SH7216 Group

Performance Evaluation Software

REJ06B0990-0101
Rev.1.01
Jun 25, 2010

Introduction

This application note presents a sample program that uses multi-function timer pulse unit 2 (MTU2) of the SH7216 to evaluate the performance of tasks (functions) created by the user.

Target Device

SH7216

Contents

1. Introduction 2

2. Configuration of Sample Program 3

3. Using the Sample Program 6

4. Operation of Sample Program 10

5. Reference Documents 13

1. Introduction

1.1 Specifications

The sample program uses multi-function timer pulse unit 2 (MTU2) of the SH7216 to evaluate the performance of tasks (functions) created by the user.

The sample program is designed to have user tasks embedded in it. It evaluates the performance of the tasks by counting the required number of cycles from start to end using multi-function timer pulse unit 2 (MTU2) and converting the result into a CPU clock cycle count.

Figure 1 shows an outline of the performance evaluation model.

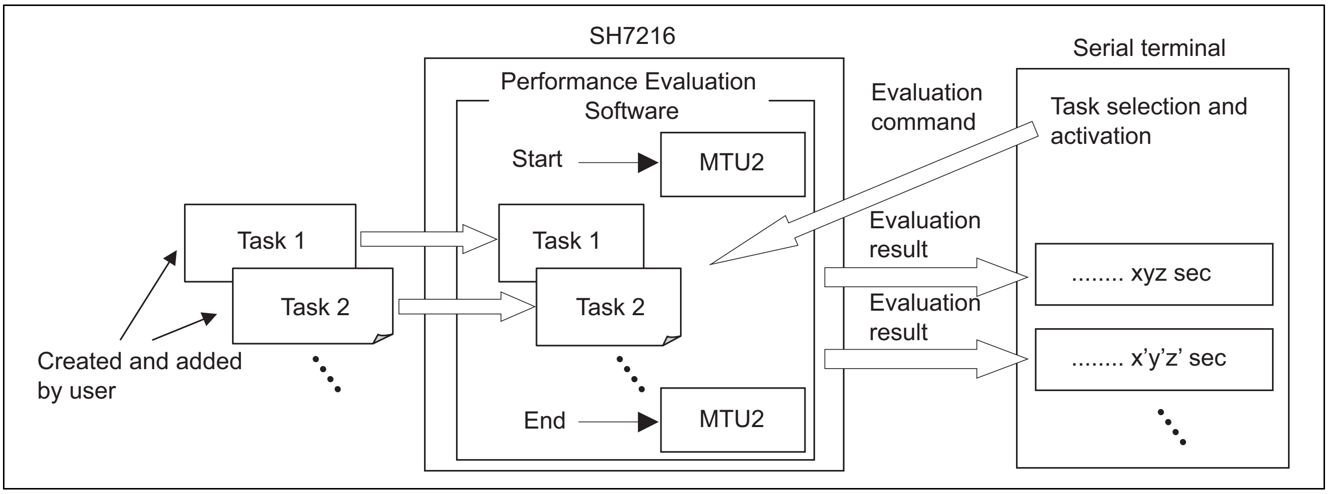

Figure 1 Performance Evaluation Outline

1.2 Functions Used

- Interrupt controller (INTC)
- Multi-function timer pulse unit 2 (MTU2)
- Serial communication interface (SCI)

1.3 Applicable Conditions

MCU	SH7216
Operating frequency	Internal clock: 200 MHz Bus clock: 50 MHz Peripheral clock: 50 MHz
Integrated development environment	Renesas Electronics High-performance Embedded Workshop, Ver. 4.07.00.007
C compiler	Renesas Electronics SuperH RISC Engine Family C/C++ Compiler Package, Ver. 9.03 Release 00
Compile options	-cpu=sh2afpu –debug –goptimize -gbr=auto -chgincpsth -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 -nologo

1.4 Related Application Notes

Fixed-point Library (Ver. 1.01): Compiler Use Guide

2. Configuration of Sample Program

The configuration of the sample program is described below.

2.1 System Configuration

Figure 2 shows the system configuration used for performance evaluation.

