RZ/T1 Group

Performance Monitor Sample Program

R01AN4100EJ0110
Rev.1.10
Jun. 07, 2018

Introduction

This application note explains how to set up the performance monitor and cache operation that operate on a Cortex-R4 mounted on an RZ/T1 Group microcomputer.

Target Devices

RZ/T1

When applying the sample program covered in this application note to another microcomputer, modify the program according to the specifications for the target microcomputer and conduct an extensive evaluation of the modified program.

Table of Contents

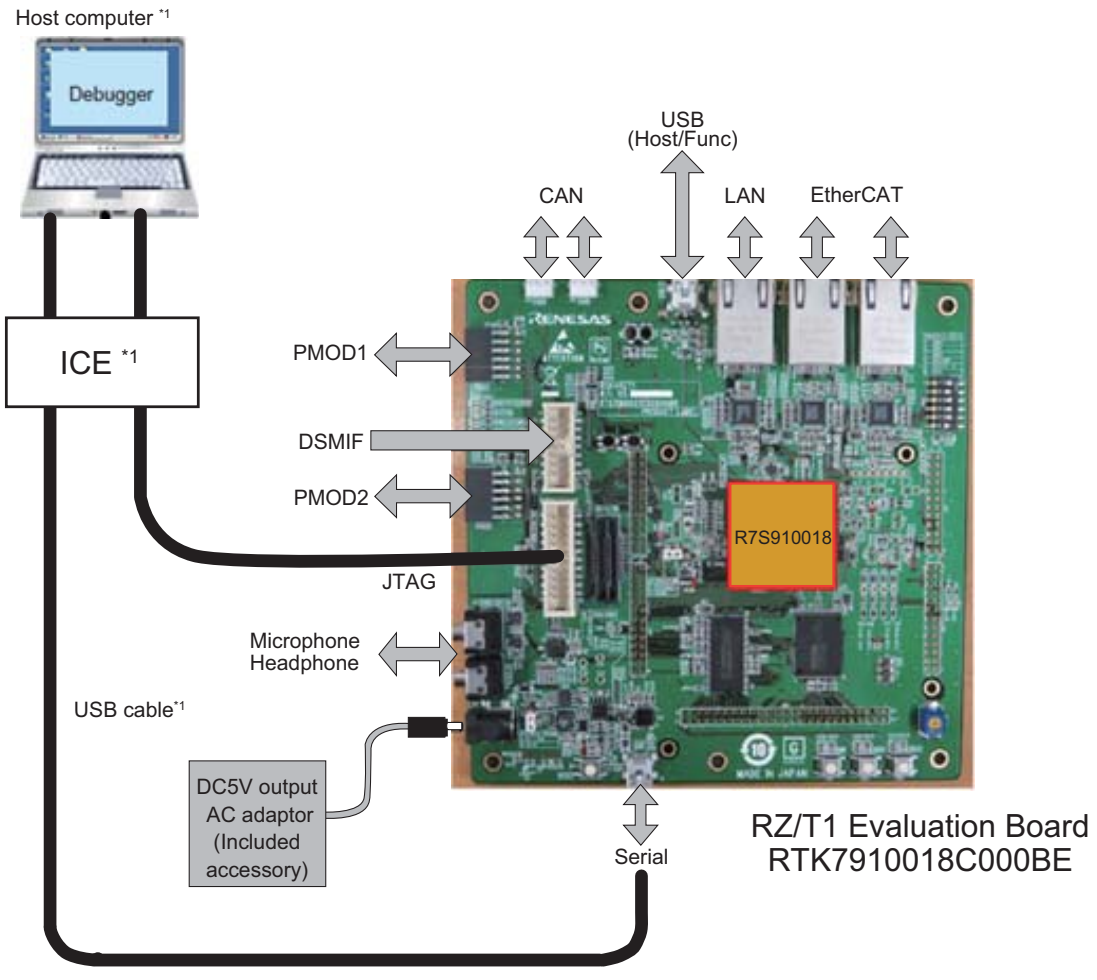
1. Specifications

A Cortex-R4 mounted on an RZ/T1 microcomputer has a performance monitor that provides three event count registers and one cycle count register. Each event counter operates independently, and can count the different target event that is assigned to that counter. In addition, the Cortex-R4 of an RZ/T1 includes the instruction cache of 8 Kbytes and the data cache of 8Kbytes. These caches adopt a line size of 32 bytes. The Invalidate, Clean, and Clean & Invalidate operations can be performed on a line size basis for each cache.

"Table 1.1 Major Peripheral Functions and Applications" and "Figure 1.1 Operating Environment" show the operating environment when the sample code is being executed.

Table 1.1 Major Peripheral Functions and Applications

Peripheral Function	Application
Clock pulse generator (CPG)	The CPG produces the CPU clock and low-speed on-chip oscillator clock signals
FIFO integrated serial communication interface (SCIFA)	Asynchronous communications of the SCIFA is used for COM port communications by using an RS-232C interface
Bus state controller (BSC)	Used to set the SDRAM that is used for the sample program.
Compare match timer W (CMTW unit 0)	Used to compare to the cycle count of the performance monitor.



Note 1. Indicates the device that the user needs to prepare.

Figure 1.1 Operating Environment

2. Operating Environment

The sample code covered in this application note is for the environment below.

