Industrial TRON".
TRON, ITRON, and μITRON do not refer to any specific product or products.
EtherCAT® and TwinCAT® are registered trademarks and patented technologies, licensed by Beckhoff Automation GmbH, Germany.
Additionally all product names and service names in this document are trademarks or registered trademarks which belong to the respective owners.
Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.

2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.

3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.

4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.

5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   * “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   * “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.

6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.

7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product, a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.

8. Please contact a Renesas Electronics sales office for details as to environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.

10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.

11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.

12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

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:

www.renesas.com/contact/

© 2020 Renesas Electronics Corporation. All rights reserved.