RX Family

Snooze Mode Usage Examples

Introduction

This application note describes snooze mode, one of the low power consumption modes of the RX Family, describing its features, its effects in reducing power consumption, and examples of its use. In snooze mode on the RX Family it is possible to perform some peripheral function operations while the CPU is halted.

This application note is provided with the following sample programs as examples.

- Reception operation in asynchronous mode on the serial communications interface (SCI)
- Periodic A/D conversion operation using the low-power timer (LPT) and 12-bit A/D converter (S12AD)

Target Device

RX140 Group

RX261 Group

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

Contents

1. Snooze Mode

1.1 Overview of Snooze Mode

Snooze mode enables peripheral functions to operate temporarily while the MCU is in software standby mode. When a snooze mode transition condition occurs while in software standby mode, the oscillators and on-chip oscillators that were operating before the transition to software standby mode are restarted, supply of clock signals to the peripheral function is restarted once oscillation has stabilized, and the peripheral function is made operable.

After peripheral function operation, a return to normal operation mode can be triggered by an interrupt when peripheral function operation finishes, snooze mode status can be maintained, or a transition back to software standby mode can take place when a snooze end condition occurs. When a transition back to software standby mode occurs, operation by the oscillators, on-chip oscillators, and the peripheral function is once again halted.

Since it is possible via snooze mode to operate peripheral functions while the CPU remains halted, lower power consumption by the system can be achieved than would be the case with repeated transitions between software standby mode and normal operation mode.

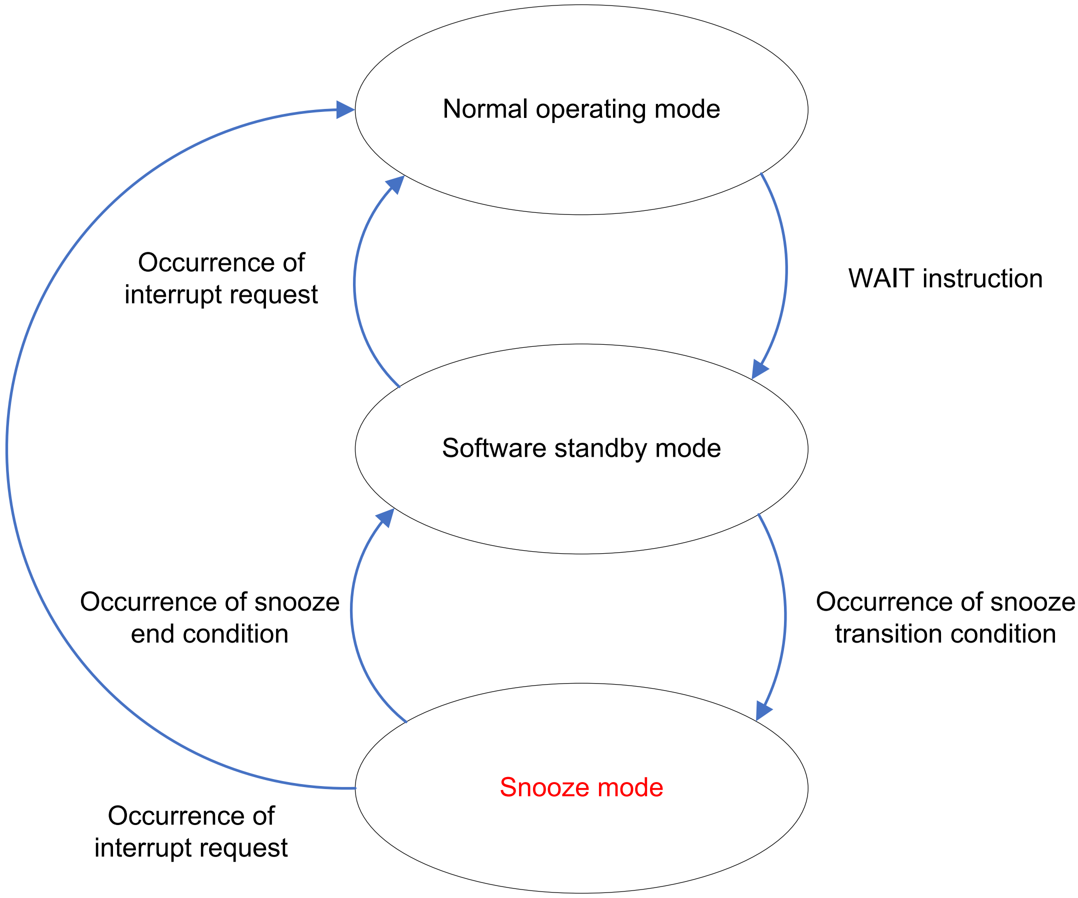


Figure 1.1 Snooze Mode State Transitions

1.2 Features of Snooze Mode

Snooze mode offers the following features.

- Ability to operate peripheral functions temporarily while in software standby mode
  It is possible to operate peripheral functions such as the 12-bit A/D converter (S12AD) and capacitive touch sensing unit (CTSU) by using as a trigger the low-power timer (LPT), which can operate in software standby mode. In addition, data reception via the serial communications interface (SCI) is supported.

- Ability to control whether: a return to normal operation mode occurs after peripheral function operation, snooze mode is maintained, or a transition back to software standby mode take places (snooze end)
  It is possible to reduce current consumption by the system overall by transitioning back to software standby mode after peripheral function operation while the CPU remains halted.

- Ability to perform transfers using the data transfer controller (DTC) in conjunction with peripheral function operation
  It is possible by using the DTC to transfer data or overwrite I/O registers while the CPU is in the stopped state.

1.3 Transitioning to and Ending Snooze Mode

A transition to snooze mode takes place when a snooze transition condition occurs while in software standby mode. Conversely, a transition from snooze mode back to software standby mode takes place when a snooze end condition occurs.

Snooze transition conditions and snooze end conditions are selected in the snooze control register (SNZCR). The SNZCR register is used to specify snooze transition conditions and snooze end conditions. Appropriate settings must be configured to match the peripheral functions to be operated in snooze mode before transitioning to software standby mode. In addition, it is necessary to set the SNZDTCE bit in the SNZCR register in order to use the DTC in snooze mode. Settings must be configured for the peripheral functions to be operated in snooze mode, the interrupt controller, etc., before transitioning to software standby mode. No transition from software standby mode to snooze mode will occur if snooze mode transition conditions have not been specified in the SNZCR register, even if an operation corresponding to a snooze transition condition takes place, and the MCU will remain in software standby mode.

Table 1.1 lists the settings for snooze transition conditions and snooze end conditions.

Table 1.1 Snooze Transition Conditions and Snooze End Conditions (SNZCR Register)

Peripheral Function	Snooze Transition Condition Settings	Snooze End Condition Settings
Serial communications interface (SCI): channel 5	• No transition to snooze mode • RXD5 falling edge detection	• Mismatch between received data and comparison data register (CDR)*1 • Mismatch between received data and comparison data register (CDR)*1 or DTC transfer*2 of received data
Low-power timer (LPT)	• No transition to snooze mode • LPT compare match 1	• No occurrence of end condition (snooze mode maintained) • DTC transfer*2 of LPT compare match 1 source
12-bit A/D converter (S12AD)	• No transition to snooze mode • LPT compare match 1	• No occurrence of end condition (snooze mode maintained) • DTC transfer*2 of A/D conversion end source
Capacitive touch sensing unit (CTSU)	• No transition to snooze mode • LPT compare match 1	• Snooze end request*3 from CTSU

Notes: 1. The data match detection function must be enabled.
2. It is necessary to use the DTC.
3. Snooze end requests are supported only on products equipped with the CTSU2SL and a ROM capacity of at least 128 KB.

1.4 Cancelling Snooze Mode

Snooze mode is canceled by an external pin interrupt (NMI or IRQ0 to IRQ7), peripheral function interrupt (RTC alarm, RTC periodic, IWDT, or voltage monitoring), pin reset, power-on reset, voltage monitoring reset, independent watchdog timer reset, or snooze mode release interrupt.

The snooze mode release interrupt is a dedicated interrupt for cancelling snooze mode that causes a return from snooze mode to normal operation mode. Snooze mode release interrupt sources are selected in snooze control register 2 (SNZCR2). When a source specified in the SNZCR2 register occurs, a snooze mode release interrupt (SNZI) is generated and snooze mode is canceled.

Table 1.2 lists items that can be specified as snooze mode release interrupt sources.

Table 1.2 Snooze Mode Release Interrupt Sources (SNZCR2 Register)

Peripheral Function	Snooze Mode Release Interrupt Sources Settings
Serial communications interface (SCI): channel 5	• No source selected • SCI5 receive error interrupt • SCI5 receive data full interrupt • DTC transfer end event*1 due to SCI5 receive data full
Low-power timer (LPT)	• No source selected • LPT compare match 1 interrupt • DTC transfer end event*1 due to LPT compare match 1
12-bit A/D converter (S12AD)	• No source selected • A/D conversion end interrupt • DTC transfer end event*1 due to A/D conversion end interrupt
Capacitive touch sensing unit (CTSU)	• No source selected • Measurement end interrupt

Note: 1. To generate a snooze mode release interrupt after a specified number of transfers has completed, clear the DTC's MRB.DISEL bit to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers).

To use the snooze mode release interrupt, it is necessary to enable interrupts for the peripheral function selected as the source and to enable the snooze mode release interrupt (SNZI) in the interrupt controller.

Table 1.3 Snooze Mode Release Interrupt (SNZI)

Name	Interrupt Source	Vector No.	Detection Method	IER	IPR	DTC
SNZI	Source selected in SNZCR2 register	81	Edge	IER0A.IEN1	IPR081	Not activatable

Note that in addition to the snooze mode release interrupt the interrupt source selected by the SNZCR2 register setting generates the interrupt selected as the source after the return to normal operation mode. If the selected peripheral function interrupt is only to be used to cancel snooze mode and for no other purpose, it can be disabled by setting its interrupt priority level to 0 (in the IPR register).

In addition, the snooze mode release interrupt is active in modes other than snooze mode. A snooze mode release interrupt will also be generated whenever an interrupt request selected in the SNZCR2 register occurs. To prevent generation of snooze mode release interrupts when not in snooze mode, disable the sources selected in the SNZCR2 register after returning to normal operation mode.

1.5 Peripheral Function Operation in Snooze Mode

In snooze mode supply of clock signals to specified peripheral functions resumes, allowing operation of peripheral functions that were halted in software standby mode. However, a trigger is required to start peripheral function operation because the transition occurs while the CPU remains stopped.

The peripheral functions that can be selected as snooze transition conditions are the low-power timer (LPT) and serial communications interface (SCI), as described in 1.3, Transitioning to and Ending Snooze Mode. These two peripheral functions are used as triggers for transitioning to snooze mode and activating peripheral functions.

However, for peripheral functions other than S12AD and CTSU it is not possible to specify settings such as a snooze end condition for transitioning back to software standby mode or a snooze mode release interrupt source. After transitioning to snooze mode it is necessary to use a suitable alternative interrupt source to return from snooze mode to normal operation mode.

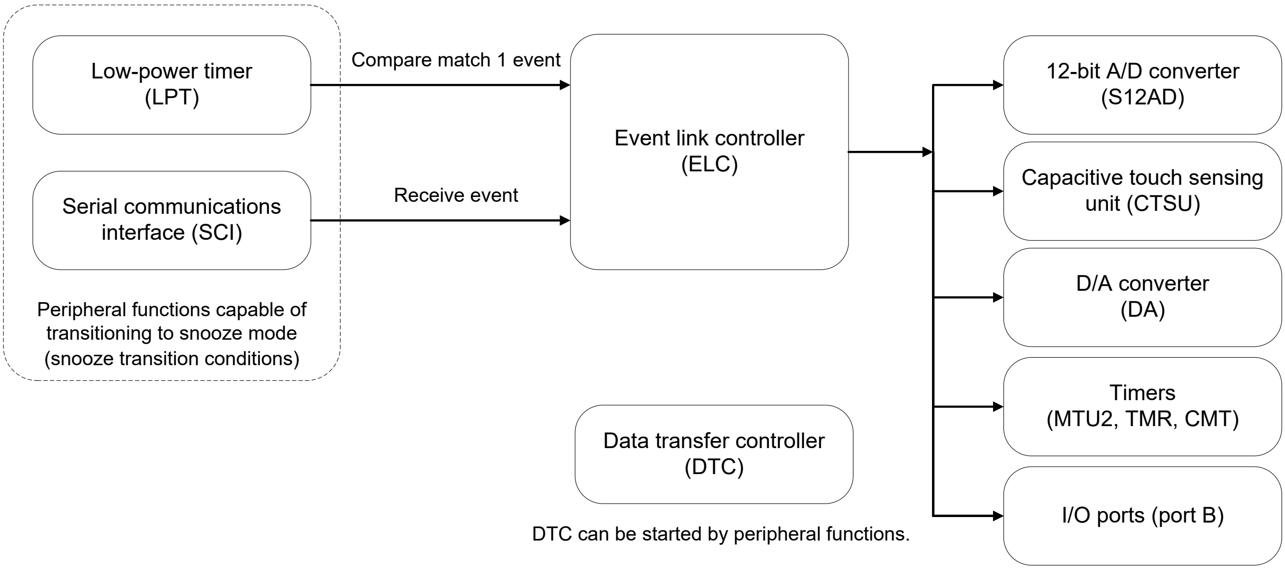


Figure 1.2 Peripheral Function Operation in Snooze Mode

Peripheral functions that can be used as snooze mode transition conditions

- Serial communications interface (SCI)
  [Snooze transition condition] RXD5 falling edge detection
- Low-power timer (LPT)
  [Snooze transition condition] LPT compare match 1

Peripheral functions that can be started via the event link controller (ELC)

- 12-bit A/D converter (S12AD)
- Capacitive touch sensing unit (CTSU)
- D/A converter (DA)
- Multi-function timer pulse unit 2 (MTU2)
- 8-bit timer (TMR)
- Compare match timer (CMT)
- I/O ports (port B only)

For operation when event signals are input, refer to Event Link Controller chapter, in each RX Family User's Manual: Hardware.

1.6 Examples of Operation Using Snooze Mode

1.6.1 Serial Communications Interface (SCI) Operation Examples

Data can be received in asynchronous mode on channel 5 of the SCI. When a falling edge is detected on RXD5 while in software standby mode, a transition to snooze mode takes place. Following the transition to snooze mode, reception operation starts after the oscillation stabilization time elapses. The operation after reception completes is specified by the settings in the SNZCR register and SNZCR2 register.

Table 1.4 lists the SCI usage conditions in snooze mode.

Table 1.4 SCI Usage Conditions in Snooze Mode

Item	Condition
Usable channel	Channel 5
Serial communication mode	Asynchronous
Transfer speed*[1]	RX140 • System clock (HOCO) frequency is 48 MHz: Up to approx. 92 kbps • System clock (HOCO) frequency is 32 MHz: Up to approx. 89 kbps • System clock (HOCO) frequency is 24 MHz: Up to approx. 87 kbps
	RX261 • System clock (HOCO) frequency is 64 MHz: Up to approx. 83 kbps • System clock (HOCO) frequency is 48 MHz: Up to approx. 83 kbps • System clock (HOCO) frequency is 32 MHz: Up to approx. 81 kbps • System clock (HOCO) frequency is 24 MHz: Up to approx. 80 kbps
Clock source	HOCO
Other clocks	LOCO, PLL, and main clock stopped
Other peripheral functions	Use of CAC prohibited (Should be set to module stop state.)

Note: 1. The data sampling timing is extended to accommodate the transition to snooze mode and the need for the oscillation stabilization time to elapse before reception operation starts. Therefore, the transfer speed is limited.

When using FIT modules, it is necessary to careful when changing the system clock source and operating frequency. If the clock source and operating frequency are changed while operation is in progress, normal operation may not be possible if each module is unable to accommodate the change in the operating frequency.

Therefore, when operating the SCI in snooze mode, either use the HOCO as the clock source from the start or, when switching to the HOCO as the clock source, ensure that the system operating frequency remains the same as before the switch.

For details of the SCI, refer to Serial Communications Interface chapter, in each RX Family User's Manual: Hardware.

Sections 1.6.1.1, Operation Example 1, to 1.6.1.3, Operation Example 3, describe examples of SCI operation in snooze mode.

1.6.1.1 Operation Example 1 (Snooze Mode Cancellation by Receive Data Full or Receive Error)

After SCI reception is enabled, a WAIT instruction is issued and a transition to software standby mode takes place. When a falling edge is detected on RXD5, a transition to snooze mode takes place and data reception is performed. After data reception finishes, a snooze mode release interrupt with a receive data full interrupt or receive error interrupt as the source triggers a return to normal operation mode.

After the return to normal operation mode, a snooze mode release interrupt and a receive data full interrupt or receive error interrupt are generated.