Table 2.1 Operating Environment

Item	Description
Microcomputer	RZ/T1 Group
Operating frequency	CPU clock (CPUCLK): 450 MHz
Operating voltage	Power supply voltage (I/O): 3.3 V
Integrated development environment	Manufactured by IAR systems Embedded Workbench® for Arm Version 8.20.2 Manufactured by Arm DS-5™ 5.26.2 Manufactured by RENESAS e2studio 6.1.0
Operating mode	SPI boot mode (serial flash memory) 16-bit bus boot mode (NOR flash memory)
Communication settings of the terminal software	• Communication speed: 115,200 bps • Data length: 8 bits • Parity: N/A • Stop bit length: 1 bit • Flow control: N/A • Line break code (reception): CR • Line break code (transmission): CR
Board used	• RZ/T1 Evaluation Board (RTK7910018C00000BE)
Devices used (functions to be used on the board)	• Serial interface (USB-Mini B connector J8) • NOR flash memory (connected to CS0/CS1 space) Manufacturer: Macronix International Co., Ltd. Model: MX29GL512FLT2I-10Q • Serial flash memory Manufacturer: Macronix International Co., Ltd. Model: MX25L51245G • SDRAM (connected to CS2/CS3 space) Manufacturer: ISSI Model: S42S16320D
USB serial port driver for PC	• For RTK7910018C00000BE • For RTK7910022C00000BR

3. Related Application Note

The application notes related to this application note are listed below for reference.

- Application Note: RZ/T1 Group Initial Settings (R01AN2554EJ)
- Application Note: RZ/T1 Group FIFO Integrated Serial Communication Interface (SCIFA) (R01AN2577EJ)
- Application Note: RZ/T1 Group CMTW & ELC Sample Program (R01AN2600EJ)

Note: Registers not mentioned in this application note should be used at a value set in the Application Note: RZ/T1 Group Initial Settings.

4. Peripheral Functions

For the basics of the operating mode, clock pulse generator (CPG), FIFO integrated serial communication interface (SCIFA), compare match timer W (CMTW), resetting, and general-purpose I/O ports, refer to the RZ/T1 Group User's Manual: Hardware.

For the basics of the performance monitor and cache operation, refer to the Cortex-R4 Technical Reference Manual of Arm.

5. Hardware

5.1 Pins

Table 5.1 lists pins to be used and their functions.

Table 5.1 Pins and Functions

Pin Name	I/O	Function
MD0	Input	Selection of operating modes MD0 = L, MD1 = L, MD2 = L (SPI boot mode) MD0 = L, MD1 = H, MD2 = L (16-bit bus boot mode)
MD1	Input	
MD2	Input	

6. Software

This sample program includes the driver for using the performance monitor and cache operation, and the utility function that converts the cycle count to the actual time.

6.1 Function Outline

This software includes a function that uses the performance monitor to count the number of cycles, the number of times control is returned from the procedure, and the number of executions of a data read instruction. Simultaneously with execution of the function, this software also uses CMTW to count the number of cycles. This software converts the number of cycles obtained using the performance monitor and the number of cycles obtained using CMTW to their actual processing times, and then outputs the times by using the serial output function (described later).

In addition, this software reads from data from the SDRAM, and then uses the serial output function (described later) to output the status of memory after execution of each cache operation.

As the execution result of each function, this software uses the asynchronous communications of the FIFO integrated serial communication interface (SCIFA) to perform COM port communications with the host PC via an RS-232 interface, and then outputs each sample program execution result by using the terminal software on the host PC.

6.2 Interrupts

This sample program uses no interrupts.

6.3 Fixed-Width Integers

Table 6.1 lists the fixed-width integers for the sample program. The fixed-width integers used in the sample code are defined in the standard library.

Table 6.1 Fixed-Width Integers for Sample Program

Symbol	Description
int8_t	8-bit signed integer
int16_t	16-bit signed integer
int32_t	32-bit signed integer
int64_t	64-bit signed integer
uint8_t	8-bit unsigned integer
uint16_t	16-bit unsigned integer
uint32_t	32-bit unsigned integer
uint64_t	64-bit unsigned integer

6.4 Structures/Unions

This sample program uses no structures/unions.

6.5 Constants

Table 6.2 and Table 6.3 list the constants to be used in the sample programs.

Table 6.2 Constants to be Used in the Performance Monitor Sample Program

Constant Name	Setting Value	Description
PMON_EVTCNT0 to PMON_EVTCNT2	(0u) to (1U)	Selecting the performance monitor number to be set
PMON_EVT_SOFTINC	(0x00)	Count event option: Software increment
PMON_EVT_ICMISS	(0x01)	Count event option: Instruction cache miss
PMON_EVT_DCMISS	(0x03)	Count event option: Data cache miss
PMON_EVT_DCACC	(0x04)	Count event option: Data cache access
PMON_EVT_DRDEX	(0x06)	Count event option: Data read executed
PMON_EVT_DWDEX	(0x07)	Count event option: Data write executed
PMON_EVT_INSTEX	(0x08)	Count event option: Instruction architecturally executed
PMON_EVT_DUALEX	(0x5E)	Count event option: Dual-issued pair of instructions architecturally executed
PMON_EVT_EXCENT	(0x09)	Count event option: Exception taken
PMON_EVT_EXCRET	(0x0A)	Count event option: Exception return architecturally executed
PMON_EVT_CHCON	(0x0B)	Count event option: Change to context ID executed
PMON_EVT_CHPC	(0x0C)	Count event option: Software change of PC architecturally executed, other than exception returns
PMON_EVT_BIMM	(0x0D)	Count event option: B immediate, BL immediate, or BLX immediate instruction architecturally executed. An event occurs even when processing does not actually branch.
PMON_EVT_PROCRET	(0x0E)	Count event option: Procedure return architecturally executed, other than exception returns
PMON_EVT_UNALIGN	(0x0F)	Count event option: Unaligned access architecturally executed
PMON_EVT_BMISSPRE	(0x10)	Count event option: Branch mispredicted or not predicted
PMON_EVT_CYCLECNT	(0x11)	Count event option: Cycle count
CPUCKSEL_150M	(0x00u)	CPU frequency setting: 150 MHz
CPUCKSEL_300M	(0x01u)	CPU frequency setting: 300 MHz
CPUCKSEL_450M	(0x02u)	CPU frequency setting: 450 MHz
CPUCKSEL_600M	(0x03u)	CPU frequency setting: 600 MHz
TEST_TIME	(3u)	Execution number of read data evaluations in the performance monitor sample program
TST_MEM_ADDR	(0x68000000)	Data read destination address (SDRAM) in the sample programs for the performance monitor and cache operation
TST_MEM_CNT	(1000000)	Execution number of read data instructions in the performance monitor sample program