(1) PC used for development
A PC with HEW4 installed is used for software development based on the sample program.

(2) E10A emulator and E10A cable
A E10A emulator writes the software developed to the on-chip ROM of the SH7216.
A E10A cable is used to connect the PC used for development to the SH7216 CPU board.

(3) SH7216 CPU board (model: R0K572167C001BR)
A SH7216 MCU, E10A connector, SCI connector, external SDRAM, and other components are mounted on the SH7216 CPU board. Evaluation takes place according to evaluation commands received from the host PC. Then the evaluation result is transmitted to the host PC.

(4) SCI cable
The SCI cable is used to connect the SH7216 CPU board to the host PC, enabling transfer of evaluation commands and evaluation results.

(5) Host PC
The user inputs evaluation commands to a PC with a serial communication application such as Tera Term or HyperTerminal installed, and the evaluation commands are transferred to the SH7216 CPU board via the SCI cable. In addition, evaluation results are received from the SH7216 CPU board via the SCI cable and displayed on the host PC as notifications to the user.

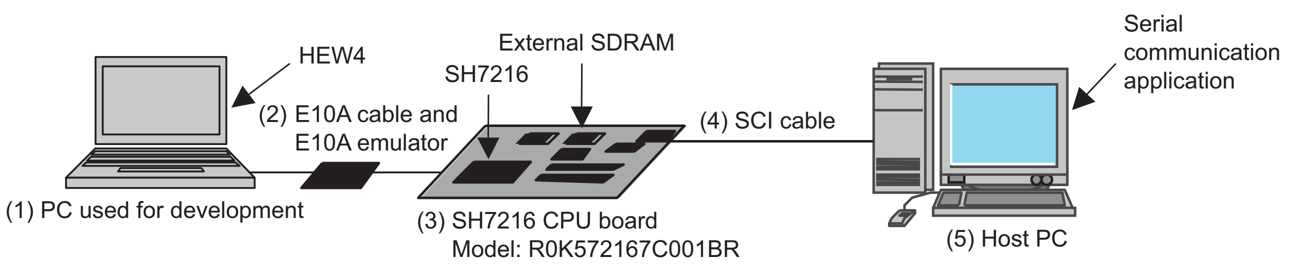

Figure 2 System Configuration

2.2 File Configuration

The sample program comprises routines for SH7216 module settings, serial I/O, and user task performance evaluation, as well as user tasks.

The SH7216 module settings routine makes settings for the CPU operating frequency (CPG), interrupt (INTC), timer (MTU2), etc.

The serial I/O routine starts performance evaluation at the command of the user, displays performance evaluation results for the user, etc.

The user task performance evaluation routine executes user tasks, counts the required number of cycles from start to end using a timer (MTU2), and converts the result into a CPU clock cycle count.

The user tasks are functions written in the C language. A function that processes a loop 1,000 times is provided as a sample.

Figure 3 shows the file configuration of the sample program.