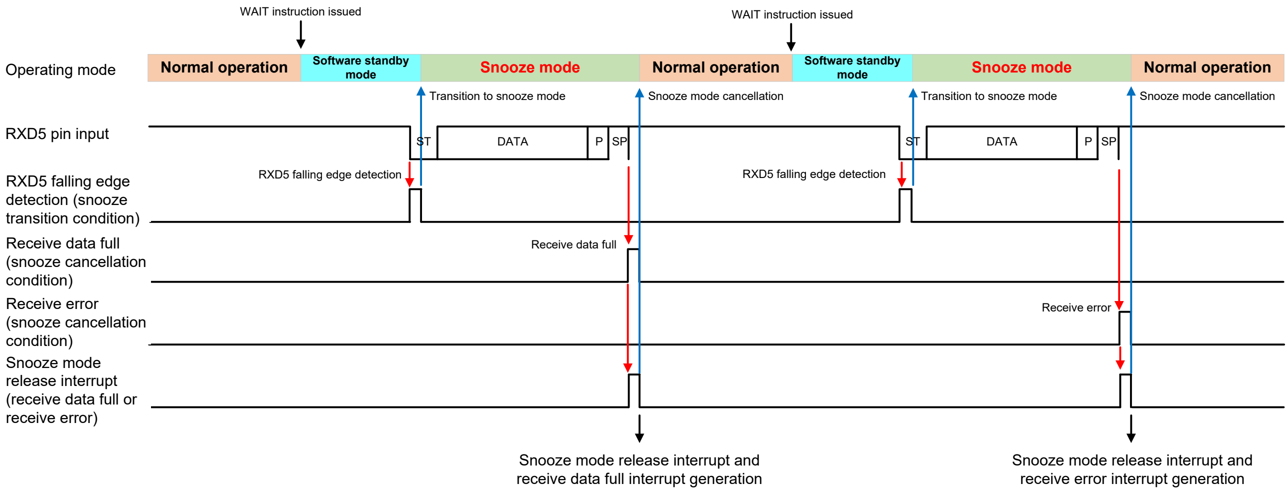


Figure 1.3 SCI Operation Example 1 (Snooze Mode Cancellation by Receive Data Full or Receive Error)

Table 1.5 SCI Operation Example 1 Setting Values

Register	Bit	Setting Value	Description
SNZCR	SCISNZSEL	10b	When a falling edge is detected on RXD5, a transition to snooze mode takes place, and a return to software standby mode occurs if the received data does not match the SCI5.CDR register value.
SNZCR2	SCIERE	1b	Selects the SCI5 receive error interrupt as the source for the snooze mode release interrupt.
	SCIRXE[1:0]	01b	Selects the SCI5 receive data full interrupt as the source for the snooze mode release interrupt.
IER0A	IEN1	1b	SNZI interrupt requests enabled.
IPR081	IPR[3:0]	Other than 0	Priority level of SNZI interrupt

1.6.1.2 Operation Example 2 (Snooze Mode End by Data Mismatch and Snooze Mode Cancellation by Receive Data Full)

This operation example requires that the SCI's data match detection function be enabled.

After SCI reception is enabled, a WAIT instruction is issued and a transition to software standby mode takes place. When a falling edge is detected on RXD5, a transition to snooze mode takes place and data reception is performed. After data reception finishes, a data comparison is performed using the data match detection function of the SCI. If the comparison result is "mismatch," a transition to software standby mode takes place. If the comparison result is "match," a snooze mode release interrupt with a receive data full interrupt as the source triggers a return to normal operation mode.

After the return to normal operation mode, a snooze mode release interrupt and a receive data full interrupt are generated.

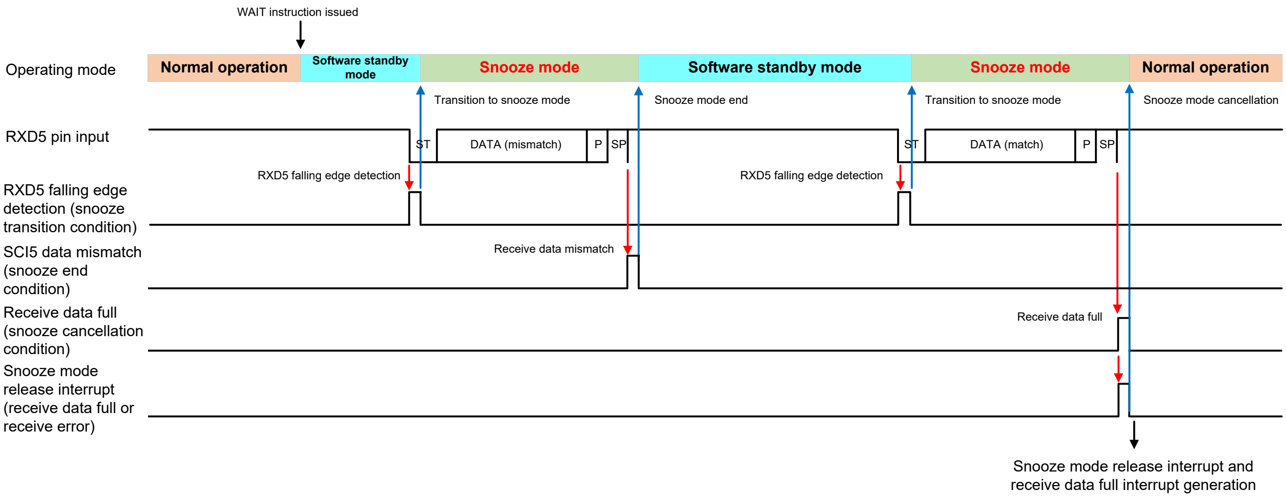


Figure 1.4 SCI Operation Example 2 (Snooze Mode End by Data Mismatch and Snooze Mode Cancellation by Receive Data Full)

Table 1.6 SCI Operation Example 2 Setting Values

Register	Bit	Setting Value	Description
SNZCR	SCISNZSEL	10b	When a falling edge is detected on RXD5, a transition to snooze mode takes place, and a return to software standby mode occurs if the received data does not match the SCI5.CDR register value.
SNZCR2	SCIERE	0b	SCI5 receive error not selected as condition for snooze mode release interrupt.
	SCIRXE[1:0]	01b	Selects the SCI5 receive data full interrupt as the source for the snooze mode release interrupt.
IER0A	IEN1	1b	SNZI interrupt requests enabled.
IPR081	IPR[3:0]	Other than 0	Priority level of SNZI interrupt

1.6.1.3 Operation Example 3 (Snooze Mode End by DTC Transfer End and Snooze Mode Cancellation by DTC Transfer End Interrupt)

After SCI reception is enabled, a WAIT instruction is issued and a transition to software standby mode takes place. When a falling edge is detected on RXD5, a transition to snooze mode takes place and data reception is performed. After data reception finishes, a DTC transfer takes place according to the transfer settings, and a transition to software standby mode occurs after the transfer ends. After the final data is transferred, a snooze mode release interrupt with a DTC transfer end event at receive data full as the source triggers a return to normal operation mode.

After the return to normal operation mode, a snooze mode release interrupt and a receive data full interrupt are generated.

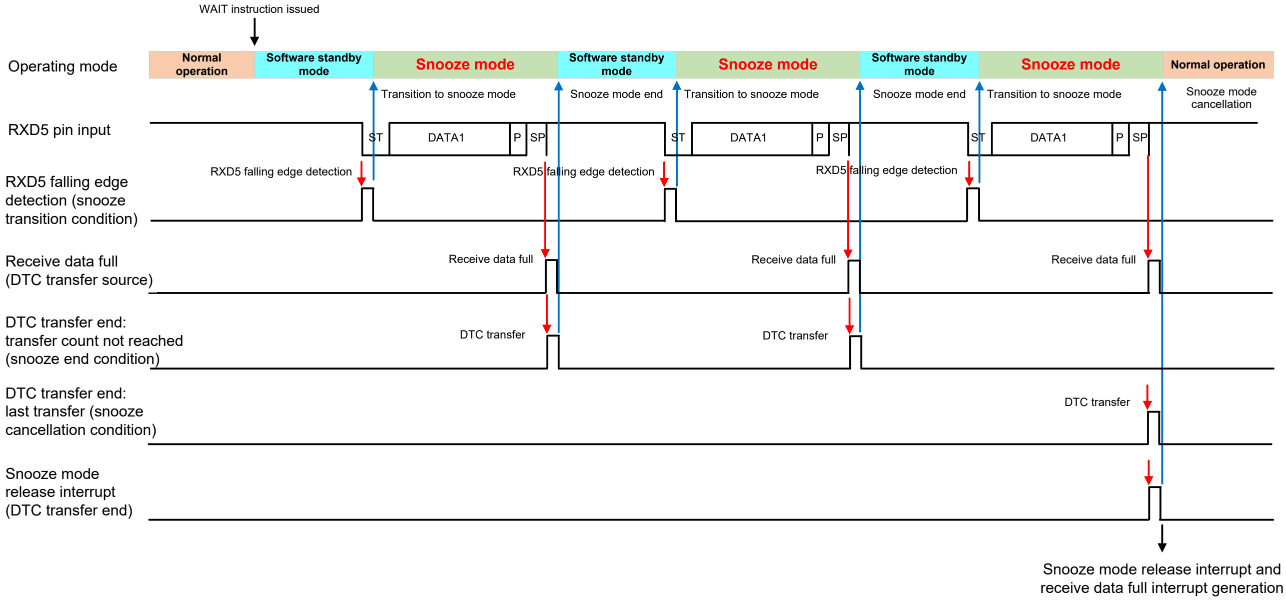


Figure 1.5 SCI Operation Example 3 (Snooze Mode End by DTC Transfer End and Snooze Mode Cancellation by DTC Transfer End Interrupt)

Table 1.7 SCI Operation Example 3 Setting Values

Register	Bit	Setting Value	Description
SNZCR	SCISNZSEL	11b	When a falling edge is detected on RXD5, a transition to snooze mode takes place, and a return to software standby mode occurs if the received data does not match the SCI5.CDR register value or after DTC transfer of the received data finishes.
	SNZDTCE	1b	DTC transfers enabled in snooze mode.
SNZCR2	SCIERE	0b	SCI5 receive error not selected as condition for snooze mode release interrupt.
	SCIRXE[1:0]	1xb	Selects a DTC transfer end event at SCI5 receive data full as the source for the snooze mode release interrupt.
IER0A	IEN1	1b	SNZI interrupt requests enabled.
IPR081	IPR[3:0]	Other than 0	Priority level of SNZI interrupt

1.6.2 12-Bit A/D Converter (S12AD) Operation Examples

Periodic A/D conversion operation can be implemented by using the LPT and ELC in combination. LPT compare match 1 can be used as the trigger for transitioning to snooze mode, and A/D conversion can be started on the S12AD via the ELC.

Table 1.8 lists the S12AD usage conditions in snooze mode.

Table 1.8 S12AD Usage Conditions in Snooze Mode

Item	Condition
Operating mode	Single scan mode
Conversion target	Analog input (conversion of temperature sensor or internal reference voltage prohibited)

For details of the LPT, ELC, and S12AD, refer to Event Link Controller chapter, Low-Power Timer chapter, and 12-Bit A/D Converter chapter, in each RX Family User's Manual: Hardware.

Sections 1.6.2.1, Operation Example 1, and 1.6.2.2, Operation Example 2, describe examples of S12AD operation in snooze mode.

1.6.2.1 Operation Example 1 (Snooze Mode Cancellation by A/D Conversion End)

After count operation starts on the LPT, a WAIT instruction is issued and a transition to software standby mode takes place. When an LPT compare match 1 occurs, a transition to snooze mode takes place and A/D conversion starts via the ELC. After A/D conversion ends, a snooze mode release interrupt with a scan end interrupt as the source triggers a return to normal operation mode.

After the return to normal operation mode, a snooze mode release interrupt and a scan end interrupt are generated.

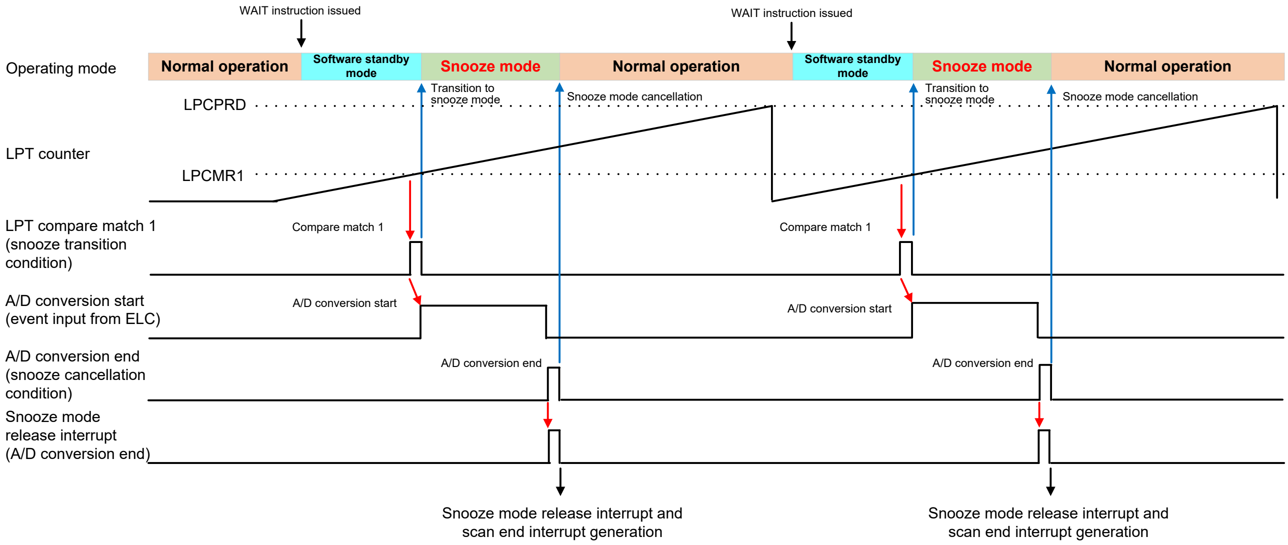


Figure 1.6 S12AD Operation Example 1 (Snooze Mode Cancellation by A/D Conversion End)

Table 1.9 S12AD Operation Example 1 Setting Values

Register	Bit	Setting Value	Description
SNZCR	ADCSNZSEL [1:0]	10b	Transition to snooze mode at LPT compare match 1, and no return to software standby mode.
SNZCR2	ADE[1:0]	01b	Selects the S12AD conversion end interrupt as the source for the snooze mode release interrupt.
IER0A	IEN1	1b	SNZI interrupt requests enabled.
IPR081	IPR[3:0]	Other than 0	Priority level of SNZI interrupt

1.6.2.2 Operation Example 2 (Snooze Mode End by DTC Transfer End and Snooze Mode Cancellation by DTC Transfer End Interrupt)

After count operation starts on the LPT, a WAIT instruction is issued and a transition to software standby mode takes place. When an LPT compare match 1 occurs, a transition to snooze mode takes place and A/D conversion starts via the ELC. After A/D conversion ends, a DTC transfer takes place according to the transfer settings, and a transition to software standby mode occurs after the transfer ends. After the final data is transferred, a snooze mode release interrupt with a DTC transfer end event at scan end as the source triggers a return to normal operation mode.

After the return to normal operation mode, a snooze mode release interrupt and a scan end interrupt are generated.