Table 6.3 Constants To be Used in the Cache Operation Sample Program

Constant Name	Setting Value	Description
CACHE_MAX_WAY	(0x3u)	Maximum number of ways of the cache
CACHE_WAY_SHIFT	(30u)	Bit shift setting: [31:30]; used to specify the number of ways
CACHE_MAX_SET	(0x3Fu)	Maximum number of sets of the cache
CACHE_SET_SHIFT	(5u)	Bit shift setting: [10:5]; used to specify the number of sets
CACHE_LINE_MASK	(0xFFFFFFE0u)	Cache operation target address to be masked
CACHE_LINE_SIZE	(0x00000020u)	Cache line size
TST_MEM_SIZE	(0x00000060u)	Size of operation-target memory in the cache operation sample program (3 cache lines)
DUMP_ALIGN_MASK	(0xFFFFFFF0u)	Aligned address to be masked
DUMP_INCR_SIZE	(0x00000010u)	Number of output bytes in a dump when a memory dump is taken in the cache operation sample program

6.6 Variables

This sample program uses no static variables.

6.7 Functions

Table 6.4 lists the functions common to the sample programs. Table 6.5 lists the functions used in the performance monitor sample program. Table 6.6 lists the functions used in the cache operation sample program.

Table 6.4 List of Common Functions

Function	Description
main	Main function of the sample program

Table 6.5 List of Functions Used in the Performance Monitor Sample Program

Function Name	Description
R_PMON_Open	Initial setting function for the performance monitor
R_PMON_Close	End setting function for the performance monitor
R_PMON_Start	Event count start function
R_PMON_Stop	Event count stop function
R_PMON_StopAll	Function that stops all counters
R_PMON_SelectCount	Counter selection function
R_PMON_GetCurCount	Count obtain function
sample_pmon	Main function of the performance monitor sample program
exec_tst_mem_dword_read	Evaluation function of the read instruction execution
cyclecnt_to_nanosec	Function that converts the cycle count to the actual time

Table 6.6 List of Functions Used in the Cache Operation Sample Program

Function	Description
R_CACHE_Inval_I_All	Function performing the Invalidate operation for all instruction caches
R_CACHE_Inval_I	Function performing the Invalidate operation for an instruction cache on a cache line basis
R_CACHE_Clean_D_All	Function performing the Clean operation for all data caches
R_CACHE_Inval_D_All	Function performing the Invalidate operation for all data caches
R_CACHE_CleanInval_D_All	Function performing the Clean & Invalidate operation for all data caches
R_CACHE_Clean_D	Function performing the Clean operation for a data cache on a cache line basis
R_CACHE_Inval_D	Function performing the Invalidate operation for a data cache on a cache line basis
R_CACHE_CleanInval_D	Function performing the Clean & Invalidate operation for a data cache on a cache line basis
sample_cache	Main function of the cache operation sample program
write_incr_data	Function for writing evaluation data
dump_memory	Memory dump function

6.8 Specifications of Functions

The following shows the function specifications of the sample code.

6.8.1 Common Functions

Function name: main	
Synopsis	Main function of the sample program
Declaration	int main (void)
Description	This function is the main processing of the sample program. This function initializes the board, and then calls sample_pmon() and sample_cache() to execute the sample program.
Arguments	None
Return value	0
Supplement	None

6.8.2 Functions Used in the Performance Monitor Sample Program

Function name: R_PMON_Open	
Synopsis	Initial setting function for the performance monitor
Declaration	void R_PMON_Open(void);
Description	This function performs the settings required to start the counters of the performance monitor. • Permitting access to the performance monitor in user mode • Resetting the counters • Enabling the counters
Arguments	None
Return value	None
Supplement	None

Function name: R_PMON_Close	
Synopsis	End setting function for the performance monitor
Declaration	void R_PMON_Close(void);
Description	This function performs the settings required to stop the counters of the performance monitor. • Prohibiting access to the performance monitor in user mode • Disabling the counters
Arguments	None
Return value	None
Supplement	None

Function name: R_PMON_Start		
Synopsis	Event count start function	
Declaration	void R_PMON_Start (int32_t evtch, int32_t event, uint32_t ini_cnt);	
Description	This function starts an event counter.	
Arguments	int32_t evtch	Counter number selected for control target Select one from PMON_EVTCNT0 to PMON_EVTCNT2.
	int32_t event	Count-target event Select one of the constants that are indicated as “Count event option” in Table 6.2.
	uint32_t ini_cnt	Initial counter value
Return value	None	
Supplement	The behavior is undefined if an invalid value is specified for the argument “evtch” or “event”.	

Function name: R_PMON_Stop		
Synopsis	Event count stop function	
Declaration	void R_PMON_Stop(int32_t evtch);	
Description	This function stops an event counter.	
Arguments	int32_t evtch	Counter number selected for control target Select one from PMON_EVTCNT0 to PMON_EVTCNT2.
Return value	None	
Supplement	The behavior is undefined if an invalid value is specified for the argument “evtch”.	

Function name: R_PMON_StopAll	
Synopsis	Function that stops all counters
Declaration	void R_PMON_StopAll(void);
Description	This function stops counting events with all counters.
Arguments	None
Return value	None
Supplement	None

Function name: R_PMON_SelectCount		
Synopsis	Counter selection function	
Declaration	void R_PMON_SelectCount(int32_t evtch);	
Description	This function specifies the current counter number. This function is used to specify the counter to be controlled.	
Arguments	int32_t evtch	Counter number selected for control target Select one from PMON_EVTCNT0 to PMON_EVTCNT2.
Return value	None	
Supplement	The behavior is undefined if an invalid value is specified for the argument “evtch”.	