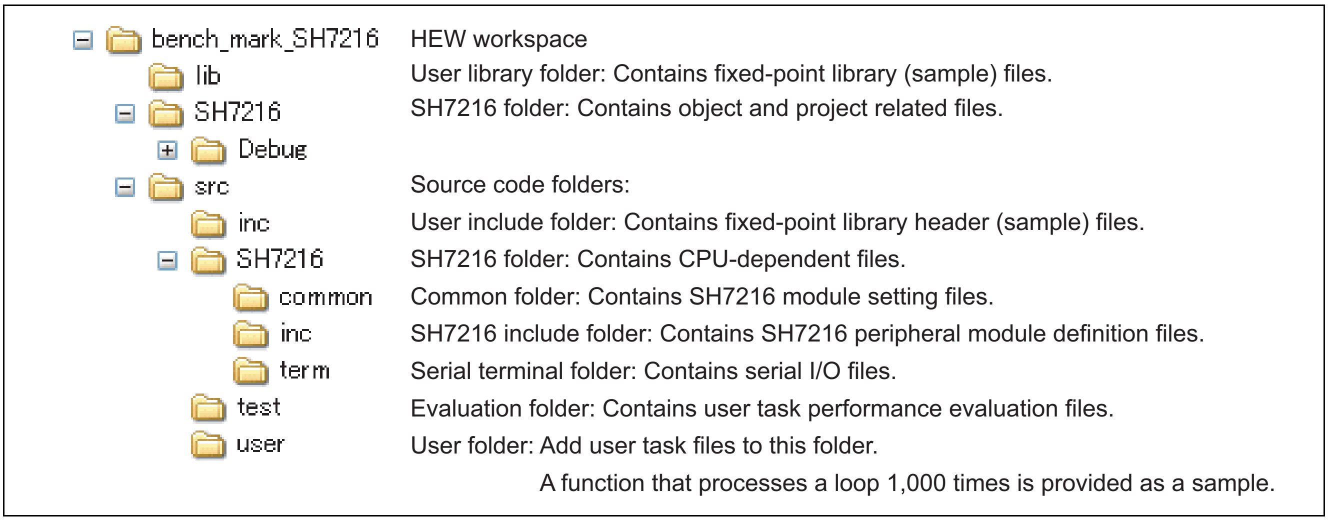

Figure 3 File Configuration

2.3 Section Settings

Table 1 lists the section settings for the sample program.

Table 1 Sample Program Section Settings

No.	Address	Section Name	Description
1	0x00000000	DVECTTBL	Vector table
2		DINTTBL	
3	0x00000800	PResetPRG	Reset handler
4		PIntPRG	Exception handler
5	0x00001000	P	Program area
6		PURAM	
7		C	Constant area
8		C$BSEC	
9		C$DSEC	
10		D	Initialization data area
11	0xFFF80000	B	Uninitialized data area
12		R	Initialization data area (allocated in RAM)
13		RPURAM	
14	0xFFF8BC00	S	Stack area

3. Using the Sample Program

How to use the sample program is described below, based on the system configuration shown in figure 2 and the file configuration shown in figure 3.

3.1 Loading and Running the Program

The procedure for loading and executing the program is described below.

(1) On the PC used for development, use HEW4 to compile and link the source code.

(2) Connect the PC used for development to the E10A emulator using the USB cable.

(3) Connect the E10A emulator to the SH7216 CPU board using the E10A cable.

(4) Power on the SH7216 CPU board.

(5) Launch the file **bench_mark.hws** in the **bench_mark_SH7216** folder.

(6) From the HEW4 menu, select **Debug** > **Connect**.

(7) A dialog box for selecting the emulator mode appears. As shown in figure 4, select **R5F72167AD** for **Device** and **E10A-USB Emulator** for **Mode**, then click **OK**.

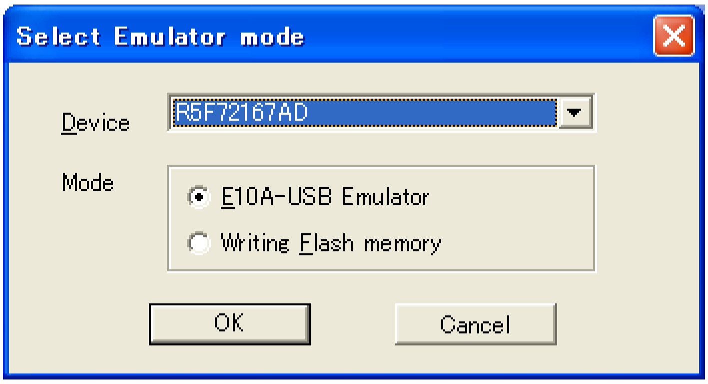

Figure 4 Emulator Mode Selection Dialog Box

(8) A message appears requesting a reset. Press the reset switch on the SH7216 CPU board, then click **OK**.

Figure 5 Reset Request

(9) A dialog box for inputting the operating frequency appears. Enter **12.50MHz** for **Clock**, then click **OK**.

(10) A dialog box for inputting the ID code appears. As shown in figure 6, enter **E10A** and check the box next to **New ID code**, then click **OK**.