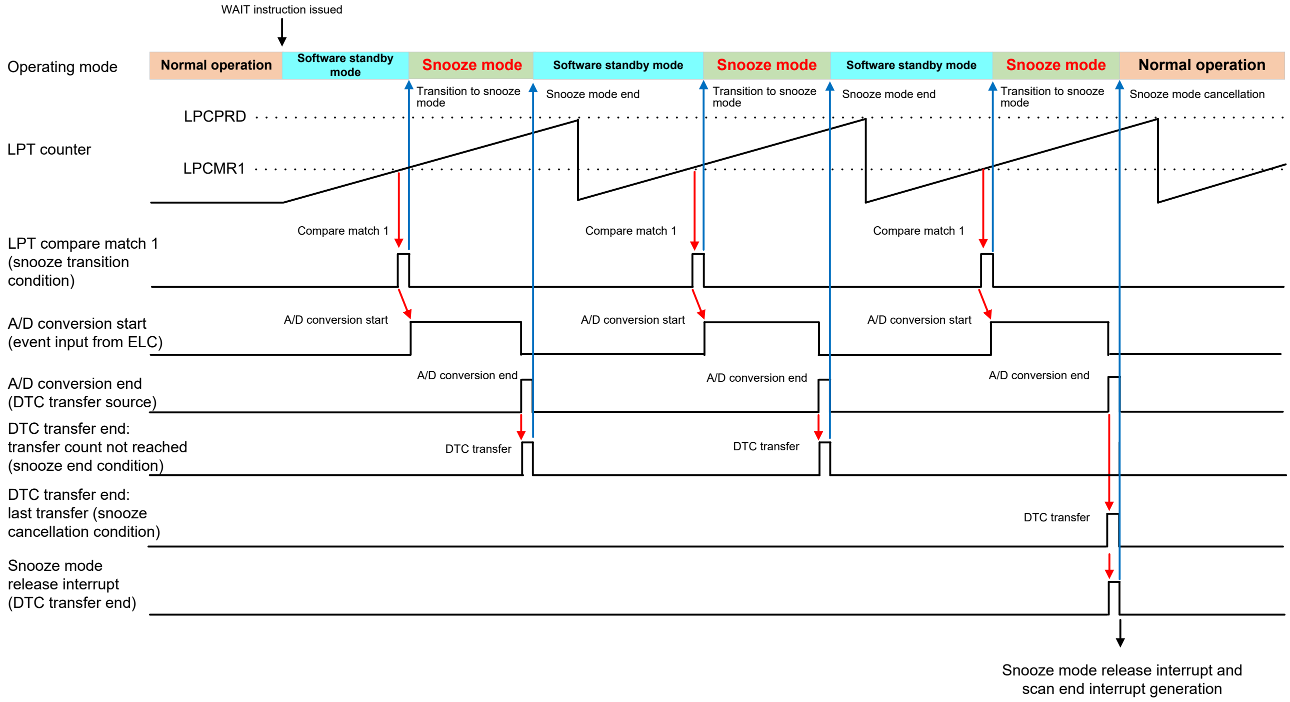


Figure 1.7 S12AD Operation Example 2 (Snooze Mode End by DTC Transfer End and Snooze Mode Cancellation by DTC Transfer End Interrupt)

Table 1.10 S12AD Operation Example 2 Setting Values

Register	Bit	Setting Value	Description
SNZCR	ADCSNZSEL [1:0]	11b	Transition to snooze mode at LPT compare match 1, and return to software standby mode after one DTC transfer ends with A/D conversion end as the source.
	SNZDTCE	1b	DTC transfers enabled in snooze mode.
SNZCR2	ADE[1:0]	1xb	Selects a DTC transfer end event at S12AD conversion as the source for the snooze mode release interrupt.
IER0A	IEN1	1b	SNZI interrupt requests enabled.
IPR081	IPR[3:0]	Other than 0	Priority level of SNZI interrupt

1.6.3 Capacitive Touch Sensing Unit (CTSU) Operation Example

Periodic measurement operation can be implemented by using the LPT and ELC in combination. LPT compare match 1 can be used as the trigger for transitioning to snooze mode, and measurement operation can be started on the CTSU via the ELC. Note that generation of snooze end conditions is supported only on products equipped with a capacitive touch sensing unit (CTSU2SL) with automatic judgment function and a ROM capacity of at least 128 KB. On products equipped with the CTSU2L it is not possible to use a snooze end condition to trigger transition to software standby mode.

Table 1.11 lists the CTSU usage conditions in snooze mode.

Table 1.11 CTSU Usage Conditions in Snooze Mode

Item	Condition
Snooze end condition generation	Product with CTSU2SL and ROM capacity of at least 128 KB

For details of the LPT, ELC, and CTSU, refer to Event Link Controller chapter, Low-Power Timer chapter, and Capacitive Touch Sensing Unit chapter, in each RX Family User's Manual: Hardware.

Section 1.6.3.1 describes an example of CTSU operation in snooze mode.

1.6.3.1 Operation Example (Snooze Mode Cancellation by Measurement End)

This operation example applies to the CTSU2SL, which can generate snooze end conditions.

After count operation starts on the LPT, a WAIT instruction is issued and a transition to software standby mode takes place. When an LPT compare match 1 occurs, a transition to snooze mode takes place and measurement operation starts via the ELC. After measurement ends on one channel, a DTC transfer takes place according to the transfer settings, and a transition to software standby mode occurs after the transfer ends. After measurement finishes on all channels and the final data is transferred, a snooze mode release interrupt with a measurement end interrupt as the source triggers a return to normal operation mode.

After the return to normal operation mode, a snooze mode release interrupt, setting register write request interrupt, measurement data transfer request interrupt, and measurement end interrupt are generated.

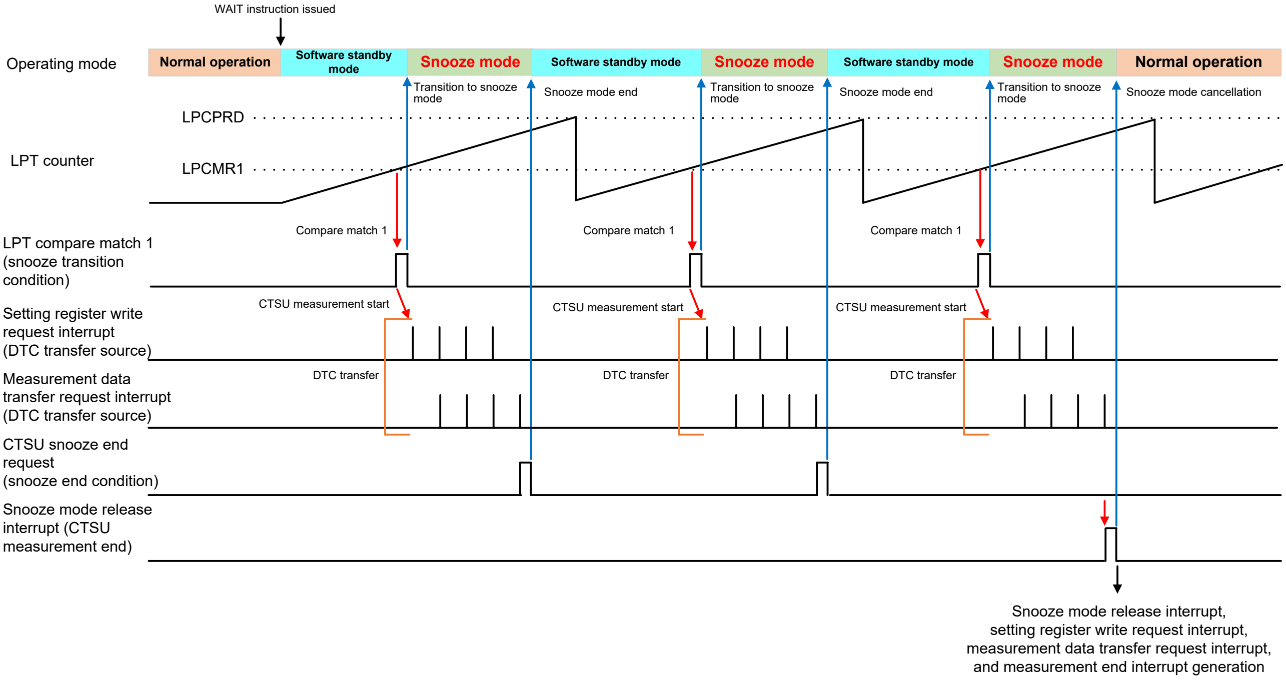


Figure 1.8 CTSU Operation Example (Snooze Mode Cancellation by Measurement End)

Table 1.12 Operation Example Setting Values

Register	Bit	Setting Value	Description
SNZCR	CTSUSNZSEL [1:0]	10b	Transition to snooze mode at LPT compare match 1, and return to software standby mode at snooze end request from CTSU.
	SNZDTCE	1b	DTC transfers enabled in snooze mode.
SNZCR2	CTSUFNE [1:0]	01b	Selects the CTSU measurement end interrupt as the source for the snooze mode release interrupt.
IER0A	IEN1	1b	SNZI interrupt requests enabled.
IPR081	IPR[3:0]	Other than 0	Priority level of SNZI interrupt

1.6.4 Low-Power Timer (LPT) Operation Example

LPT compare match 1 can be used as the trigger for transitioning to snooze mode, and a peripheral function set as an ELC event link target can be operated via the ELC. (For details of the S12AD and CTSU, refer to 1.6.2, 12-Bit A/D Converter (S12AD) Operation Examples, and 1.6.3, Capacitive Touch Sensing Unit (CTSU) Operation Example.)

For details of the LPT and ELC, refer to Low-Power Timer chapter, and Event Link Controller chapter, in each RX Family User's Manual: Hardware.

Section 1.6.4.1 describes an example of LPT operation in snooze mode.

1.6.4.1 Operation Example 1 (PWM Waveform Output, Snooze Mode End at DTC Transfer End, and Snooze Mode Cancellation by DTC Transfer End Interrupt)

In this operation example PWM waveforms are output, and the duty ratio of the waveforms is modified by overwriting low-power timer compare register 0 (LPCMR0) using DTC transfers while in snooze mode.

After count operation starts on the LPT, a WAIT instruction is issued and a transition to software standby mode takes place. When an LPT compare match 0 occurs, the output on the LPTO pin changes. After the output on the LPTO pin changes and a transition to snooze mode takes place when an LPT compare match 1 occurs, a DTC transfer takes place according to the transfer settings, and the LPCMR0 register is overwritten. A transition to software standby mode occurs after the transfer ends. After the final data is transferred, a snooze mode release interrupt with a DTC transfer end event at compare match 1 as the source triggers a return to normal operation mode.

After the return to normal operation mode, a snooze mode release interrupt and compare match 1 interrupt are generated.

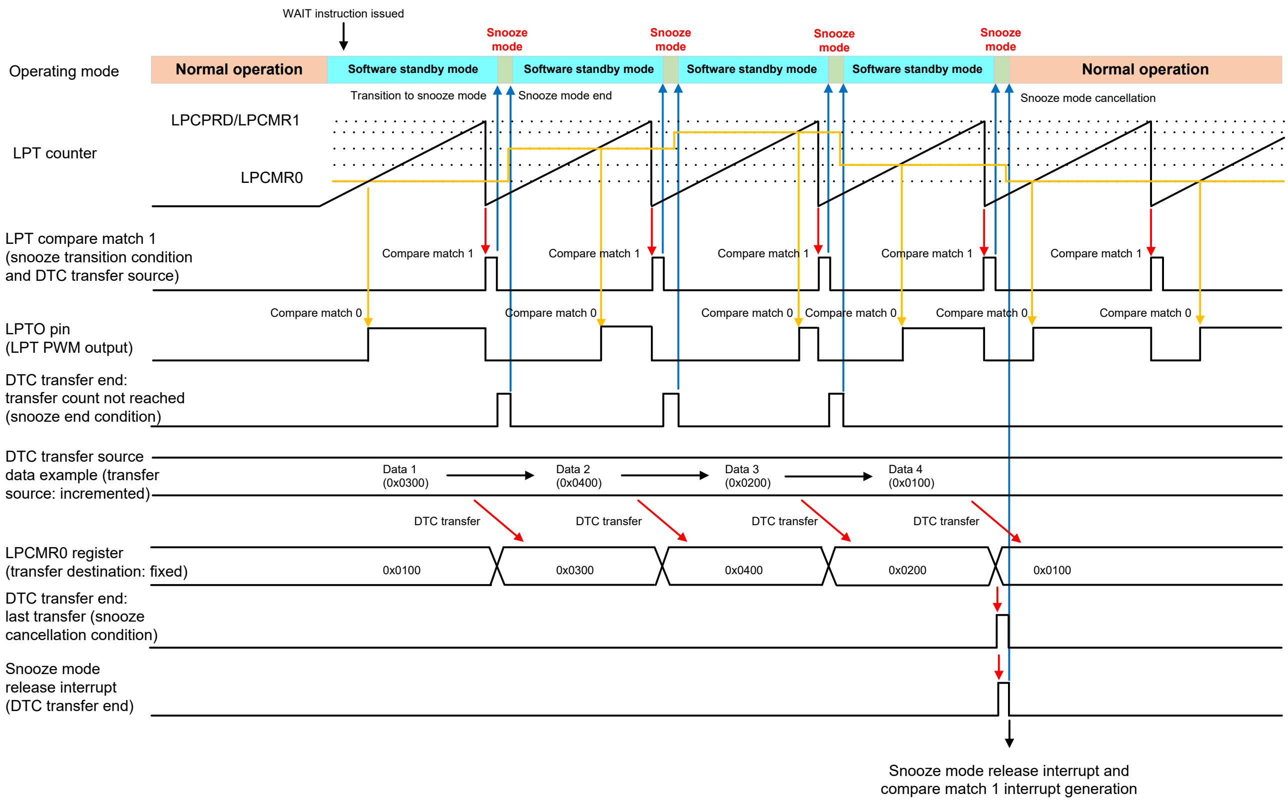


**Figure 1.9 LPT Operation Example 1
(Snooze Mode End at DTC Transfer End and Snooze Mode Cancellation by DTC Transfer End Interrupt)**

Table 1.13 LPT Operation Example 1 Setting Values

Register	Bit	Setting Value	Description
SNZCR	LPTSNZSEL [1:0]	11b	Transition to snooze mode at LPT compare match 1, and return to software standby mode after one DTC transfer ends at LPT compare match 1 as the source.
	SNZDTCE	1b	DTC transfers enabled in snooze mode.
SNZCR2	LPTCM1E [1:0]	1xb	Selects a DTC transfer end event at LPT compare match 1 as the source for the snooze mode release interrupt.
IER0A	IEN1	1b	SNZI interrupt requests enabled.
IPR081	IPR[3:0]	Other than 0	Priority level of SNZI interrupt

1.7 Snooze Mode Usage Notes

1.7.1 Duration of Transition from Software Standby Mode to Snooze Mode

When a snooze mode transition condition occurs, the oscillators and on-chip oscillators that were operating before the transition to software standby mode are restarted, and the peripheral function starts operating once oscillation has stabilized. Therefore, it is necessary to consider the duration of the transition from software standby mode to snooze mode when designing software.

For details of the duration of the transition to snooze mode, refer to Timing of Recovery from Low Power Consumption Modes chapter, in each RX Family User's Manual: Hardware.

1.7.2 Using the Data Transfer Controller (DTC)

When using the DTC, make sure to confirm the following settings as necessary.

Table 1.14 DTC Setting Values

Register	Bit	Setting Value	Notes
MRA	WBDIS	0: Write back transfer information on completion of data transfer. 1: Do not write back transfer information on completion of data transfer.	When a DTC transfer end event is selected as the source of the snooze mode release interrupt, this bit must be cleared to 0 in order to update the transfer count.
	MD[1:0]	0 0: Normal transfer mode 0 1: Repeat transfer mode 1 0: Block transfer mode	It is not possible to use repeat transfer mode when a DTC transfer end event is selected as the source of the snooze mode release interrupt and the value of MRB.DISEL is 0. In repeat transfer mode the transfer count never reaches 0, and no interrupt request to the CPU is generated when transfer ends while the value of MRB.DISEL is 0, preventing generation of a snooze mode release interrupt.
MRB	DISEL	0: Generate an interrupt request to the CPU on completion of the specified number of data transfers. 1: Generate an interrupt request to the CPU for each data transfer.	Clear this bit to 0 in order to generate a snooze mode release interrupt after the specified number of transfers. When a DTC transfer end event is selected as the source of the snooze mode release interrupt and MRB.DISEL is set to 1, a snooze mode release interrupt is generated for each transfer.

2. Sample Programs

This application note is provided with the following sample programs utilizing snooze mode.

Table 2.1 Sample Programs

Project Name	Description
snooze_sci5_rx140 snooze_sci5_rx140_rsk snooze_sci5_rx261	Reception operation in asynchronous mode on the serial communications interface (SCI)
snooze_s12ad_rx140 snooze_s12ad_rx140_rsk snooze_s12ad_rx261	Periodic A/D conversion operation using the low-power timer (LPT) and 12-bit A/D converter (S12AD)

These sample programs support the following three boards: Target Board for RX140, Renesas Starter Kit for RX140, and EK-RX261.

The integrated development environment used by the sample programs comprises e^2 studio and Smart Configurator (SC). In addition, Firmware Integration Technology (FIT) modules are used to configure settings and control the peripheral functions.

For details of the FIT modules and settings used, refer to 2.1.5, Structure of Sample Program, and 2.2.5, Structure of Sample Program.

2.1 Reception Operation in Asynchronous Mode on the Serial Communications Interface (SCI)

2.1.1 Sample Program Specifications

2.1.1.1 Software

Reception operation in asynchronous mode is performed on SCI channel 5 while in snooze mode. The operation of this sample program is described in 1.6.1.3, Operation Example 3 (Snooze Mode End by DTC Transfer End and Snooze Mode Cancellation by DTC Transfer End Interrupt)).

After initial settings for the peripheral function are configured, a transition to software standby mode takes place. A transition to snooze mode occurs when the falling edge of the start bit is detected on the RXD5 pin, and reception operation starts. After one byte of data is received and transferred to the buffer by the DTC, a transition to software standby mode takes place. Reception operation is repeated in like manner until all five bytes of data have been received, at which point a return to normal operation mode is triggered by the snooze mode release interrupt.

After the return to normal operation mode, the received data is displayed in Debug Virtual Console, and about three seconds later a transition to software standby mode takes place.

Table 2.2 lists the peripheral functions used by the sample program.