Function name: R_PMON_GetCurCount		
Synopsis	Count obtain function	
Declaration	void R_PMON_GetCurCount(void);	
Description	This function obtains the current count value.	
Arguments	int32_t evtch	Counter number selected for control target Select one from PMON_EVTCNT0 to PMON_EVTCNT2.
Return value	Counter value of the event counter	
Supplement	The counter value is obtained from the counter that has been selected with R_PMON_SelectCount. If R_PMON_SelectCount has not been executed, the behavior of this function is undefined.	

Function name: sample_pmon	
Synopsis	Main function of the performance monitor sample program
Declaration	void sample_pmon (void);
Description	This is the main function of the performance monitor sample program. This function executes required operations, and outputs the execution results to the serial terminal. For details, see section 6.8.2, Functions Used in the Performance Monitor Sample Program.
Arguments	None
Return value	None
Supplement	None

Function name: exec_tst_mem_dword_read	
Synopsis	Evaluation function of the read instruction execution
Declaration	void exec_tst_mem_dword_read (void);
Description	This function executes read instructions one million times to obtain the execution number of read instructions.
Arguments	None
Return value	None
Supplement	None

Function name: cyclecnt_to_nanosec		
Synopsis	Function that converts the cycle count to the actual time	
Declaration	void cyclecnt_to_nanosec (uint32_t cycle_cnt);	
Description	This function converts the cycle count value of the performance monitor to a time in nanoseconds.	
Arguments	uint32_t cycle_cnt	Cycle count value of the performance monitor
Return value	None	
Supplement	None	

6.8.3 Functions Used in the Cache Operation Sample Program

Function name: R_CACHE_Inval_I_All	
Synopsis	Function performing the Invalidate operation for all instruction caches
Declaration	void R_CACHE_Inval_I_All (void);
Description	This function performs the Invalidate operation for all instruction caches.
Arguments	None
Return value	None
Supplement	None

Function name: R_CACHE_Inval_I		
Synopsis	Function performing the Invalidate operation for an instruction cache on a cache line basis	
Declaration	void R_CACHE_Inval_I (uint32_t sta_addr, uint32_t size);	
Description	This function performs the Invalidate operation for an instruction cache on a cache line basis.	
Arguments	uint32_t sta_addr	Cache address where the Invalidate operation starts
	uint32_t size	Cache size of the Invalidate operation
Return value	None	
Supplement	If the address specified for the argument "sta_addr" is not the start address of a cache line boundary, the Invalidate operation starts with the cache line that includes the specified address. If the value specified for the argument "size" is not a multiple of the cache line size, the Invalidate operation is performed for the range up to the nearest multiple of the cache line size.	

Function name: R_CACHE_Clean_D_All	
Synopsis	Function performing the Clean operation for all data caches
Declaration	void R_CACHE_Clean_D_All (void);
Description	This function performs the Clean operation for all data caches.
Arguments	None
Return value	None
Supplement	None

Function name: R_CACHE_Inval_D_All	
Synopsis	Function performing the Invalidate operation for all data caches
Declaration	void R_CACHE_Inval_D_All (void);
Description	This function performs the Invalidate operation for all data caches.
Arguments	None
Return value	None
Supplement	None

Function name: R_CACHE_CleanInval_D_All	
Synopsis	Function performing the Clean & Invalidate operation for all data caches
Declaration	void R_CACHE_CleanInval_D_All (void);
Description	This function performs the Clean & Invalidate operation for all data caches.
Arguments	None
Return value	None
Supplement	None

Function name: R_CACHE_Clean_D		
Synopsis	Function performing the Clean operation for a data cache on a cache line basis	
Declaration	void R_CACHE_Clean_D (uint32_t sta_addr, uint32_t size);	
Description	This function performs the Clean operation for a data cache on a cache line basis.	
Arguments	uint32_t sta_addr	Cache address where the Clean operation starts
	uint32_t size	Cache size of the Clean operation
Return value	None	
Supplement	If the address specified for the argument "sta_addr" is not the start address of a cache line boundary, the Clean operation starts with the cache line that includes the specified address. If the value specified for the argument "size" is not a multiple of the cache line size, the Clean operation is performed for the range up to the nearest multiple of the cache line size.	

Function name: R_CACHE_Inval_D		
Synopsis	Function performing the Invalidate operation for a data cache on a cache line basis	
Declaration	void R_CACHE_Inval_D (uint32_t sta_addr, uint32_t size);	
Description	This function performs the Invalidate operation for a data cache on a cache line basis.	
Arguments	uint32_t sta_addr	Cache address where the Invalidate operation starts
	uint32_t size	Cache size of the Invalidate operation
Return value	None	
Supplement	If the address specified for the argument “sta_addr” is not the start address of a cache line boundary, the Invalidate operation starts with the cache line that includes the specified address. If the value specified for the argument “size” is not a multiple of the cache line size, the Invalidate operation is performed for the range up to nearest multiple of the cache line size.	

Function name: R_CACHE_CleanInval_D		
Synopsis	Function performing the Clean & Invalidate operation for a data cache on a cache line basis	
Declaration	void R_CACHE_CleanInval_D_All (uint32_t sta_addr, uint32_t size);	
Description	This function performs the Clean & Invalidate operation for a data cache on a cache line basis.	
Arguments	uint32_t sta_addr	Cache address where the Clean & Invalidate operation starts
	uint32_t size	Cache size of the Clean & Invalidate operation
Return value	None	
Supplement	If the address specified for the argument "sta_addr" is not the start address of a cache line boundary, the Clean & Invalidate operation starts with the cache line that includes the specified address. If the value specified for the argument "size" is not a multiple of the cache line size, the Clean & Invalidate operation is performed for the range up to the nearest multiple of the cache line size.	