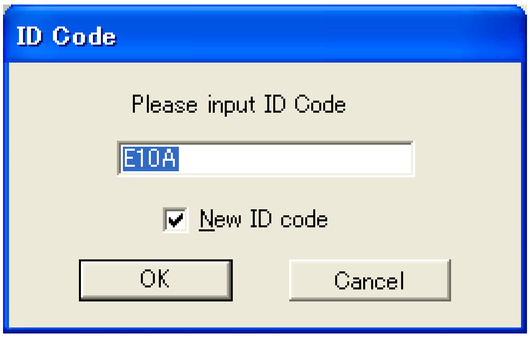

Figure 6 ID Code Request

(11) From the menu, select **Build** > **All**. When the build completes, a confirmation dialog box appears. Click **OK** to download SH7216**.abs**.

(12) From the menu, select **Debug** > **Execute after Reset** to run the program.

3.2 Performance Evaluation

The performance evaluation procedure[1] is described below.

(1) Connect the host PC to the SH7216 CPU board using the SCI cable. If the PC does not have an SCI port, use a USB-SCI conversion cable.

(2) Launch a serial communication application on the host PC and set the communication options as follows: Transfer rate: 9,600 bps, flow control: Xon/Xoff

(3) Run the program as described in 3.1 above.

(4) When the program starts, the following message is displayed in the serial communication application.

```
===============================================================
SH7216 Evaluation Program. Ver.1.00.00
COPYRIGHT (C) 2010 Renesas Electronics Corporation. ALL rights reserved
and Renesas Solutions Corporation. ALL rights reserved
===============================================================
>
```

(5) Enter **HELP** from the console of the serial communication application to display the available evaluation commands. The example below shows the display when only the elementary function calculation and the sample user task that processes a loop 1,000 times (**user_func1**), both of which are included in the sample program, have been embedded. Other tasks for evaluation and corresponding commands may be added as necessary.

```
> HELP
commands help
MATH : math function test
FUNC1 : user task(=func1) test
>
```

(6) Enter **MATH** from the console of the serial communication application to perform elementary function calculations using the SH-2A embedded libraries (**math.h**, **mathf.h**) and the fixed-point library included in the sample program. After the calculations complete, the number of cycles required is displayed. The display below shows the average values obtained when the SH-2A embedded libraries and the fixed-point library were used to perform calculations, each of which were repeated 256 times.

```
> MATH
============ Start cosine calculation ===========================   Cosine calculation
Evaluation Result: Number Of Data = 256   <---- Number of repetitions

Renesas SH Fixed Point Calculation Cycles (256 point average) = 143   <---- Fixed-point library calculation

Renesas SH Double Calculation Cycles (256 point average) = 396   <---- SH-2A embedded library, double-precision

Renesas SH Float Calculation Cycles (256 point average) = 148   <---- SH-2A embedded library, single-precision
============ End cosine calculation =============================

============ Start square root calculation ========================   Square root calculation
Evaluation Result: Number Of Data = 256   <---- Number of repetitions

Renesas SH Fixed Point Calculation Cycles (256 point average) = 143   <---- Fixed-point library calculation

Renesas SH Double Calculation Cycles (256 point average) = 64   <---- SH-2A embedded library, double-precision

Renesas SH Float Calculation Cycles (256 point average) = 48   <---- SH-2A embedded library, single-precision
============ End square root calculation ========================
>
```

(7) Enter **FUNC1** from the console of the serial communication application to process a loop 1,000 times, which is included in the sample program as a sample user task. After the task completes, the number of cycles required for processing is displayed. The display below shows the average value obtained when the function that processes a loop 1,000 times was run 10 times in succession.

```
> FUNC1
======== Start user task(=func1) operation =====================
Evaluation Result: Number Of Evaluation Times = 10

User Task(=func1) Operation Cycles (10 times average) = 1044
============ End user task(=func1) operation =====================
>
```

1. Note on Evaluation

The sample program uses a timer to measure the time required for tasks to complete, but the processing by the CPU to start and stop the timer also uses several dozen cycles. This means it is necessary to measure the number of cycles used when no task is executed and subtract that number from the number of cycles used when a task is executed in order to obtain accurate results.