Table 2.2 Descriptions of Peripheral Functions Used

Peripheral Function	Description
Low power consumption functions	Controls software standby mode and snooze mode.
SCI (channel 5)	Performs serial communication in asynchronous mode.
DTC	Transfers received data to the buffer via the SCI.
I/O ports	Controls LED illumination. The LED turns on in normal operation mode and turns off in low power consumption mode.

The peripheral function settings are listed below.

- Low power consumption function settings
  - Low power consumption modes: Software standby mode, snooze mode
  - Snooze mode transition condition: Falling edge detection on RXD5
  - Snooze mode end condition: Mismatch between received data and value of comparison data register (CDR) or after DTC transfer of received data
  - Snooze mode release interrupt source: DTC transfer end event at SCI5 receive data full

- SCI (channel 5) settings
  - Operating mode: Asynchronous mode
  - Transfer speed (RX140): 90 kbps
  - Transfer speed (RX261): 80 kbps
  - Data length: 8 bits
  - Stop bits: 1 bit
  - Parity bits: None
  - Flow control: None
  - Data match detection function: Disabled

- DTC settings (configured in SCI FIT module)
  - Transfer mode: Normal transfer mode
  - Transfer unit: 1 byte
  - Transfer count: 5
  - Transfer source: Receive data register 5 (RDR5)
  - Transfer destination: Buffer variable (See 2.1.5.5, Variables.)
  - Transfer source: SCI5 receive data full interrupt
  - CPU interrupt timing: An interrupt request is issued to the CPU upon completion of the specified number of data transfers.
  - Write-back disable: Write-back of transfer information occurs upon completion of data transfer.
  - Chain transfer: Disabled
  - Sequence transfer: Disabled

- I/O port settings*1
  - Port 31: Initially low output, high (LED off) or low (LED on) output according to program state
  - Port 21: Initially low output, high (LED off) or low (LED on) output according to program state
  - Port J1: Initially high output, low (LED off) or high (LED on) output according to program state

Note: 1. When using the sample programs for the Target Board for RX140 board, use port 31. When using the sample programs for the Renesas Starter Kit for RX140 board, use port 21. When using the sample programs for the EK-RX261 board, use port J1.

2.1.1.2 Pins Used

Table 2.3 and Table 2.4 lists the pins and functions used by the sample program.

Table 2.3 Pins and Functions Used (RX140)

Pin Name	I/O	Description
P31	Output	LED0 (normal operation mode: on, low power consumption mode: off)
P21		
PC2/RXD5	Input	SCI5 receive data input pin
PC3/TXD5	Output	SCI5 transmit data input pin

Table 2.4 Pins and Functions Used (RX261)

Pin Name	I/O	Description
PJ1	Output	LED1 (normal operation mode: on, low power consumption mode: off)
PC2/RXD5	Input	SCI5 receive data input pin
PC3/TXD5	Output	SCI5 transmit data input pin

2.1.1.3 Outline Flowchart

Figure 2.1 is an outline flowchart of the sample program.

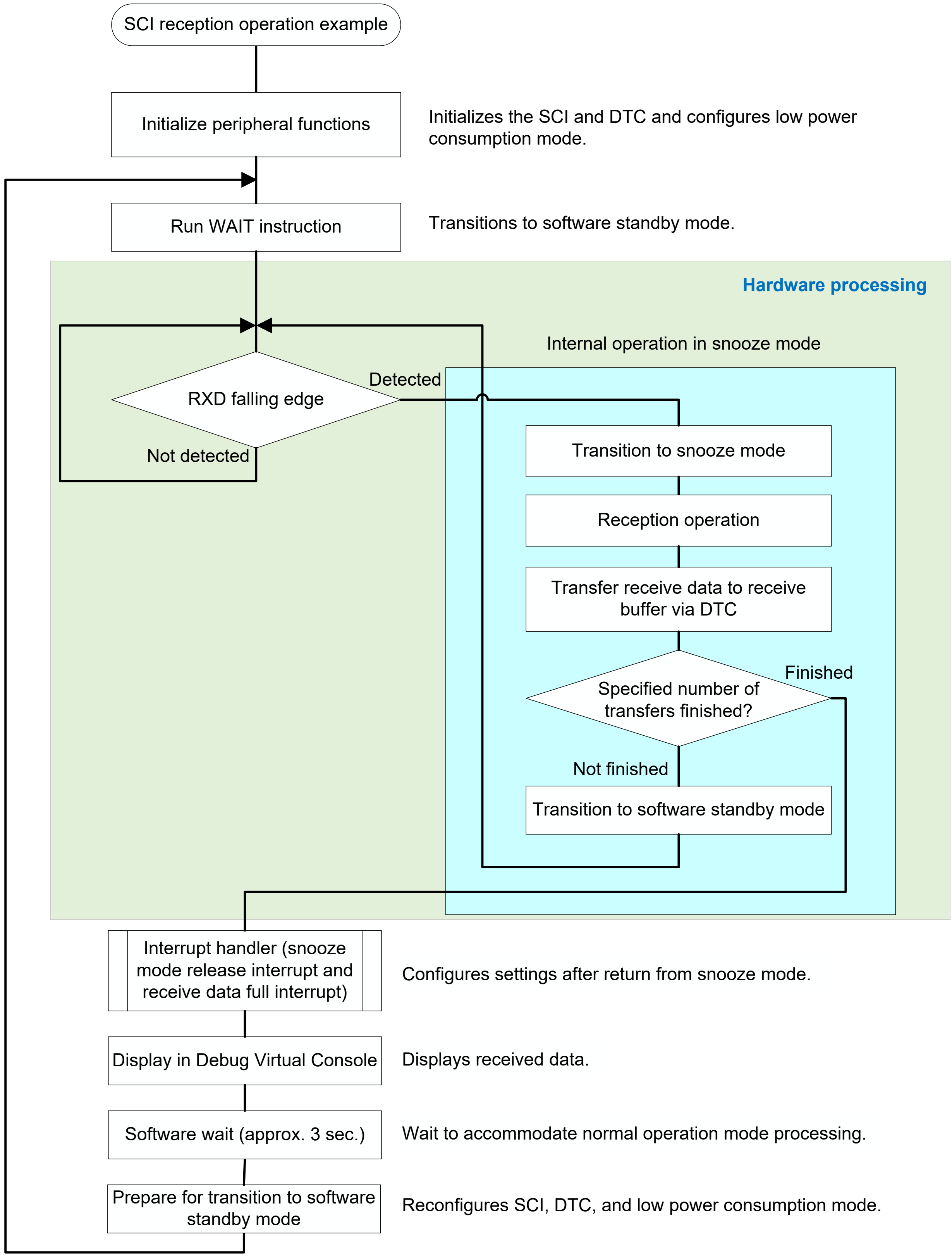


Figure 2.1 Outline Flowchart of Reception Operation in SCI Asynchronous Mode

2.1.2 Operation Confirmation Conditions

The operation of the sample program has been confirmed under the following conditions.

Table 2.5 Operation Confirmation Conditions (Reception Operation in SCI Asynchronous Mode)

Item	Description
MCU	R5F51403ADFM (RX140 Group) R5F51406BDFN (RX140 Group) R5F52618BGFP (RX261 Group)
Operating frequency	RX140 • HOCO clock: 48 MHz • System clock (ICLK): 48 MHz (HOCO clock divided by 1) • Peripheral module clock B (PCLKB): 24 MHz (HOCO clock divided by 2) • Peripheral module clock D (PCLKD): 48 MHz (HOCO clock divided by 1) • FlashIF clock (FCLK): 48 MHz (HOCO clock divided by 1)
	RX261 • HOCO clock: 64 MHz • System clock (ICLK): 64 MHz (HOCO clock divided by 1) • Peripheral module clock B (PCLKB): 32 MHz (HOCO clock divided by 2) • Peripheral module clock D (PCLKD): 64 MHz (HOCO clock divided by 1) • FlashIF clock (FCLK): 64 MHz (HOCO clock divided by 1)
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics e^2 studio Version 2024-07
C compiler	Renesas Electronics C/C++ Compiler Package for RX Family V.3.06.00
	Compiler options -lang = c99
iodefine.h version	V1.10A (RX140) V1.00 (RX261)
Endian order	Little endian or big endian
Operating mode	Single-chip mode
Processor mode	Supervisor mode
Emulator	On-board emulator E2 Lite emulator
Board used	Target Board for RX140 (product No.: RTK5RX140xxxxxxxxx) Renesas Starter Kit for RX140 (product No.: RTK551406xxxxxxxxx) EK-RX261 (product No.: RTK5EK261xxxxxxxxx)

2.1.3 Confirming Operation of the Sample Program

How to confirm the operation of the sample program is described below.

2.1.3.1 Preparing Devices

To prepare to run the sample program, make connections as shown in Figure 2.2. Prepare a device to input serial data on the PC2/RXD5 pin.

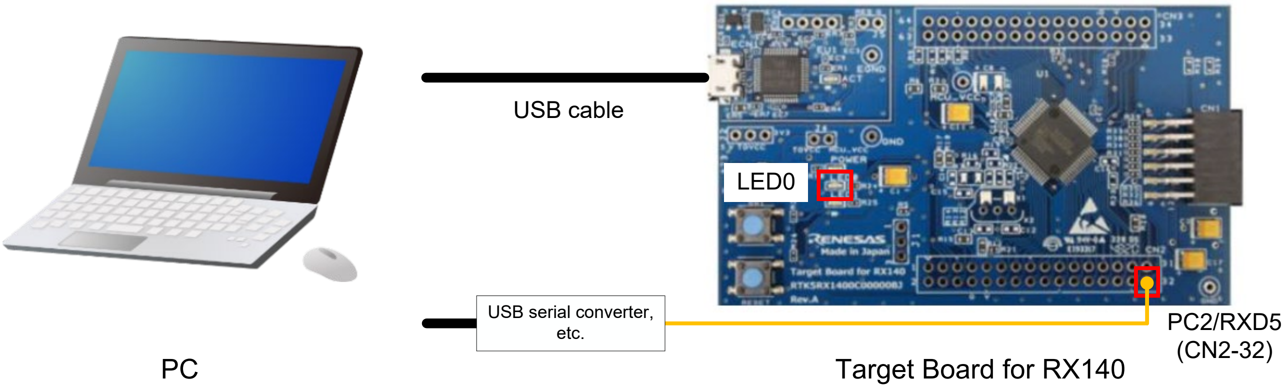


Figure 2.2 Target Board Connection Diagram

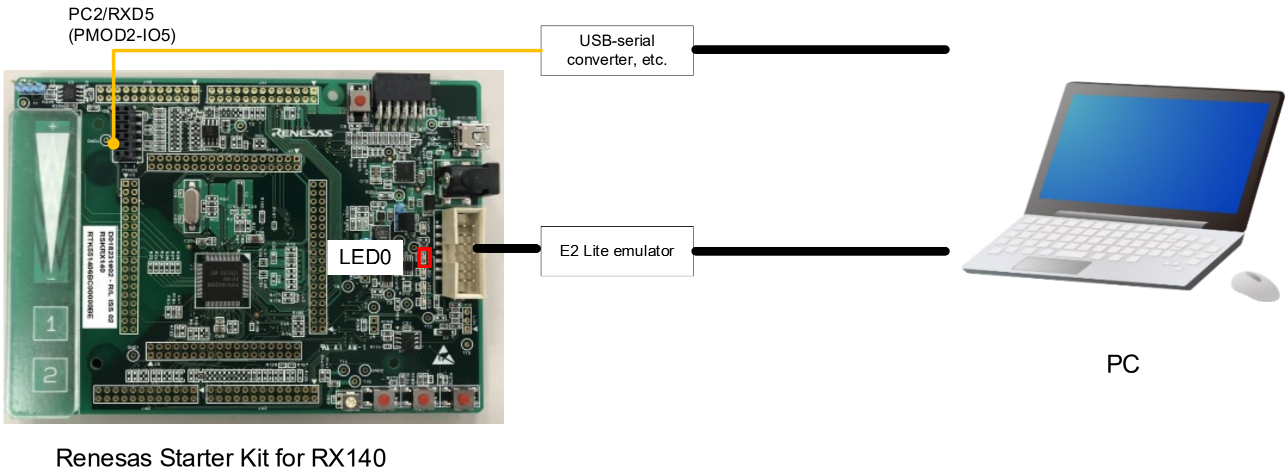


Figure 2.3 Renesas Starter Kit Connection Diagram

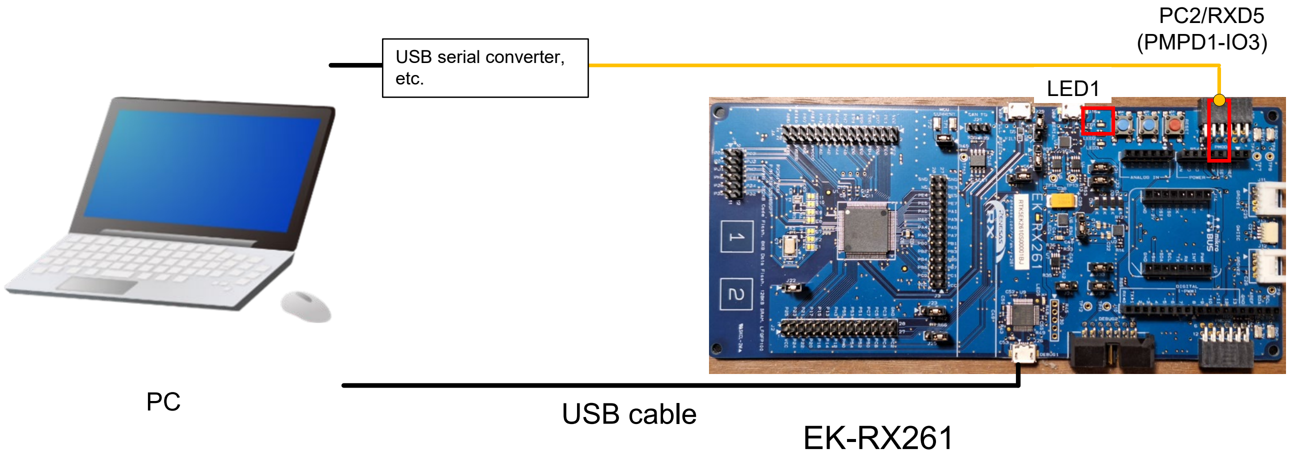


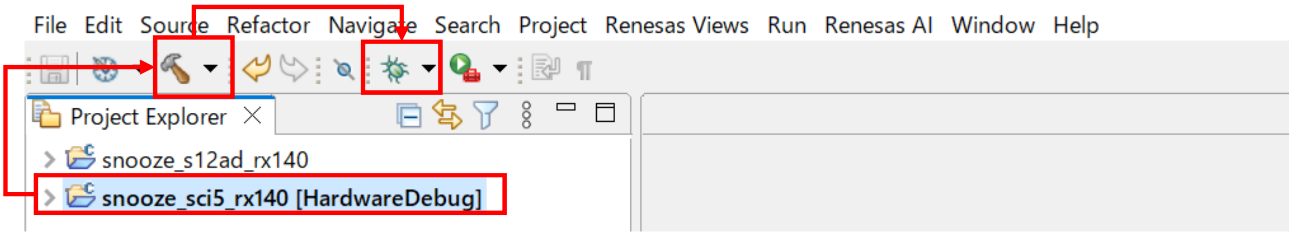
Figure 2.4 EK-RX261 Connection Diagram

2.1.3.2 Running the Sample Program

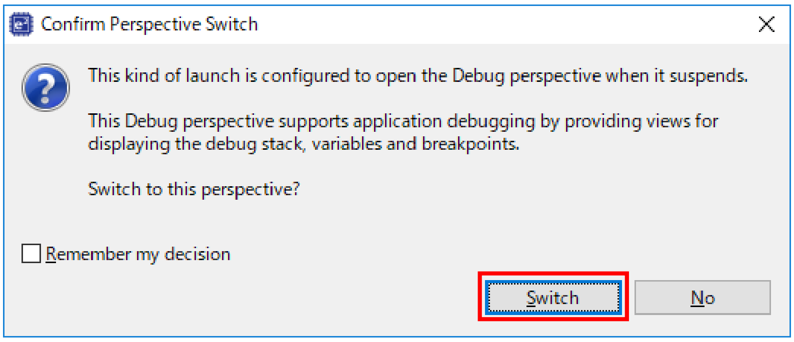
The first step is to import the sample program project. Refer to section 3, Importing a Project, for instructions.

After importing the project, select the **snooze_sci5_rx140** project under **Project Explorer**. After making the selection, click the **Build** button and wait for the build process to complete. If build fails, it could be because the compiler listed in 2.1.2, Operation Confirmation Conditions, is not installed. Check the toolchain setting in the project properties if necessary reset it to match the installed compiler.