Function name: sample_cache	
Synopsis	Main function of the cache operation sample program
Declaration	void sample_cache(void);
Description	This is the main function of the cache operation sample program. This function executes required operations, and outputs the execution results to the serial terminal. For details, see section 6.8.3, Functions Used in the Cache Operation Sample Program.
Arguments	None
Return value	None
Supplement	None

Function name: write_incr_data		
Synopsis	Function for writing evaluation data	
Declaration	void write_incr_data (uint32_t addr, uint32_t size);	
Description	This function writes increment data as a preparation for performing cache operations.	
Arguments	uint32_t addr	Data write destination address
	uint32_t size	Write data size
Return value	None	
Supplement	None	

Function name: dump_memory		
Synopsis	Memory dump function	
Declaration	void dump_memory (uint32_t addr, uint32_t size);	
Description	This function reads the contents of the specified memory, and then outputs them to the serial terminal.	
Arguments	uint32_t addr	Data dump source address
	uint32_t size	Dump data size
Return value	None	
Supplement	None	

6.9 Flowcharts of the Sample Program Functions

6.9.1 Main Function

Figure 6.1 shows the flowchart of the main processing.

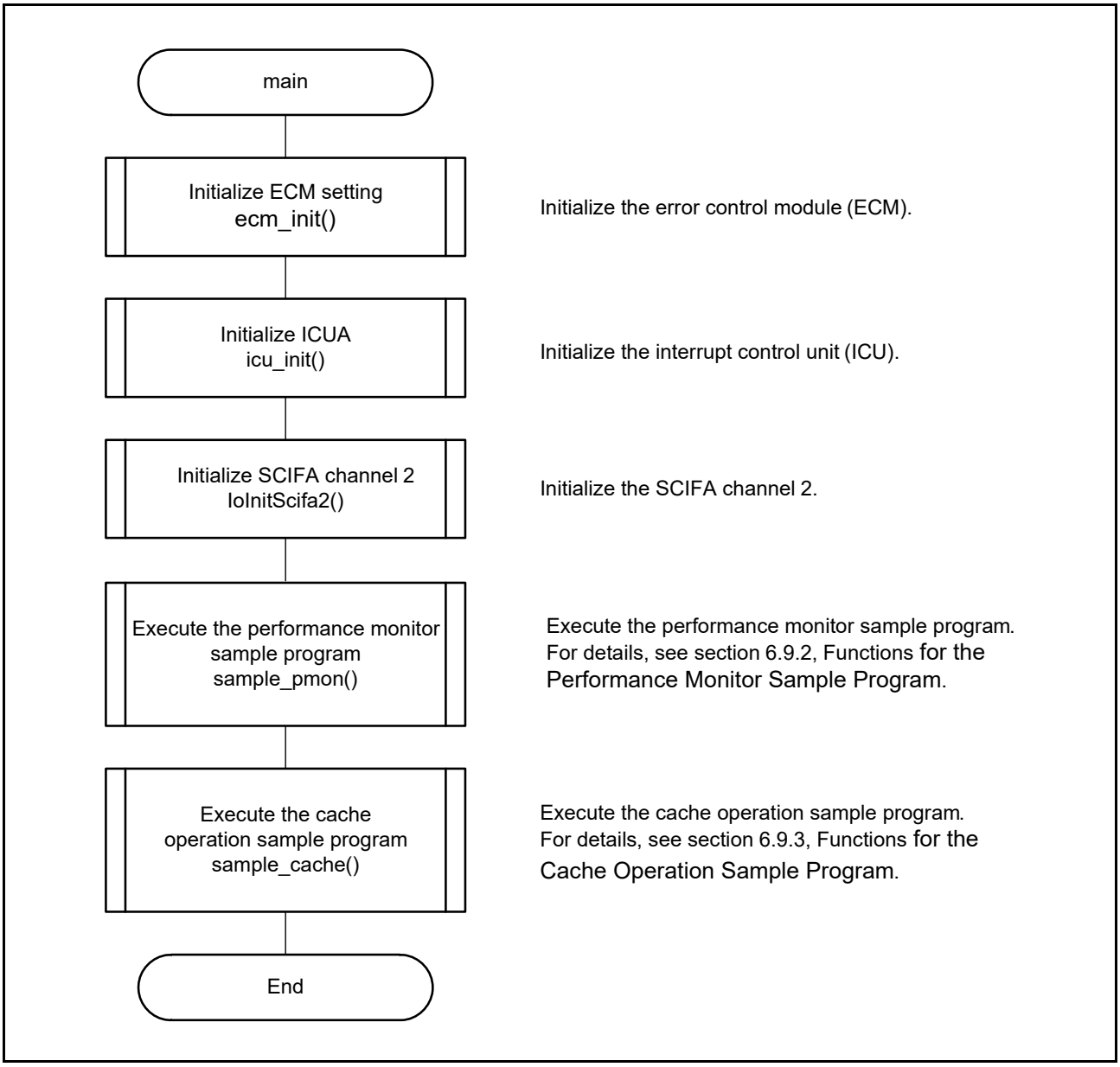


Figure 6.1 Flowchart of main

6.9.2 Functions for the Performance Monitor Sample Program

Figure 6.2 shows the flowchart of sample_pmon.