For example, in the case of the elementary function calculation included in the sample program, the number of cycles used when no task is executed can be measured by enabling **EMPTY_LOOP** in **test_math.c** (by commenting out **#undef EMPTY_LOOP**).

```
/* ---- empty loop operation ---- */
/* operation result is subtract from empty loop operation value */
#define EMPTY_LOOP
//#undef EMPTY_LOOP // if comment out, execute empty loop operation.
```

3.3 Adding User Tasks

The procedure for adding user tasks is described below.

(1) Create the task to be added as a function written in C (**user_func2** in the description below).

(2) Embed **user_func2** in **test_user.c**. To do this, execute the **start_eval** function before executing **user_func2** and the **execute stop_eval** function afterward, as with **user_func1** in **test_user.c**.

```
for(i = 0;i < EVAL_TIMES;i++){
        start_eval();        ← Execute start_eval() before executing user_func2.
        #ifndef EMPTY_LOOP
        user_func2();
        #endif
        stop_eval();         ← Execute stop_eval() after executing user_func2.
        cyc_func1 += get_eval_cycle();        ← Get number of cycles used for processing.
}
```

4. Operation of Sample Program

4.1 Overall Sequence

Figure 7 shows the overall sequence of the sample program. The sample program includes functions for elementary function calculation evaluation (**test_math** function), user task evaluation (**test_user_func1** function), and evaluation command display (**help** function). Additional user tasks may be added, as needed, by following the procedure described in 3.3.