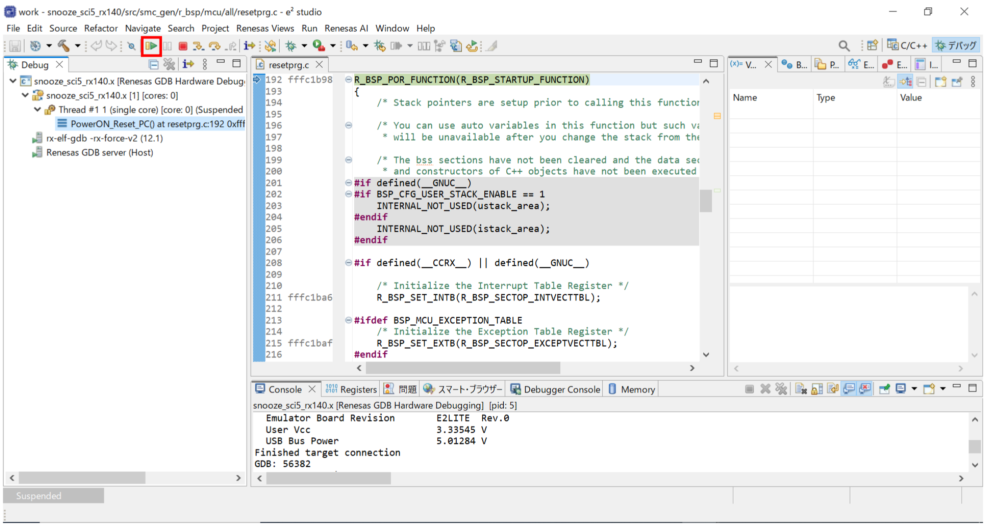
After build finishes, click the **Launch in 'Debug' mode** button to start the connection to the debugger.



Shortly after you start the connection to the debugger, the **Confirm Perspective Switch** window appears. Click the **Switch** button.

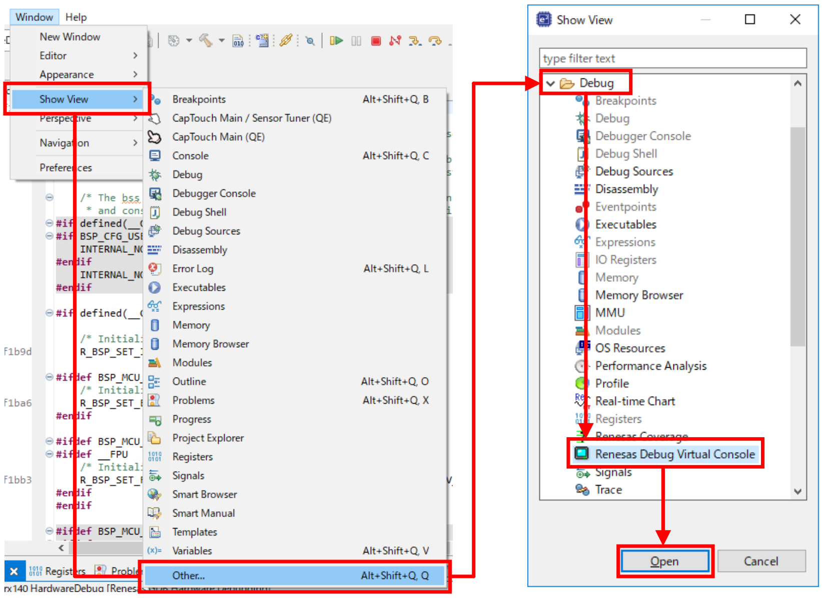


After the switch to the Debug perspective, click the **Resume** button.

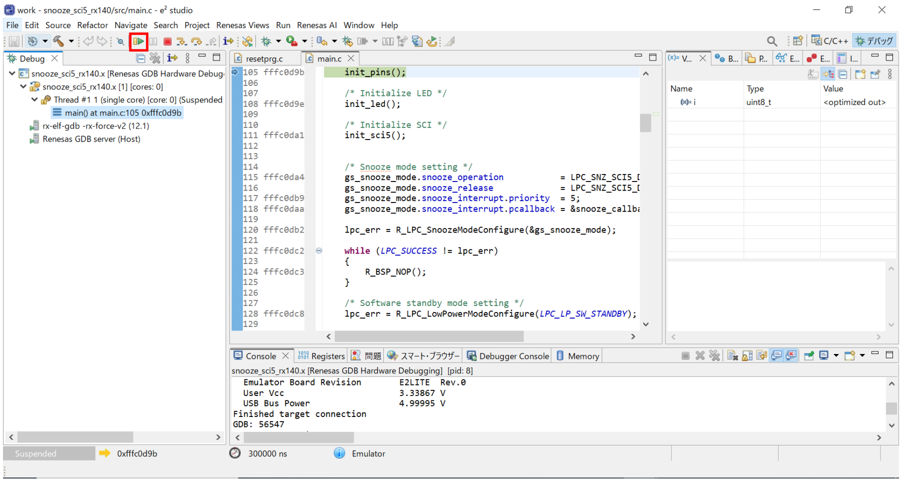


A break occurs at the start of the main function, and Renesas Debug Virtual Console opens. Select **Windows** → **Show View** → **Others…** to open the **Show View** dialog box.

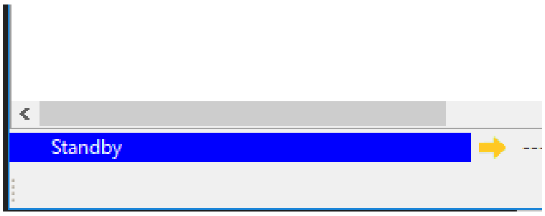
From here, select **Debug** → **Renesas Debug Virtual Console** and then click the **Open** button.



After Renesas Debug Virtual Console opens, once again click the **Resume** button to run the program.

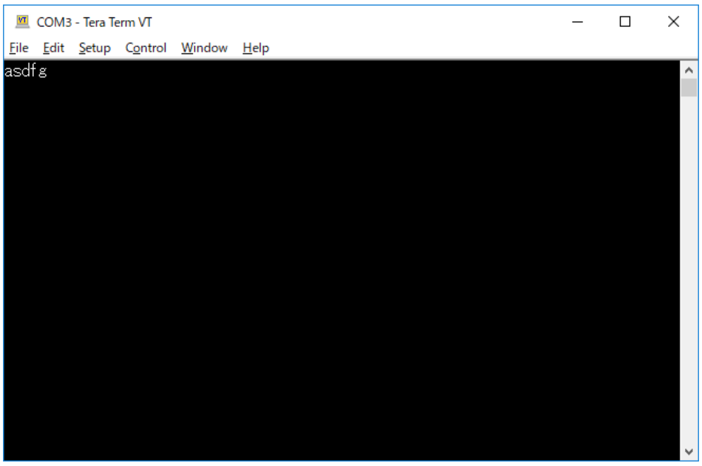


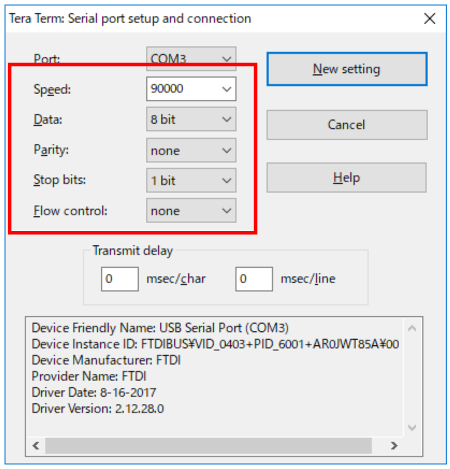
When the program runs, the indication of the status bar in the lower left corner of the window changes to **Standby**. After confirming that the status has changed to **Standby**, transmit serial data to the RXD5 pin.



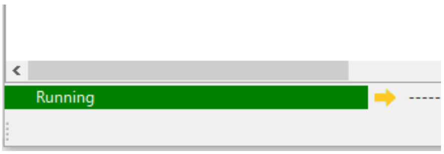
The example below illustrates using the terminal software Tera Term and a USB to serial converter module to input serial data on the RXD5 pin. When 5 bytes (5 characters) of serial data are sent from the Tera Term console, the indication of the status bar in the lower left corner of the window changes to **Running**, and the input serial data is displayed in the Renesas Debug Virtual Console.

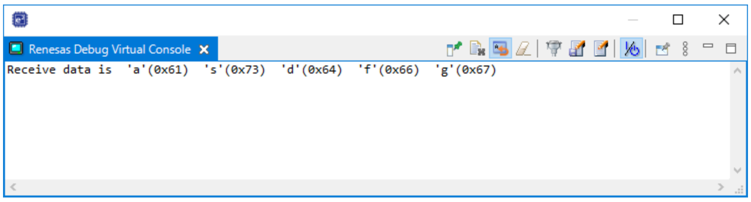
Then, after about three seconds elapse, the indication of the status bar in the lower left corner of the window changes back to **Standby**, indicating that the program is ready to receive serial data.





In addition, **Local Echo** is enabled in the terminal settings.





2.1.4 Illustrations of Current Consumption

Figure 2.5 illustrates the current consumption while the sample program is operating.

After the reset and configuration of initial settings in normal operation mode, the current consumption drops when the transition to software standby mode takes place. After the transition to snooze mode triggered by the falling edge of the signal input on the RXD5 pin, during data reception and DTC transfer operation, current consumption rises temporarily until the return to software standby mode. After the DTC transfer of the fifth byte of receive data, current consumption rises again with the return to normal operation mode.

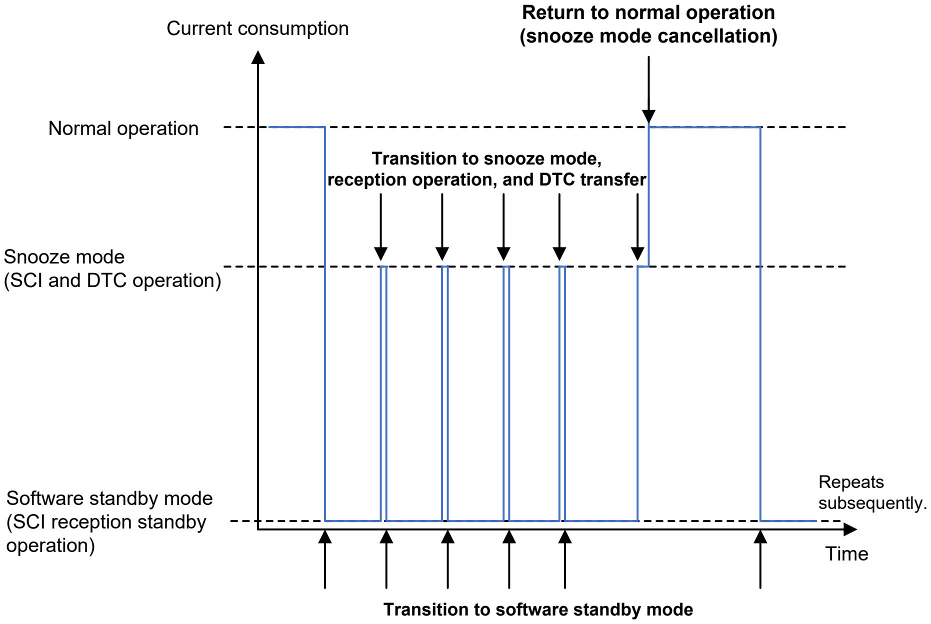


Figure 2.5 Illustration of Current Consumption during Reception Operation in SCI Asynchronous Mode

To perform SCI or DTC on earlier MCUs using low power consumption modes it was necessary use sleep mode or deep sleep mode (DTC halted). With the ability to use snooze mode on the target devices for this application note, it is now possible to reduce current consumption during periods when peripheral functions are not operating. Figure 2.6 illustrates the current consumption while the sample program is operating in sleep mode.

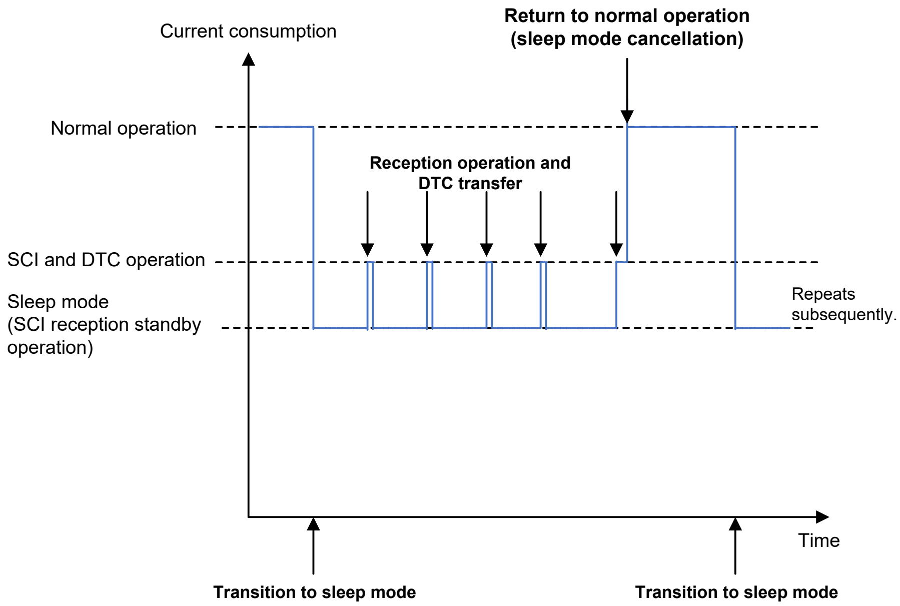


Figure 2.6 Illustration of Current Consumption during Reception Operation in SCI Asynchronous Mode (Sleep Mode)

2.1.5 Structure of Sample Program

2.1.5.1 FIT Modules Used

Table 2.6 lists the FIT modules used by the sample program.

Table 2.6 FIT Modules Used

Module	Document Title	Document No.
BSP	RX Family Board Support Package Module Using Firmware Integration Technology	R01AN1685
LPC	RX Family LPC Module Using Firmware Integration Technology	R01AN2769
SCI	RX Family SCI Module Using Firmware Integration Technology	R01AN1815
BYTEQ	RX Family BYTEQ Module Using Firmware Integration Technology	R01AN1683
DTC	RX Family DTC Module Using Firmware Integration Technology	R01AN1819

2.1.5.2 FIT Module Settings

The FIT module and e^2 studio SC settings used in the sample program are listed below. For SC settings, the items and setting details match those displayed on the setting menu. For details of the FIT modules, refer to the associated FIT module documents.

Table 2.7 BSP Module Settings

Category	Item	Setting/Description
Smart Configurator >> Components >> r_bsp		Other than the changes listed below, properties are left in the default settings.
	Heap size (BSP_CFG_HEAP_BYTES)	Changed to 0x1000.
Smart Configurator >> Clock		The following settings are made on the “Clocks” tab and reflected in r_bsp_config.h.
	VCC setting	3.3 (V)
	Using RX140(Target Board and Renesas Starter Kit)	
	Main clock settings	Stopped: Unchecked.
	PLL circuit setting	Disabled
	Sub-clock oscillator setting	Stopped: Unchecked.
	HOCO clock settings	Operation: Checked. Frequency: 48 (MHz) HOCO oscillation enabled after reset: Checked.
	LOCO clock settings	Stopped: Unchecked.
	System clock settings	Clock source: HOCO System clock (ICLK): ×1 48 (MHz) Peripheral module clock (PCLKB): ×1/2 24 (MHz) Peripheral module clock (PCLKD): ×1 48 (MHz) FlashIF clock (FCLK): ×1 48 (MHz)
	IWDT dedicated clock settings	Stopped: Unchecked.
	Using EK-RX261	
	Main clock settings	Stopped: Unchecked.
	PLL circuit setting	Disabled
	Sub-clock oscillator setting	Stopped: Unchecked.
	HOCO clock settings	Operation: Checked. Frequency: 64 (MHz) HOCO oscillation enabled after reset: Checked.
	LOCO clock settings	Stopped: Unchecked.
	System clock settings	Clock source: HOCO System clock (ICLK): ×1 64 (MHz) Peripheral module clock (PCLKB): ×1/2 32 (MHz) Peripheral module clock (PCLKD): ×1 64 (MHz) FlashIF clock (FCLK): ×1 64 (MHz)
	IWDT dedicated clock settings	Stopped: Unchecked.

Table 2.8 LPC Module Settings

Category	Item	Setting/Description
Smart Configurator >> Components >> r_lpc_rx		Default settings are used (no changes).