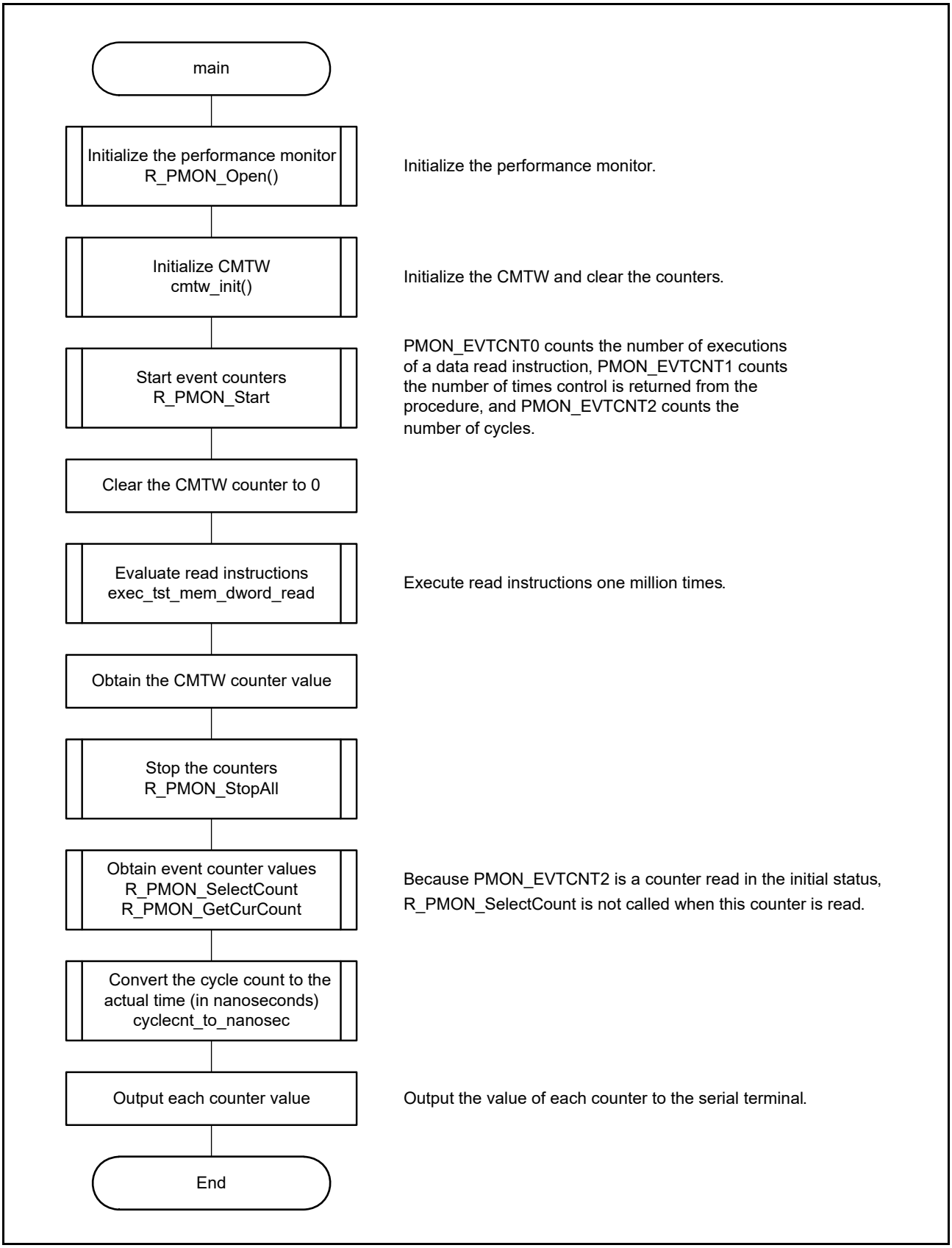


Figure 6.2 Flow of sample_pmon

6.9.3 Functions for the Cache Operation Sample Program

Figure 6.3 shows the flow of sample_cache.