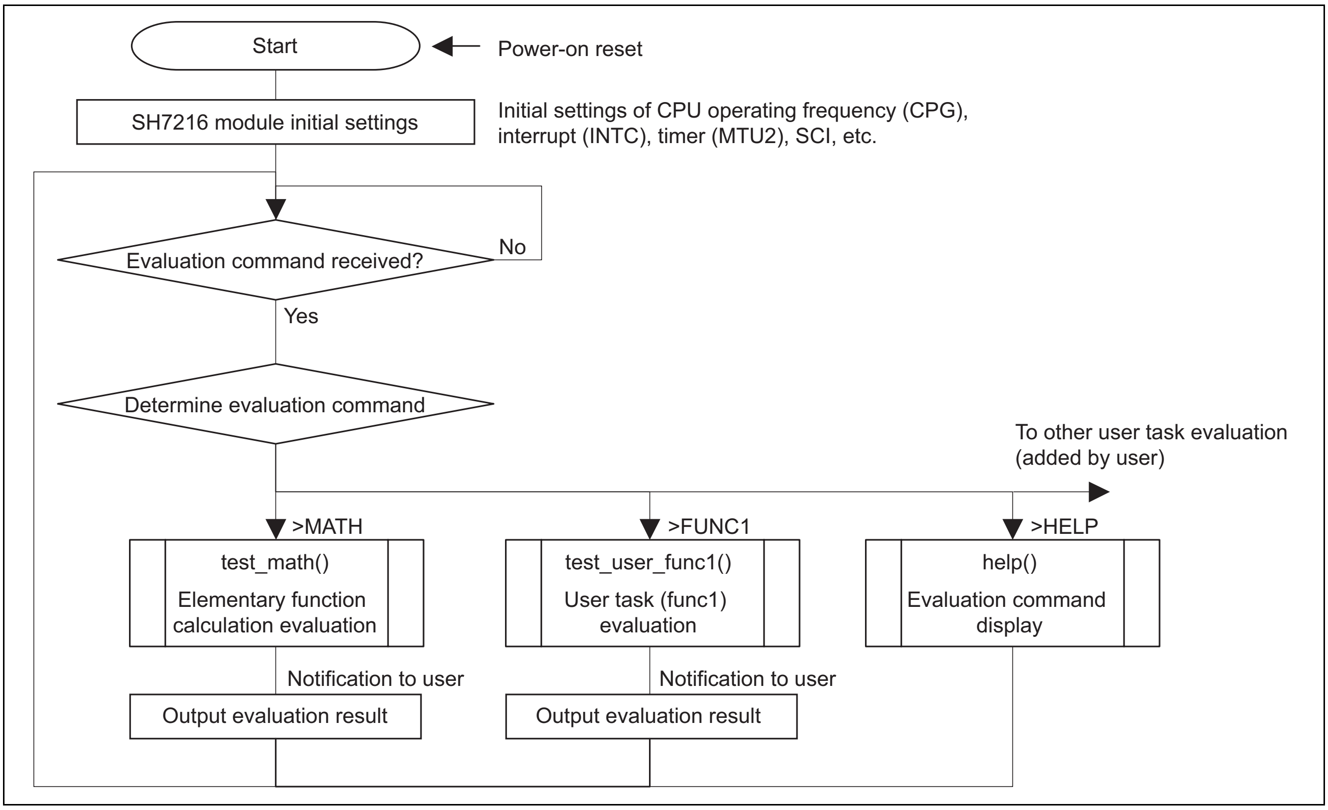

Figure 7 Overall Sequence of Sample Program

4.2 Elementary Function Calculation Evaluation

Figure 8 shows the processing sequence of the elementary function calculation evaluation function.

Entering **MATH** from the console of the serial communication application starts the elementary function calculation evaluation function.

A timer (CH2 of MTU2) is used for measurement, and when an overflow occurs the overflow count is incremented by an overflow interrupt (TGIA_2 of MTU2_2).

Since the MTU2 operates on the peripheral clock, the MTU2 cycle count is multiplied by four (peripheral clock / internal clock) and output as the evaluation result.