Table 2.9 SCI Module Settings

Category	Item	Setting/Description
Smart Configurator >> Components >> r_sci_rx		Other than the changes listed below, default settings are used.
	Include software support for channel 1 (SCI_CFG_CH1_INCLUDED)	Changed to Not (0) (channel 1 not used).
	Include software support for channel 5 (SCI_CFG_CH5_INCLUDED)	Changed to Include (1) (use channel 5).
	Use DTC/DMAC for transmit (SCI5) (SCI_CFG_CH5_TX_DTC_DMACA_ENABLE)	Changed to 1 (use DTC).
	Use DTC/DMAC for receive (SCI5) (SCI_CFG_CH5_RX_DTC_DMACA_ENABLE)	Changed to 1 (use DTC).
	Resource >> SCI	
	SCI5	Checked.
	RXD5/SMISO5/SSCL5 pin	Use: Checked.
	TXD5/SMOSI5/SSDA5 pin	Use: Checked.
Smart Configurator >> Pins >> Serial Communications Interface >> SCI5		All unchecked except for the following settings.
	RXD5	Use: Checked. Pin assignment: Set to PC2.
	TXD5	Use: Checked. Pin assignment: Set to PC3.

Table 2.10 BYTEQ Module Settings

Category	Item	Setting/Description
Smart Configurator >> Components >> r_byteq		Default settings are used (no changes).

Table 2.11 DTC Module Settings

Category	Item	Setting/Description
Smart Configurator >> Components >> r_dtc_rx		Other than the changes listed below, default settings are used.
	DMAC FIT check (DTC_CFG_USE_DMAC_FIT_MODULE)	Changed to DMAC FIT module is not used with DTC FIT module (DTC_DISABLE) (not used together with DMAC FIT module).

2.1.5.3 File Structure

Table 2.12 lists the files used by the sample program. Note that files belonging to FIT modules or generated automatically by SC are omitted.

Table 2.12 Files Used by Sample Program

File Name	Description	Remarks
main.c	This is the main processing routine of the sample program. It initializes and controls the peripheral functions, transitions to low power consumption mode, and controls operation after return.	-

2.1.5.4 Option-Setting Memory

Table 2.13 lists the option-setting memory settings used by the sample program. If necessary, modify the setting values to match your system.

Table 2.13 Option-Setting Memory Settings Used by Sample Program

Symbol	Address	Setting Value	Description
OFS0	FFFF FF8Ch to FFFF FF8Fh	FFFF FFFFh	The IWDT is halted after a reset.
OFS1	FFFF FF88h to FFFF FF8Bh	FFFF DEFFh	Using RX140 Fast startup time disabled. Voltage monitoring 0 reset after a reset disabled. HOCO oscillation after a reset enabled. The HOCO frequency is set to 48 MHz.
		FFFF CEFFh	Using RX261 Fast startup time disabled. Voltage monitoring 0 reset after a reset disabled. HOCO oscillation after a reset enabled. The HOCO frequency is set to 64 MHz.
MDE	FFFF FF80h to FFFF FF83h	FFFF FFFFh	Little endian

2.1.5.5 Variables

Table 2.14 lists the variables used by the sample program.

Table 2.14 Variables Used by Sample Program

Variable Name	Type	Description
gs_snooze_mode	static lpc_snooze_mode_t	Snooze mode setting information
gs_sci_handle	static sci_hdl_t	SCI setting information
gs_receive_data [DATA_LENGTH]	static uint8_t	Receive data storage buffer

2.1.5.6 Constants

Table 2.15 and Table 2.16 lists the constants used by the sample program.

Table 2.15 Constants Used by Sample Program (RX140)

Constant Name	Setting Value	Description
LED0	Target Board PORT3.PODR.BIT.B1	LED0 port output data storage bit
	Renesas Starter Kit PORT2.PODR.BIT.B1	
LED0_PDR	Target Board PORT3.PDR.BIT.B1	LED0 port direction control bit
	Renesas Starter Kit PORT2.PDR.BIT.B1	
LED_ON	(0)	LED0 on
LED_OFF	(1)	LED0 off
DATA_LENGTH	(5)	Reception and DTC transfer count

Table 2.16 Constants Used by Sample Program (RX261)

Constant Name	Setting Value	Description
LED1	PORTJ.PODR.BIT.B1	LED1 port output data storage bit
LED1_PDR	PORTJ.PDR.BIT.B1	LED1 port direction control bit
LED_ON	(1)	LED1 on
LED_OFF	(0)	LED1 off
DATA_LENGTH	(5)	Reception and DTC transfer count

2.1.5.7 Functions

Table 2.17 lists the functions used by the sample program.

Table 2.17 Functions Used by Sample Program

Function Name	Description
main	Main processing
init_pins	Initialization of unused pins
init_led	LED initialization
init_sci5	SCI (channel 5) and DTC initialization
activate_standby_callback	Callback function for processing immediately before transition to software standby mode [Processing details] • Enables reception and configures DTC transfer settings (runs R_SCI_Receive function). • Turns LED off.
snooze_callback	Snooze mode release interrupt callback function [Processing details] • Turns LED on. • Disables snooze mode release interrupt (runs R_LPC_SnoozeModeConfigure function).
sci_callback	SCI FIT module callback function [Processing details] • Runs NOP instruction.

2.1.5.8 Function Specifications

The functions of the sample programs are listed below.

main	
Outline	Main processing
Header	None
Declaration	void main (void)
Description	Initializes peripheral functions, transitions to low power consumption mode, configures settings after return, and displays receive data on the Debug Virtual Console.
Arguments	None
Return Value	None

init_pins	
Outline	Initialization of unused pins
Header	None
Declaration	static void init_pins(void)
Description	Initializes unused pins. It is necessary to deal with each pins appropriately. Failure to do this can increase the current consumption.
Arguments	None
Return Value	None

init_led	
Outline	LED initialization
Header	None
Declaration	static void init_led(void)
Description	Specifies the port assigned to LED.
Arguments	None
Return Value	None

init_sci5	
Outline	SCI (channel 5) and DTC initialization
Header	None
Declaration	static void init_sci5(void)
Description	Uses FIT module API functions to initialize the SCI and DTC.
Arguments	None
Return Value	None
Remarks	To link operation of the SCI FIT module and DTC FIT module, it is necessary to call the R_DTC_Open function from the user program before starting communication.

activate_standby_callback	
Outline	Callback function for processing immediately before transition to software standby mode
Header	None
Declaration	static void activate_standby_callback(void *p_data)
Description	This callback function is called by the LPC FIT module immediately before the transition to software standby mode. This function enables reception on the SCI, configures DTC transfer settings (by running the R_SCI_Receive function), and turns off LED.
Arguments	void *p_data (Not used.)
Return Value	None

snooze_callback	
Outline	Snooze mode release interrupt callback function
Header	None
Declaration	static void snooze_callback(void *p_data)
Description	This callback function is called by the LPC FIT module when a snooze mode release interrupt occurs. This function turns on LED and disables the snooze mode release interrupt (by running the R_LPC_SnoozeModeConfigure function).
Arguments	void *p_data (Not used.)
Return Value	None
Remarks	A snooze mode release interrupt is also generated in normal operation mode when the selected source occurs. To prevent snooze mode release interrupts from being generated in normal operation mode, disable the corresponding sources immediately after returning to normal operation mode.

sci_callback	
Outline	SCI FIT module callback function
Header	None
Declaration	static void sci_callback(void *p_data)
Description	This callback function is called by the SCI FIT module when an SCI interrupt occurs. This function executes a NOP instruction, but appropriate processing should be provided if necessary.
Arguments	void *p_data SCI callback event
Return Value	None

2.2 Periodic A/D Conversion Operation Using the Low-Power Timer (LPT) and 12-Bit A/D Converter (S12AD)

2.2.1 Sample Program Specifications

2.2.1.1 Software

The LPT and S12AD are used to perform periodic A/D conversion operation in snooze mode. The operation of this sample program is described in 1.6.2.2, Operation Example 2 (Snooze Mode End by DTC Transfer End and Snooze Mode Cancellation by DTC Transfer End Interrupt).

After initial settings for the peripheral functions are configured, LPT count operation starts and a transition to software standby mode takes place. One second later a transition to snooze mode occurs when an LPT compare match 1 occurs, and A/D conversion starts. After conversion ends and the data is transferred to the buffer by the DTC, a transition to software standby mode takes place. A/D conversion operation is repeated in like manner in one-second cycles until all five units of data have been converted, at which point a return to normal operation mode is triggered by the snooze mode release interrupt.

After the return to normal operation mode, the A/D-converted data is displayed in the Debug Virtual Console and approximately three seconds later a transition to software standby mode takes place.

Table 2.18 lists the peripheral functions used by the sample program.

Table 2.18 Descriptions of Peripheral Functions Used

Peripheral Function	Description
Low power consumption functions	Controls software standby mode and snooze mode.
S12AD	Performs A/D conversion.
LPT	Operates as a cycle timer while in software standby mode. Used as a snooze mode transition condition and A/D conversion start condition.
ELC	Uses LPT compare match 1 event as S12AD synchronous trigger.
DTC	Transfers A/D conversion results to the buffer.
I/O ports	Controls LED illumination. The LED turns on in normal operation mode and turns off in low power consumption mode.

The peripheral function settings are listed below.

- Low power consumption function settings
  - Low power consumption modes: Software standby mode, snooze mode
  - Snooze mode transition condition: LPT compare match 1
  - Snooze mode end condition: After DTC transfer with A/D conversion end as source
  - Snooze mode release interrupt source: DTC transfer end event at A/D conversion end interrupt

- S12AD settings
- RX140
  - A/D conversion clock (ADCLK): 48 MHz (= PCLKD)
  - Operating mode: Single scan mode
  - Conversion target: AN000 pin
  - A/D conversion start condition: Synchronous trigger (event from ELC)
  - Scan conversion time: 1.23 μs
    (conversion cycles: 32 cycles (initial register value), sampling states: 13 states (initial register value))
- RX261
  - A/D conversion clock (ADCLK): 64 MHz (= PCLKD)
  - Operating mode: Single scan mode
  - Conversion target: AN000 pin
  - A/D conversion start condition: Synchronous trigger (event from ELC)
  - Scan conversion time: 0.92 μs
    (conversion cycles: 32 cycles (initial register value), sampling states: 13 states (initial register value))

- LPT settings
  - Clock source: IWDT-dedicated LOCO (15 kHz)
  - Timer cycle: 1 second (no division, cycle setting: 14999 + 1)
  - Compare match: Compare match 0 disabled, compare match 1 enabled
  - PWM output: Disabled
- ELC settings
  - Link source and event signal: LPT compare match 1
  - Link destination and operation: S12AD A/D conversion start
- DTC settings
  - Transfer mode: Normal transfer mode
  - Transfer unit: 2 bytes
  - Transfer count: 5
  - Transfer source: A/D data register 0 (ADDR0)
  - Transfer destination: Buffer variable (See 2.2.5.5, Variables.)
  - Transfer source: Scan end interrupt
  - CPU interrupt timing: An interrupt request is issued to the CPU upon completion of the specified number of data transfers.
  - Write-back disable: Write-back of transfer information occurs upon completion of data transfer.
  - Chain transfer: Disabled
  - Sequence transfer: Disabled
- I/O port settings*[1]
  - Port 31: Initially low output, high (LED off) or low (LED on) output according to program state
  - Port 21: Initially low output, high (LED off) or low (LED on) output according to program state
  - Port J1: Initially high output, low (LED off) or high (LED on) output according to program state

Note: 1. When using the sample programs for the Target Board for RX140 board, use port 31. When using the sample programs for the Renesas Starter Kit for RX140 board, use port 21. When using the sample programs for the EK-RX261 board, use port J1.

2.2.1.2 Pins Used

Table 2.19 and Table 2.20 lists the pins and functions used by the sample program.

Table 2.19 Pins and Functions Used (RX140)

Pin Name	I/O	Description
P31	Output	LED0 (normal operation mode: on, low power consumption mode: off)
P21		
P40/AN000	Input	Analog input pin

Table 2.20 Pins and Functions Used (RX261)

Pin Name	I/O	Description
PJ1	Output	LED1 (normal operation mode: on, low power consumption mode: off)
P40/AN000	Input	Analog input pin

2.2.1.3 Outline Flowchart

Figure 2.7 is an outline flowchart of the sample program.

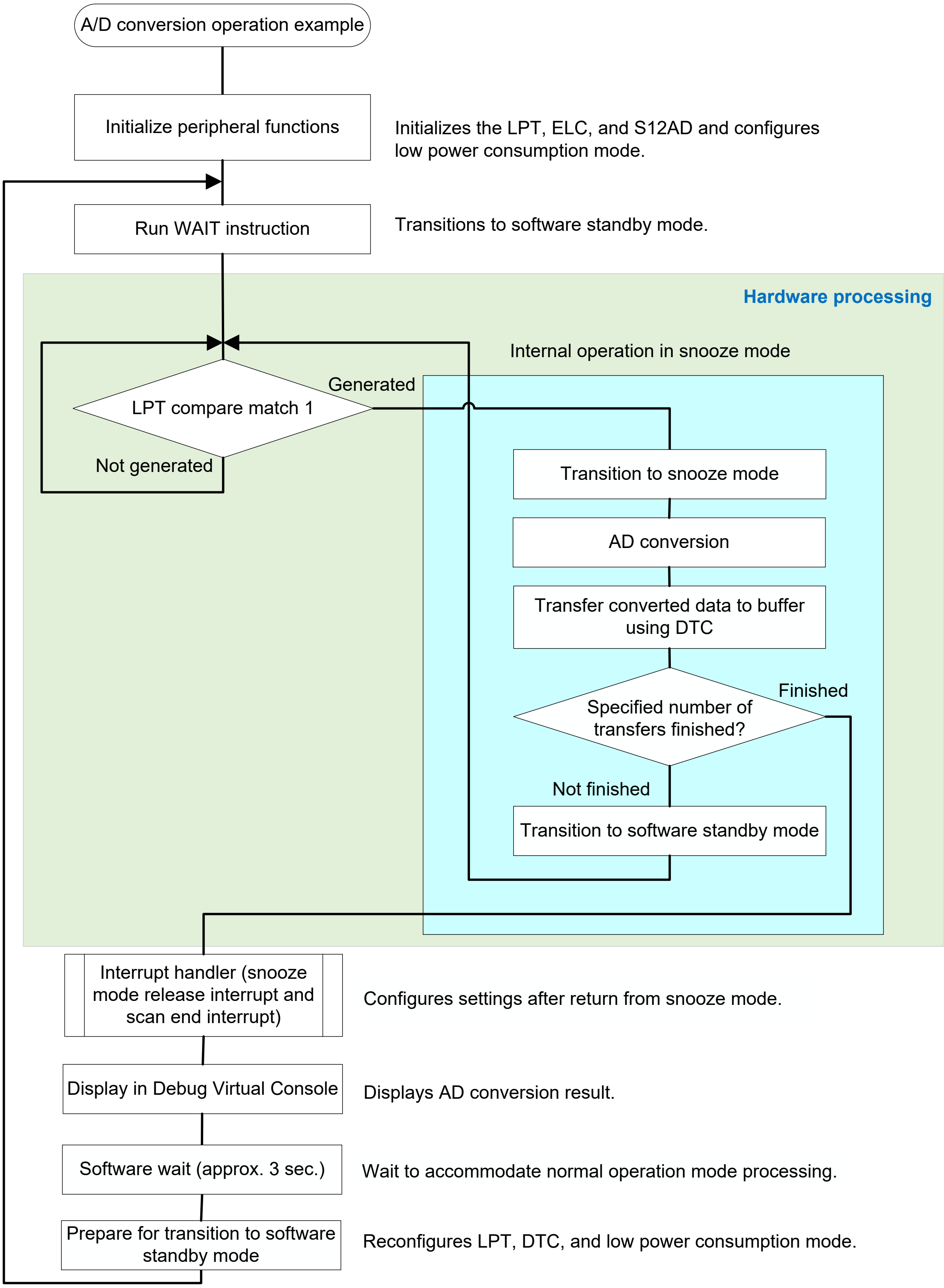


Figure 2.7 Outline Flowchart of Periodic A/D Conversion Operation Using LPT and S12AD

2.2.2 Operation Confirmation Conditions

The operation of the sample program has been confirmed under the following conditions.