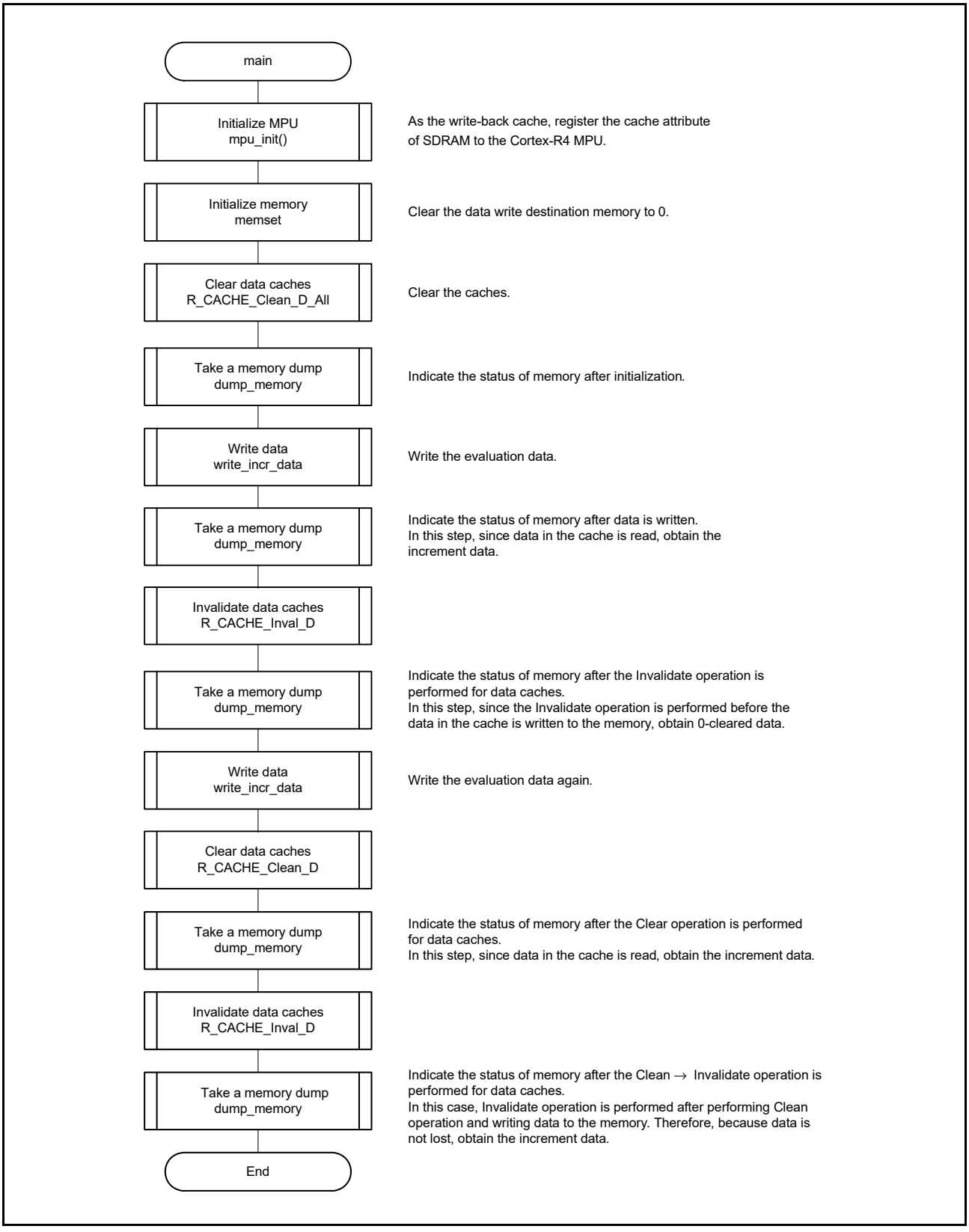


Figure 6.3 Flow of sample_cache

6.10 Operation of the Sample Program

This section describes the operation of the sample program that runs on the terminal software of the PC.

6.10.1 Project Settings

For the project settings of the development environment on the EWARM (Embedded Workbench® for Arm), DS-5, and e2studio, refer to the Application Note: RZ/T1 Group Initial Settings.

6.10.2 Preparations

This sample program uses communication with a PC. The following preparations are required for PC communication.

(1) Start the terminal software on a host PC, and then set the serial port. The following figure shows an example when COM3 is used for Tera Term.

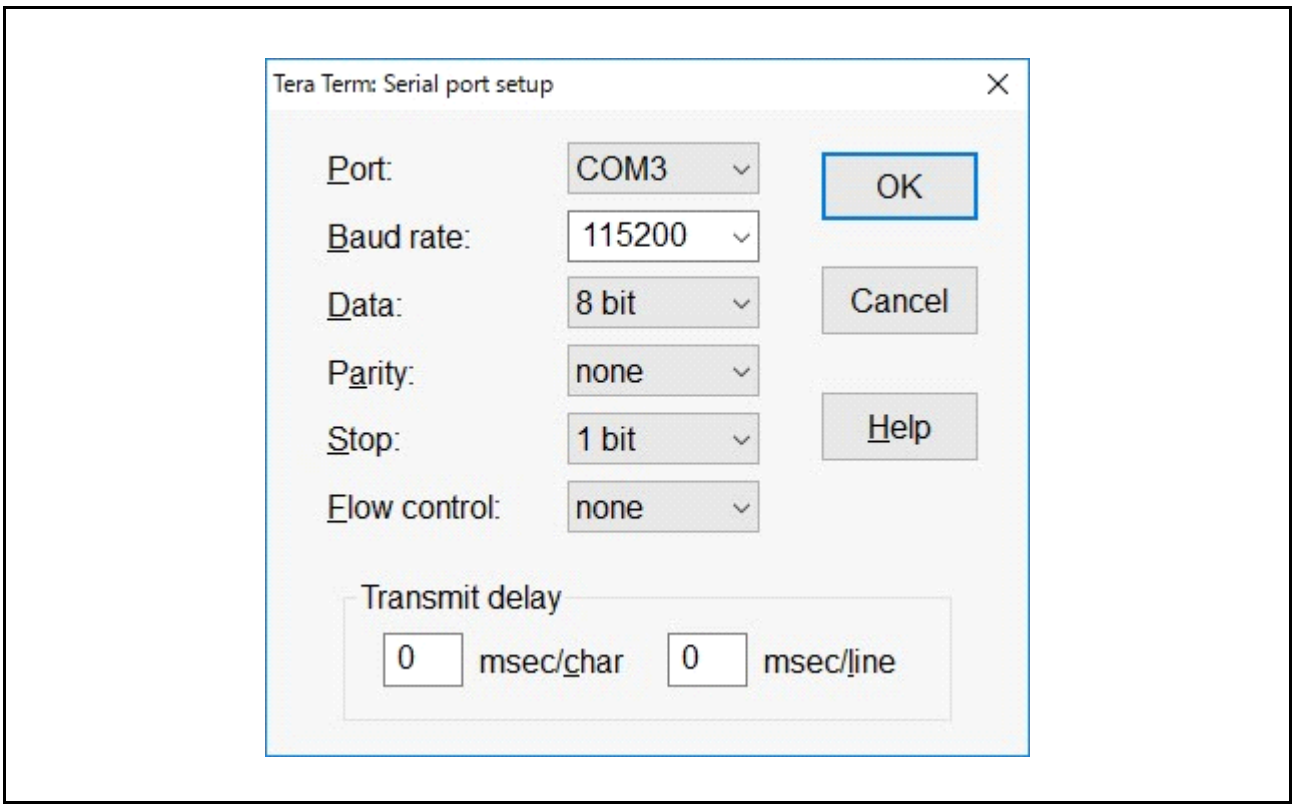


Figure 6.4 Settings of Serial Port

(2) When communications are enabled after executing the sample program, the data received from the sample program is displayed on the terminal software as shown below.