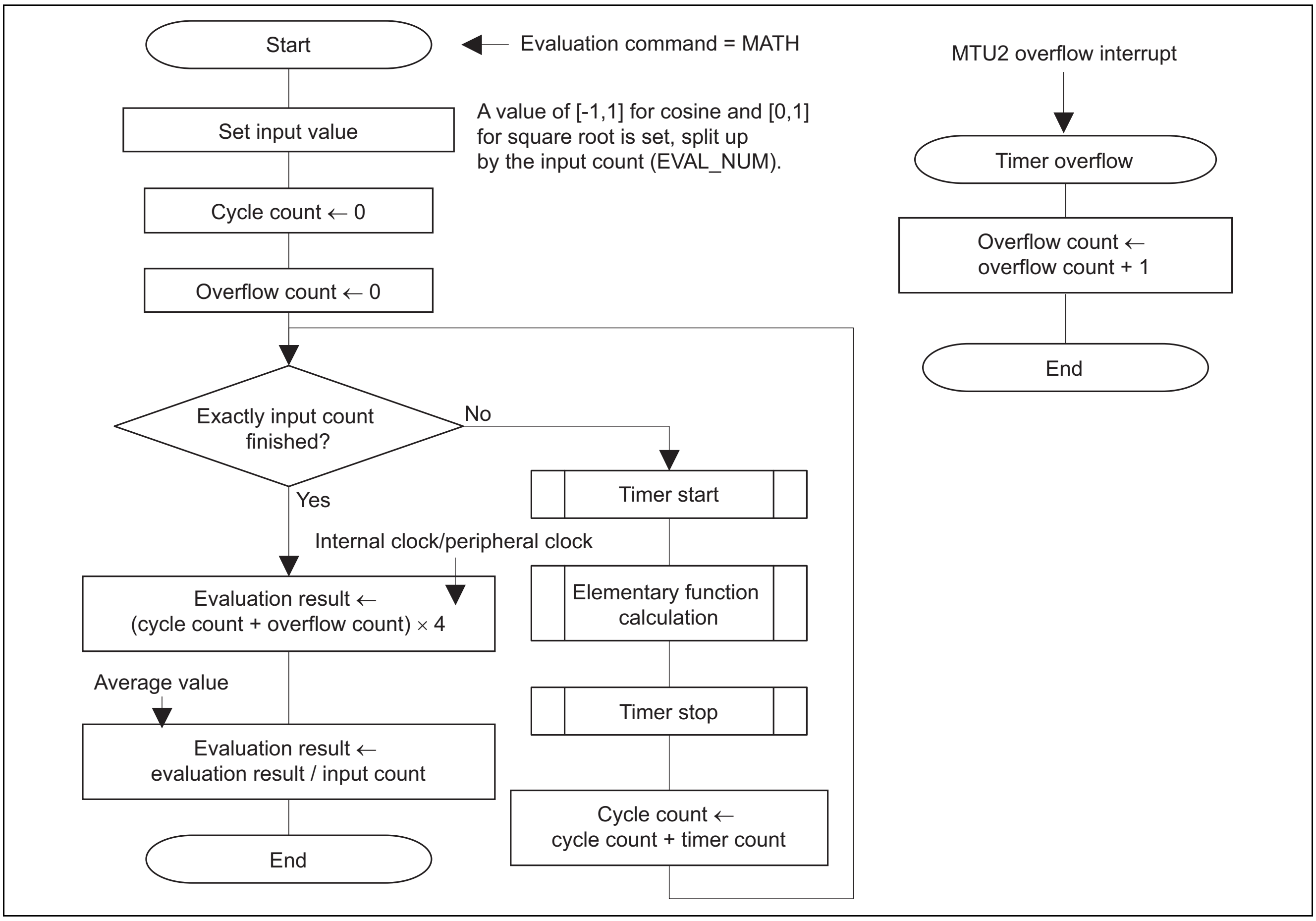

Figure 8 Elementary Function Calculation Evaluation Processing Sequence

4.3 User Task Evaluation

Figure 9 shows the processing sequence of the user task (func1) evaluation function.

Entering **FUNC1** from the console of the serial communication application starts the user task evaluation function.

A timer (CH2 of MTU2) is used for measurement, and when an overflow occurs the overflow count is incremented by an overflow interrupt (TGIA_2 of MTU2_2).

Since the MTU2 operates on the peripheral clock, the MTU2 cycle count is multiplied by four (peripheral clock / internal clock) and output as the evaluation result.