Table 2.21 Operation Confirmation Conditions (Periodic A/D Conversion Operation Using LPT and S12AD)

Item	Description
MCU	R5F51403ADFM (RX140 Group) R5F51406BDFN (RX140 Group) R5F52618BGFP (RX261 Group)
Operating frequency	Target Board for RX140 • HOCO clock: 48 MHz • System clock (ICLK): 48 MHz (HOCO clock divided by 1) • Peripheral module clock B (PCLKB): 24 MHz (HOCO clock divided by 2) • Peripheral module clock D (PCLKD): 48 MHz (HOCO clock divided by 1) • FlashIF clock (FCLK): 48 MHz (HOCO clock divided by 1)
	Renesas Starter Kit for RX140 • Main clock: 8 MHz • PLL: 48 MHz (6 × main clock divided by 1) • System clock (ICLK): 48 MHz (PLL clock divided by 1) • Peripheral module clock B (PCLKB): 24 MHz (PLL clock divided by 2) • Peripheral module clock D (PCLKD): 48 MHz (PLL clock divided by 1) • FlashIF clock (FCLK): 48 MHz (PLL clock divided by 1)
	EK-RX261 • Main clock: 8 MHz • PLL: 64 MHz (8 × main clock divided by 1) • System clock (ICLK): 64 MHz (PLL clock divided by 1) • Peripheral module clock B (PCLKB): 32 MHz (PLL clock divided by 2) • Peripheral module clock D (PCLKD): 64 MHz (PLL clock divided by 1) • FlashIF clock (FCLK): 64 MHz (PLL clock divided by 1)
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics e^2 studio Version 2024-07
C compiler	Renesas Electronics C/C++ Compiler Package for RX Family V.3.06.00
	Compiler options -lang = c99
iodefine.h version	V1.10A (RX140) V1.00 (RX261)
Endian order	Little endian or big endian
Operating mode	Single-chip mode
Processor mode	Supervisor mode
Emulator	On-board emulator E2 Lite emulator
Board used	Target Board for RX140 (product No.: RTK5RX140xxxxxxxxx) Renesas Starter Kit for RX140 (product No.: RTK551406xxxxxxxxx) EK-RX261 (product No.: RTK5EK261xxxxxxxxx)

2.2.3 Confirming Operation of the Sample Program

2.2.3.1 Preparing Devices

To prepare to run the sample program, make connections as shown in Figure 2.8. When using the Target Board, adjust the input voltage for A/D conversion by connecting the appropriate resistor (or the like) to the P40 or AN000 pin. When using the Renesas Starter Kit board, you can use an on-board single-turn potentiometer to vary the analog input voltage between Board_VCC and AVSS0. For the specifications of the potentiometer, see the information provided on the manufacturer's website (manufacturer: VISHAY, model: TS53 series).

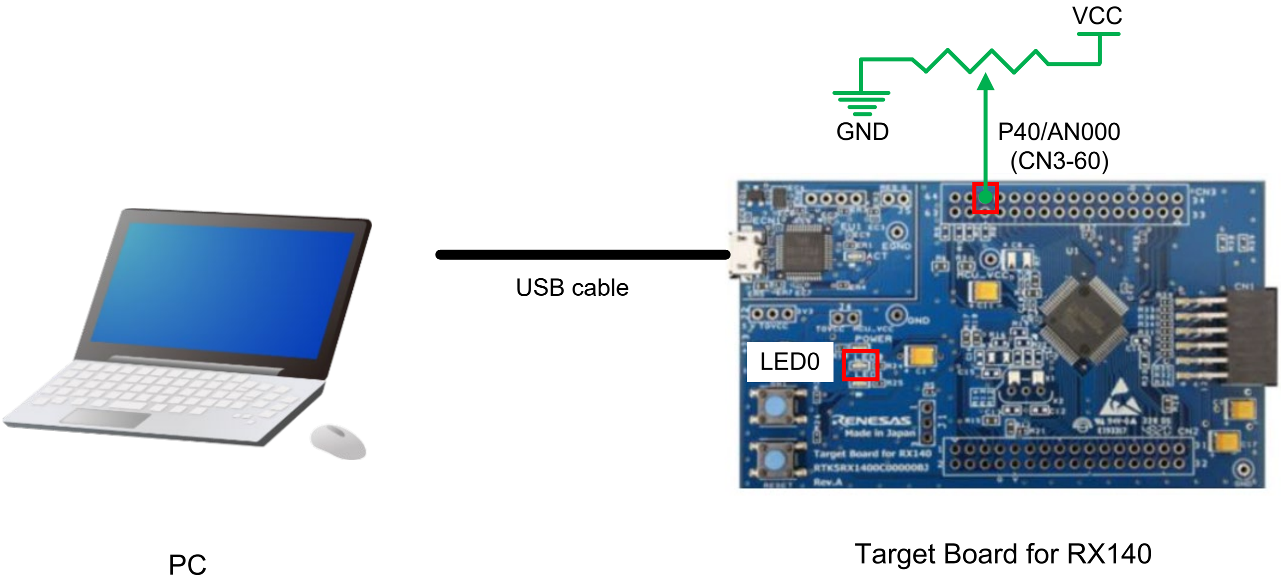


Figure 2.8 Target Board Connection Diagram

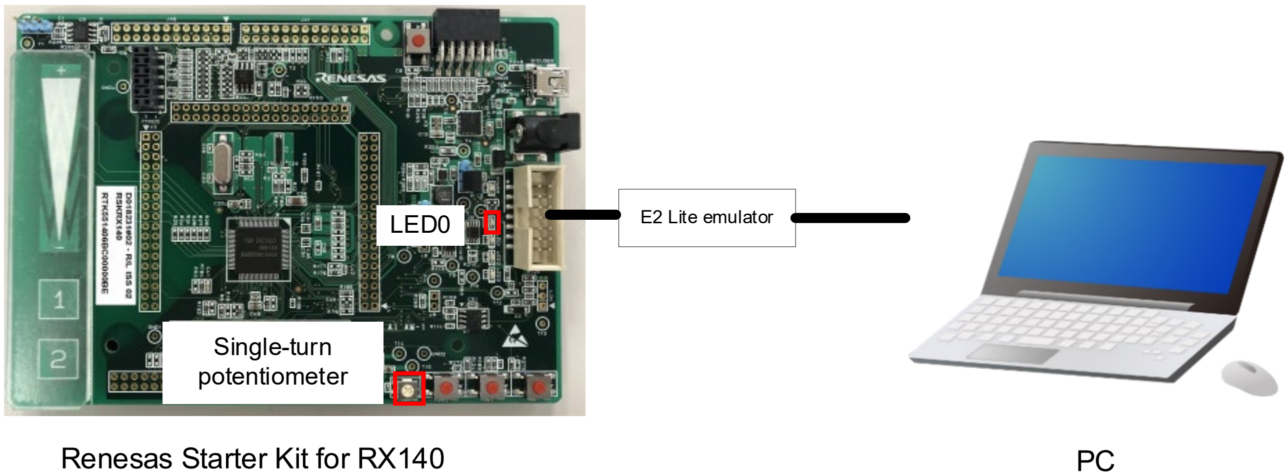


Figure 2.9 Renesas Starter Kit Connection Diagram

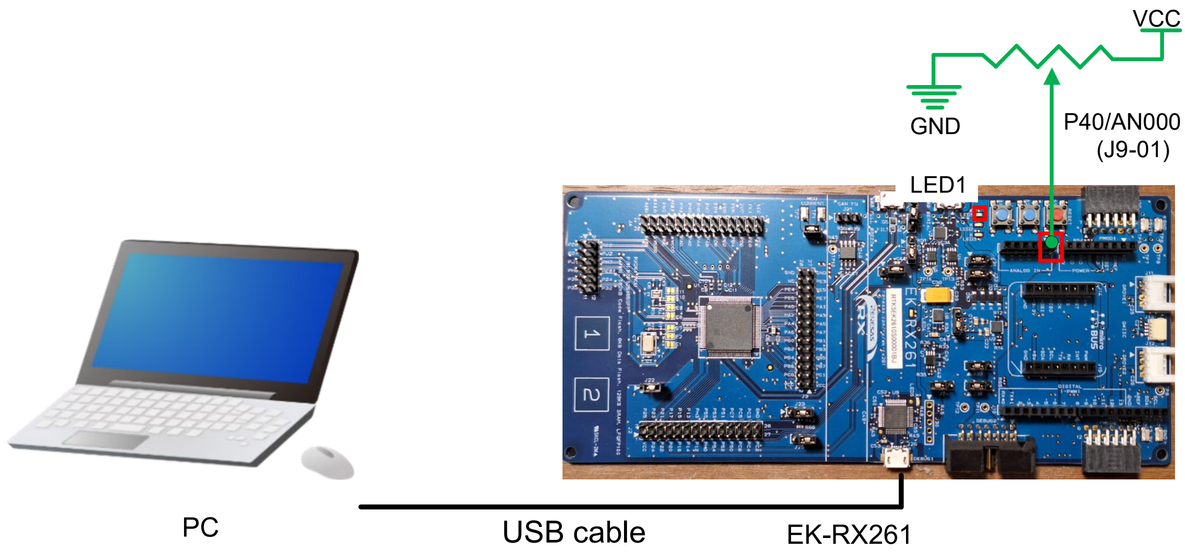


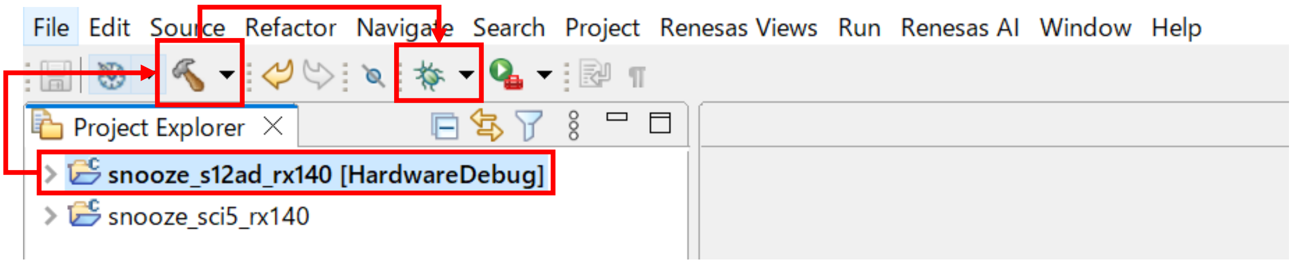
Figure 2.10 EK-RX261 Connection Diagram

2.2.3.2 Running the Sample Program

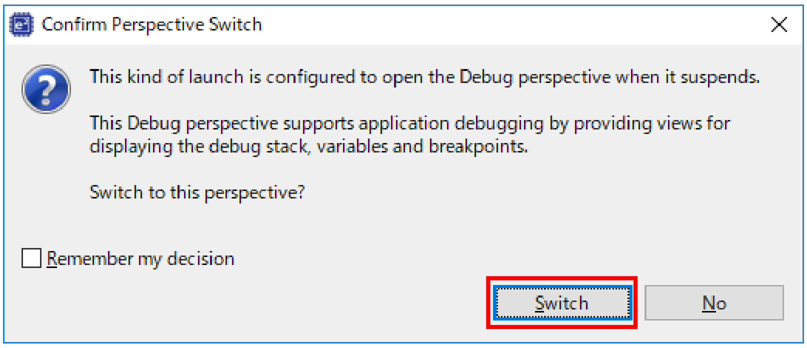
The first step is to import the sample program project. Refer to section 3, Importing a Project, for instructions.

After importing the project, select the **snooze_s12ad_rx140** project under **Project Explorer**. After making the selection, click the **Build** button and wait for the build process to complete. If build fails, it could be because the compiler listed in 2.2.2, Operation Confirmation Conditions, is not installed. Check the toolchain setting in the project properties if necessary reset it to match the installed compiler.

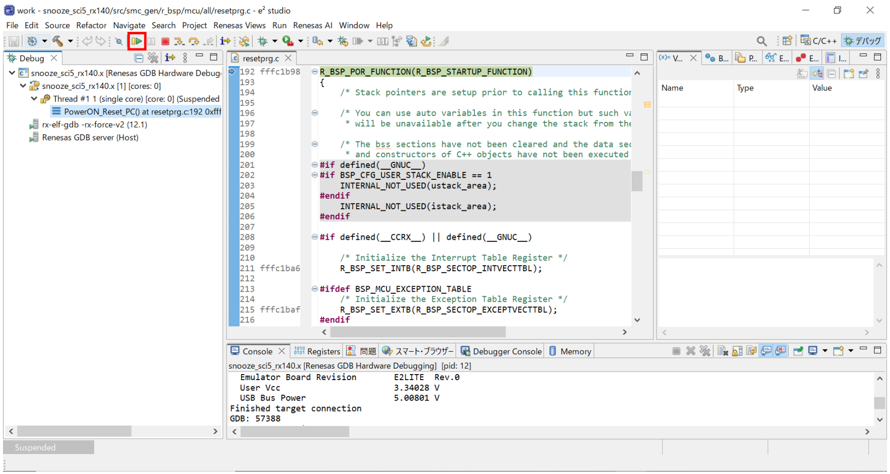
After build finishes, click the **Launch in 'Debug' mode** button to start the connection to the debugger.



Shortly after you start the connection to the debugger, the **Confirm Perspective Switch** window appears. Click the **Switch** button.

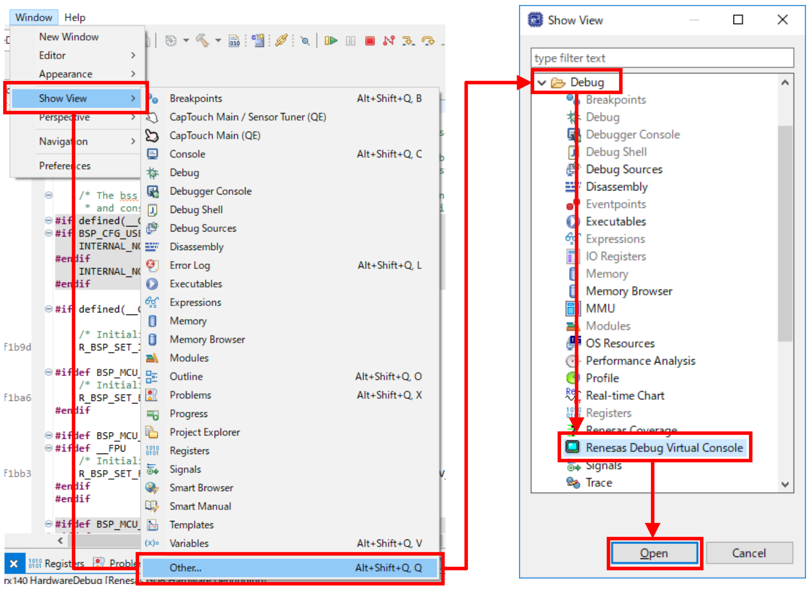


After the switch to the Debug perspective, click the **Resume** button.

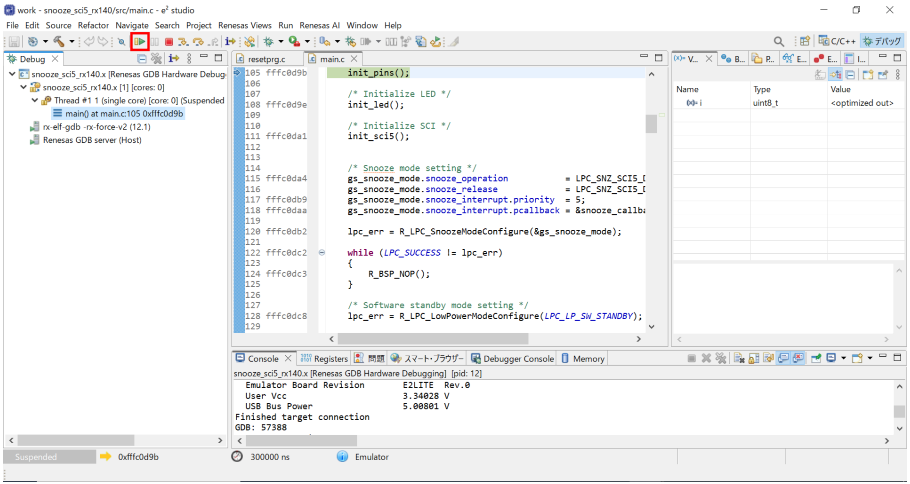


A break occurs at the start of the main function, and Renesas Debug Virtual Console opens. Select **Windows** → **Show View** → **Others…** to open the **Show View** dialog box.

From here, select **Debug** → **Renesas Debug Virtual Console** and then click the **Open** button.

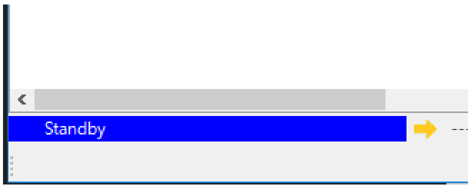


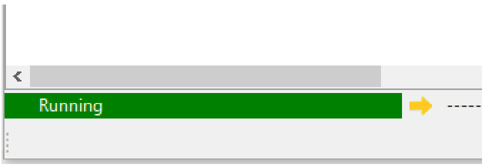
After Renesas Debug Virtual Console opens, once again click the **Resume** button to run the program.

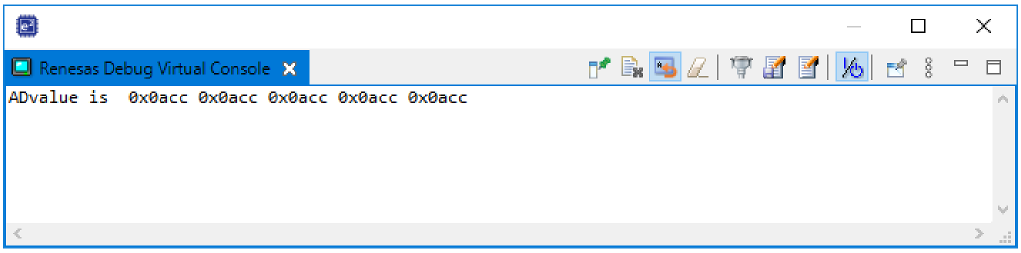


When the program runs, the indication of the status bar in the lower left corner of the window changes to **Standby**. After about five seconds, the status bar indication changes to **Running**, and five cycles' worth of A/D conversion results are displayed in the Renesas Debug Virtual Console.