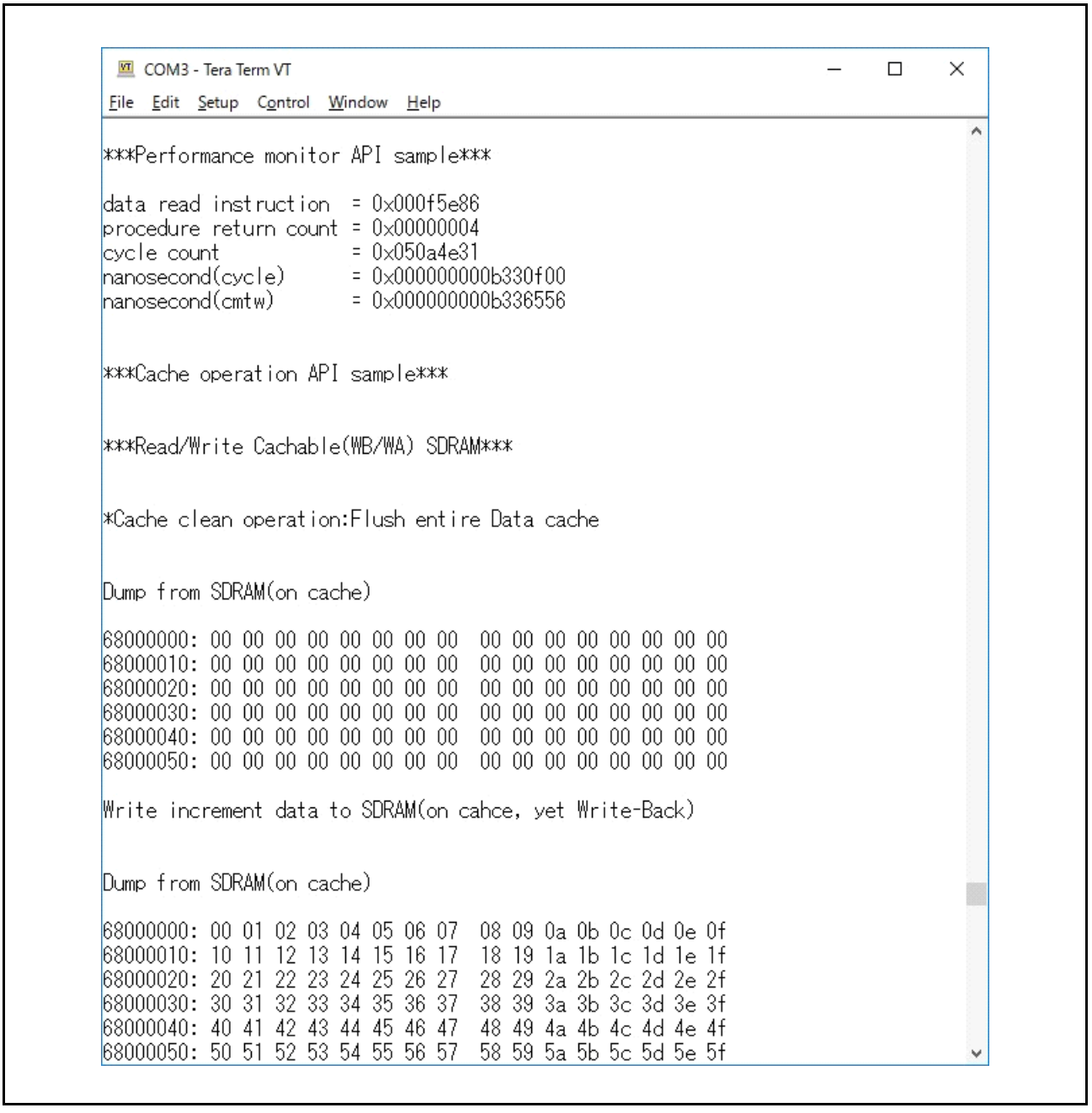


Figure 6.5 Display on Terminal Software after the Sample Program is Executed

7. Sample Program

The sample program is available on the Renesas Electronics website.

8. Related Documents and Development Environment

- User's manual: Hardware
  RZ/T1 Group User's Manual: Hardware
  (Download the latest version from the Renesas Electronics website.)

- RZ/T1 Evaluation Board RTK7910018C00000BE User's Manual
  (Download the latest version from the Renesas Electronics website.)

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

- IAR integrated development environment
  For IAR Embedded Workbench® for Arm, visit the IAR Systems website.
  (Download the latest version from the IAR Systems website.)

- Arm integrated development environment
  For an Arm Compiler toolchain, Arm DS-5, and other items, visit the Arm Corporation website.
  (Download the latest version from the Arm Corporation website.)

- Renesas integrated development environment
  For e2studio, visit the Renesas Electronics website.
  (Download the latest version from the Renesas Electronics website.)
  For a compiler and its toolchain (GNUARM-NONE), visit the GNU TOOLS & SUPPORT website (https://gcc-renesas.com/).
  (Download the latest version from the GNU TOOLS & SUPPORT website (https://gcc-renesas.com/).)

9. Website and Support

Renesas Electronics website

http://www.renesas.com/

Inquiries

http://www.renesas.com/inquiry

Revision History	Application Note: Performance Monitor Sample Program

Rev.	Date	Description	
		Page	Summary
1.00	Feb. 02, 2018	—	First Edition issued
1.10	Jun. 07, 2018	2. Operating Environment	
		4	Table 2.1 Operating Environment: The description on the integrated development environment, modified
		4. Peripheral Functions	
		6	"ARM" changed to "Arm"
		6. Software	
		21	"ARM" changed to "Arm"
		8. Related Documents and Development Environment	
		24	"ARM" changed to "Arm"

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. 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 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 Microprocessing unit or Microcontroller unit products 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.

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 or 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 matters such as the 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.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

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.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

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: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, 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 1207, 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 India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 7.0