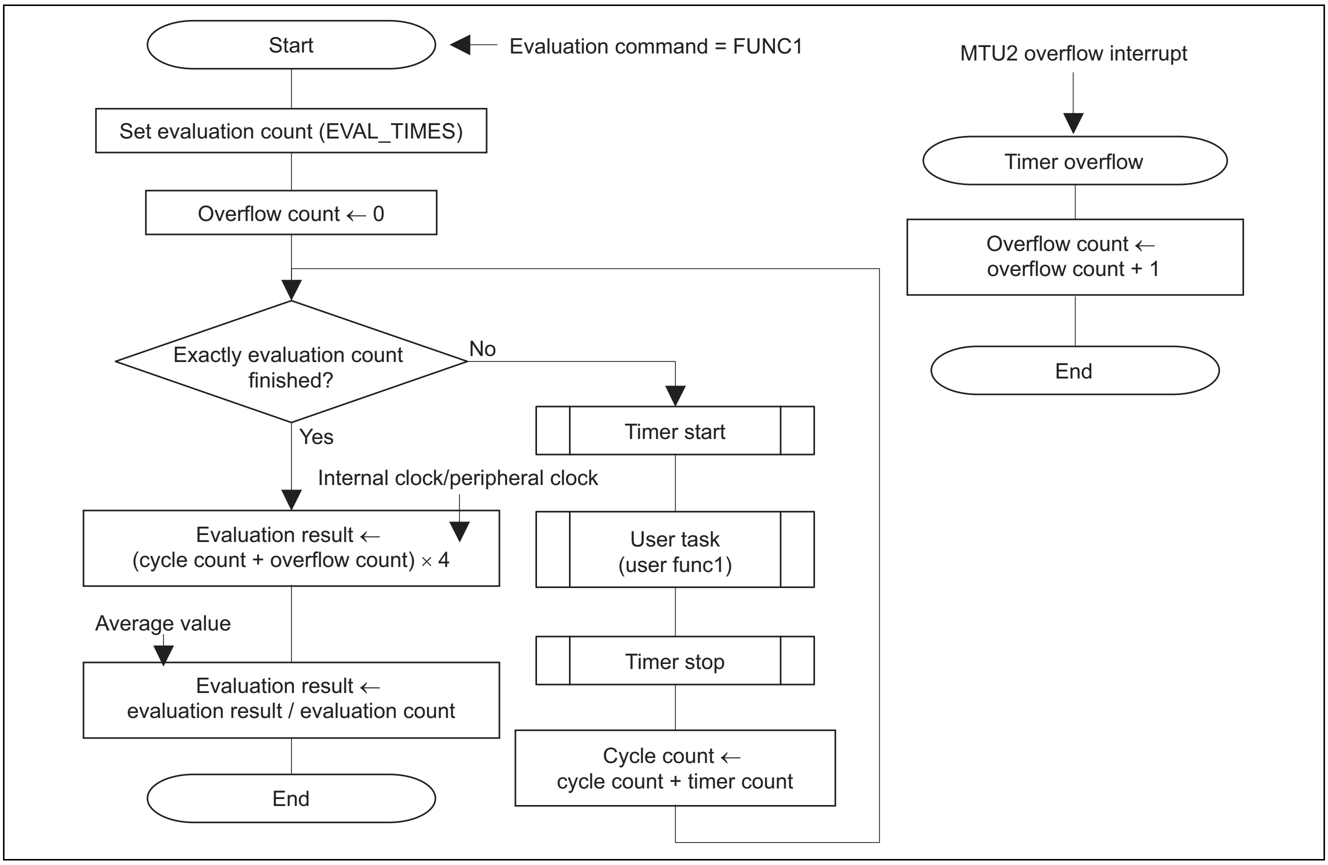

Figure 9 User Task Evaluation Processing Sequence

4.4 Evaluation Command Display

Figure 10 shows the processing sequence of the evaluation command display function.

Entering **HELP** from the console of the serial communication application displays the evaluation commands.

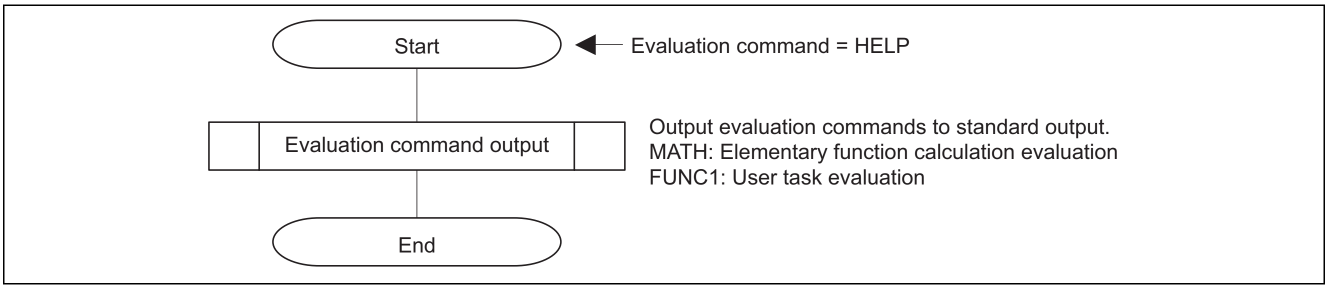

Figure 10 Evaluation Command Display Processing Sequence

5. Reference Documents

- Hardware Manual
  SH7216 Group Hardware Manual, Rev. 1.01
  (The latest version can be downloaded from the Renesas Electronics Web site.)
- SH716 CPU Board User’s Manual
  SH7216 CPU Board R0K572167C001BR User’s Manual, Rev. 0.03

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	May.18.10	—	First edition issued
1.01	Jun.25.10	—	Modifications to source project due to change in FRQCR setting method
		5	Table 1 Sample Program Section Settings amended

General Precautions in the Handling of MPU/MCU Products

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

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.
   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.
   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.
5. Differences between Products
   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

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

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

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

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

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

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

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

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

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

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

Colophon 1.0