Then, after about three seconds elapse, the indication of the status bar in the lower left corner of the window changes back to **Standby**. After this the operation repeats.







2.2.4 Illustrations of Current Consumption

Figure 2.11 illustrates the current consumption while the sample program is operating.

After the reset and configuration of initial settings in normal operation mode, the current consumption drops when the transition to software standby mode takes place. After the transition to snooze mode triggered by the LPT compare match 1, during A/D conversion and DTC transfer operation, current consumption rises temporarily until the return to software standby mode. After the DTC transfer of the fifth A/D conversion result, current consumption rises again with the return to normal operation mode.

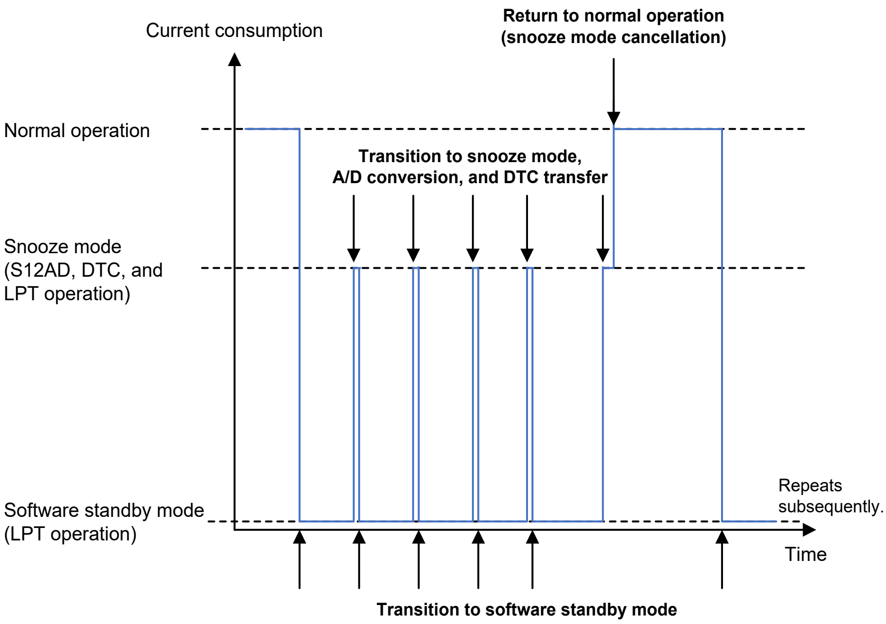


Figure 2.11 Illustration of Current Consumption during Periodic A/D Conversion Operation Using LPT and S12AD

To perform S12AD or DTC on earlier MCUs using low power consumption modes it was necessary use sleep mode or deep sleep mode (DTC halted). With the ability to use snooze mode on the target devices for this application note, it is now possible to reduce current consumption during periods when peripheral functions are not operating. Figure 2.12 illustrates the current consumption while the sample program is operating in sleep mode.

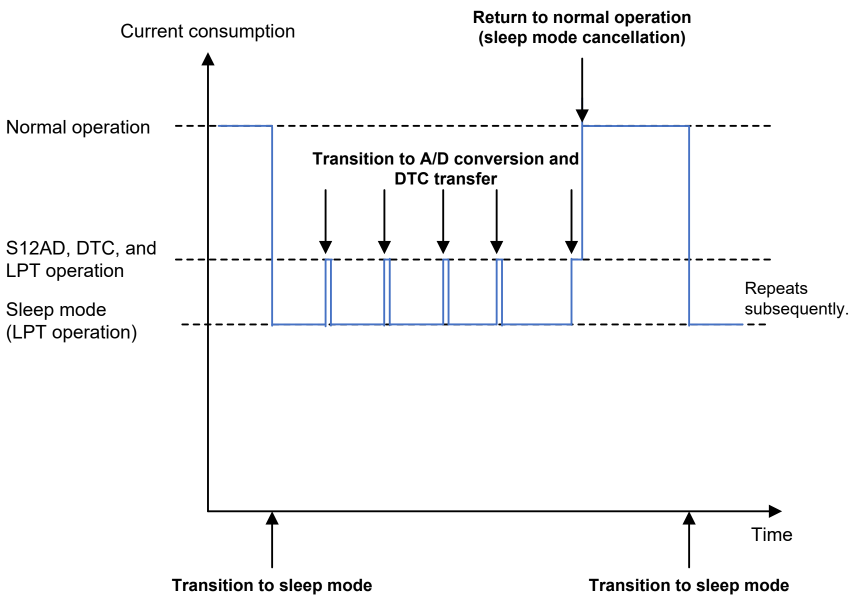


Figure 2.12 Illustration of Current Consumption during Periodic A/D Conversion Operation Using LPT and S12AD (Sleep Mode)

2.2.5 Structure of Sample Program

2.2.5.1 FIT Modules Used

Table 2.22 lists the FIT modules used by the sample program.

Table 2.22 FIT Modules Used

Module	Document Title	Document No.
BSP	RX Family Board Support Package Module Using Firmware Integration Technology	R01AN1685
LPC	RX Family LPC Module Using Firmware Integration Technology	R01AN2769
S12AD	RX Family ADC Module Using Firmware Integration Technology	R01AN1666
LPT	RX Family LPT Module Using Firmware Integration Technology	R01AN2571
ELC	RX Family ELC Module Using Firmware Integration Technology	R01AN3066
DTC	RX Family DTC Module Using Firmware Integration Technology	R01AN1819

2.2.5.2 FIT Module Settings

The FIT module and e[2] studio SC settings used in the sample program are listed below. For SC settings, the items and setting details match those displayed on the setting menu. For details of the FIT modules, refer to the associated FIT module documents.

Table 2.23 BSP Module Settings

Category	Item	Setting/Description
Smart Configurator >> Components >> r_bsp		Other than the changes listed below, properties are left in the default settings.
	Heap size (BSP_CFG_HEAP_BYTES)	Changed to 0x1000.
Smart Configurator >> Clock		The following settings are made on the “Clocks” tab and reflected in r_bsp_config.h.
	VCC setting	3.3 (V)
	Using Target Board for RX140	
	Main clock settings	Stopped: Unchecked.
	PLL circuit setting	Disabled
	Sub-clock oscillator setting	Stopped: Unchecked.
	HOCO clock settings	Operation: Checked. Frequency: 48 (MHz) HOCO oscillation enabled after reset: Checked.
	LOCO clock settings	Stopped: Unchecked.
	System clock settings	Clock source: HOCO System clock (ICLK): ×1 48 (MHz) Peripheral module clock (PCLKB): ×1/2 24 (MHz) Peripheral module clock (PCLKD): ×1 48 (MHz) FlashIF clock (FCLK): ×1 48 (MHz)
	Using Renesas Starter Kit for RX140	
	Main clock settings	Operation: Checked. Oscillation source: Resonator Frequency: 8 (MHz) Wait time: 8192 (2048μs)
	PLL circuit setting	Frequency Division: x1 Frequency Multiplication: x6
	Sub-clock oscillator setting	Stopped: Unchecked.
	HOCO clock settings	Stopped: Unchecked.
	LOCO clock settings	Stopped: Unchecked.
	System clock settings	Clock source: PLL System clock (ICLK): ×1 48 (MHz) Peripheral module clock (PCLKB): ×1/2 24 (MHz) Peripheral module clock (PCLKD): ×1 48 (MHz) FlashIF clock (FCLK): ×1 48 (MHz)
	IWDT dedicated clock settings	Operation: Checked.
	Low-power timer clock (LPTCLK) settings	IWDT-dedicated clock selected
	Using EK-RX261	
	Main clock settings	Operation: Checked. Oscillation source: Resonator Frequency: 8 (MHz) Wait time: 8192 (2048μs)
	PLL circuit setting	Frequency Division: x1 Frequency Multiplication: x8

Category	Item	Setting/Description
	Sub-clock oscillator setting	Stopped: Unchecked.
	HOCO clock settings	Stopped: Unchecked.
	LOCO clock settings	Stopped: Unchecked.
	System clock settings	Clock source: PLL System clock (ICLK): ×1 64 (MHz) Peripheral module clock (PCLKB): ×1/2 32 (MHz) Peripheral module clock (PCLKD): ×1 64 (MHz) FlashIF clock (FCLK): ×1 64 (MHz)
	IWDT dedicated clock settings	Operation: Checked.
	Low-power timer clock (LPTCLK) settings	IWDT-dedicated clock selected

Table 2.24 LPC Module Settings

Category	Item	Setting/Description
Smart Configurator >> Components >> r_lpc_rx		Default settings are used (no changes).

Table 2.25 S12AD Module Settings

Category	Item	Setting/Description
Smart Configurator >> Components >> r_s12ad_rx		Other than the changes listed below, default settings are used.
	Resource >> S12AD	
	S12AD0	Checked.
	AN000 pin	Use: Checked.

Table 2.26 LPT Module Settings

Category	Item	Setting/Description
Smart Configurator >> Components >> r_lpt_rx		Default settings are used (no changes).

Table 2.27 ELC Module Settings

Category	Item	Setting/Description
Smart Configurator >> Components >> r_elc_rx		Default settings are used (no changes).

Table 2.28 DTC Module Settings

Category	Item	Setting/Description
Smart Configurator >> Components >> r_dtc_rx		Other than the changes listed below, default settings are used.
	DTC_CFG_USE_DMAC_FIT_MODULE	Changed to DMAC FIT module is not used with DTC FIT module (DTC_DISABLE) (not used together with DMAC FIT module).

2.2.5.3 File Structure

Table 2.29 lists the files used by the sample program. Note that files belonging to FIT modules or generated automatically by SC are omitted.

Table 2.29 Files Used by Sample Program

File Name	Description	Remarks
main.c	This is the main processing routine of the sample program. It initializes and controls the peripheral functions, transitions to low power consumption mode, and controls operation after return.	

2.2.5.4 Option-Setting Memory

Table 2.30 lists the option-setting memory settings used by the sample program. If necessary, modify the setting values to match your system.

Table 2.30 Option-Setting Memory Settings Used by Sample Program

Symbol	Address	Setting Value	Description
OFS0	FFFF FF8Ch to FFFF FF8Fh	FFFF FFFFh	The IWDT is halted after a reset.
OFS1	FFFF FF88h to FFFF FF8Bh	FFFF DEFFh	Using Target Board for RX140: Fast startup time disabled. Voltage monitoring 0 reset after a reset disabled. HOCO oscillation after a reset enabled. The HOCO frequency is set to 48 MHz.
		FFFF DFFFh	Using Renesas Starter Kit for RX140: Fast startup time disabled. Voltage monitoring 0 reset after a reset disabled. HOCO oscillation after a reset disabled. The HOCO frequency is set to 48 MHz
		FFFF CFFFh	EK-RX261: Fast startup time disabled. Voltage monitoring 0 reset after a reset disabled. HOCO oscillation after a reset disabled. The HOCO frequency is set to 64 MHz
MDE	FFFF FF80h to FFFF FF83h	FFFF FFFFh	Little endian

2.2.5.5 Variables

Table 2.28 lists the variables used by the sample program.

Table 2.31 Variables Used by Sample Program

Variable Name	Type	Description
gs_snooze_mode	static lpc_snooze_mode_t	Snooze mode setting information
gs_ad_data [DATA_LENGTH]	static uint16_t	A/D conversion result storage buffer
gs_transfer_data	static dtc_transfer_data_t	Sets DTC transfer information

2.2.5.6 Constants

Table 2.32 and Table 2.33 lists the constants used by the sample program.

Table 2.32 Constants Used by Sample Program (RX140)

Constant Name	Setting Value	Description
LED0	Target Board PORT3.PODR.BIT.B1	LED0 port output data storage bit
	Renesas Starter Kit PORT2.PODR.BIT.B1	
LED0_PDR	Target Board PORT3.PDR.BIT.B1	LED0 port direction control bit
	Renesas Starter Kit PORT2.PDR.BIT.B1	
LED_ON	(0)	LED0 on
LED_OFF	(1)	LED0 off
DATA_LENGTH	(5)	A/D conversion and DTC transfer count
LPT_PERIOD	(1000000)	Sets LPT cycle to 1 second

Table 2.33 Constants Used by Sample Program (RX261)

Constant Name	Setting Value	Description
LED1	PORTJ.PODR.BIT.B1	LED1 port output data storage bit
LED1_PDR	PORTJ.PDR.BIT.B1	LED1 port direction control bit
LED_ON	(1)	LED1 on
LED_OFF	(0)	LED1 off
DATA_LENGTH	(5)	A/D conversion and DTC transfer count
LPT_PERIOD	(1000000)	Sets LPT cycle to 1 second

2.2.5.7 Functions

Table 2.34 lists the functions used by the sample program.

Table 2.34 Functions Used by Sample Program

Function Name	Description
main	Main processing
init_pins	Initialization of unused pins
init_led	LED initialization
init_s12ad	S12AD initialization
init_elc_lpt	ELC and LPT initialization
init_dtc	DTC initialization
activate_standby_callback	Callback function for processing immediately before transition to software standby mode [Processing details] • Starts LPT count operation (run R_LPT_Control function). • Turns LED off.
snooze_callback	Snooze mode release interrupt callback function [Processing details] • Turns LED on. • Stops LPT count operation and clears counter (runs R_LPT_Control function). • Disables snooze mode release interrupt (runs R_LPC_SnoozeModeConfigure function).
s12ad_callback	AD FIT module callback function [Processing details] • Runs NOP instruction.

2.2.5.8 Function Specifications

The functions of the sample programs are listed below.

main	
Outline	Main processing
Header	None
Declaration	void main (void)
Description	Initializes peripheral functions, transitions to low power consumption mode, configures settings after return, and displays A/D conversion results on the Debug Virtual Console.
Arguments	None
Return Value	None

init_pins	
Outline	Initialization of unused pins
Header	None
Declaration	static void init_pins(void)
Description	Initializes unused pins. It is necessary to deal with each pins appropriately. Failure to do this can increase the current consumption.
Arguments	None
Return Value	None

init_led	
Outline	LED initialization
Header	None
Declaration	static void init_led(void)
Description	Specifies the port assigned to LED.
Arguments	None
Return Value	None

init_s12ad	
Outline	S12AD initialization
Header	None
Declaration	static void init_s12ad (void)
Description	Uses a FIT module API function to initialize the S12AD.
Arguments	None
Return Value	None

init_elc_lpt	
Outline	ELC and LPT initialization
Header	None
Declaration	static void init_elc_lpt (void)
Description	Uses FIT module API functions to initialize the ELC and LPT.
Arguments	None
Return Value	None

init_dtc	
Outline	DTC initialization
Header	None
Declaration	static void init_dtc (void)
Description	Uses FIT module API functions to initialize the DTC.
Arguments	None
Return Value	None

activate_standby_callback	
Outline	Callback function for processing immediately before transition to software standby mode
Header	None
Declaration	static void activate_standby_callback(void *p_data)
Description	This callback function is called by the LPC FIT module immediately before the transition to software standby mode. This function starts LPT count operation (by running the R_LPT_Control function), and turns off LED.
Arguments	void *p_data (Not used.)
Return Value	None

snooze_callback	
Outline	Snooze mode release interrupt callback function
Header	None
Declaration	static void snooze_callback(void *p_data)
Description	This callback function is called by the LPC FIT module when a snooze mode release interrupt occurs. This function turns on LED, stops LPT count operation and clears the counter (by running the R_LPT_Control function), and disables the snooze mode release interrupt (by running the R_LPC_SnoozeModeConfigure function).
Arguments	void *p_data (Not used.)
Return Value	None
Remarks	A snooze mode release interrupt is also generated in normal operation mode when the selected source occurs. To prevent snooze mode release interrupts from being generated in normal operation mode, disable the corresponding sources immediately after returning to normal operation mode.

s12ad_callback	
Outline	AD FIT module callback function
Header	None
Declaration	static void s12ad_callback(void *p_data)
Description	This callback function is called by the AD FIT module when an S12AD scan end interrupt occurs. This function executes a NOP instruction, but appropriate processing should be provided if necessary.
Arguments	void *p_data S12AD callback event
Return Value	None

3. Importing a Project

The sample programs are distributed in e² studio project format. This section shows how to import a project into e² studio or CS+. After importing a project, check the build and debug settings.

3.1 Procedure in e² studio

To use sample programs in e² studio, follow the steps below to import them into e² studio. In projects managed by e2 studio, do not use space codes, multibyte characters, and symbols such as "$", "#", "%" in folder names or paths to them.

(Note that depending on the version of e² studio you are using, the interface may appear somewhat different from the screenshots below.)

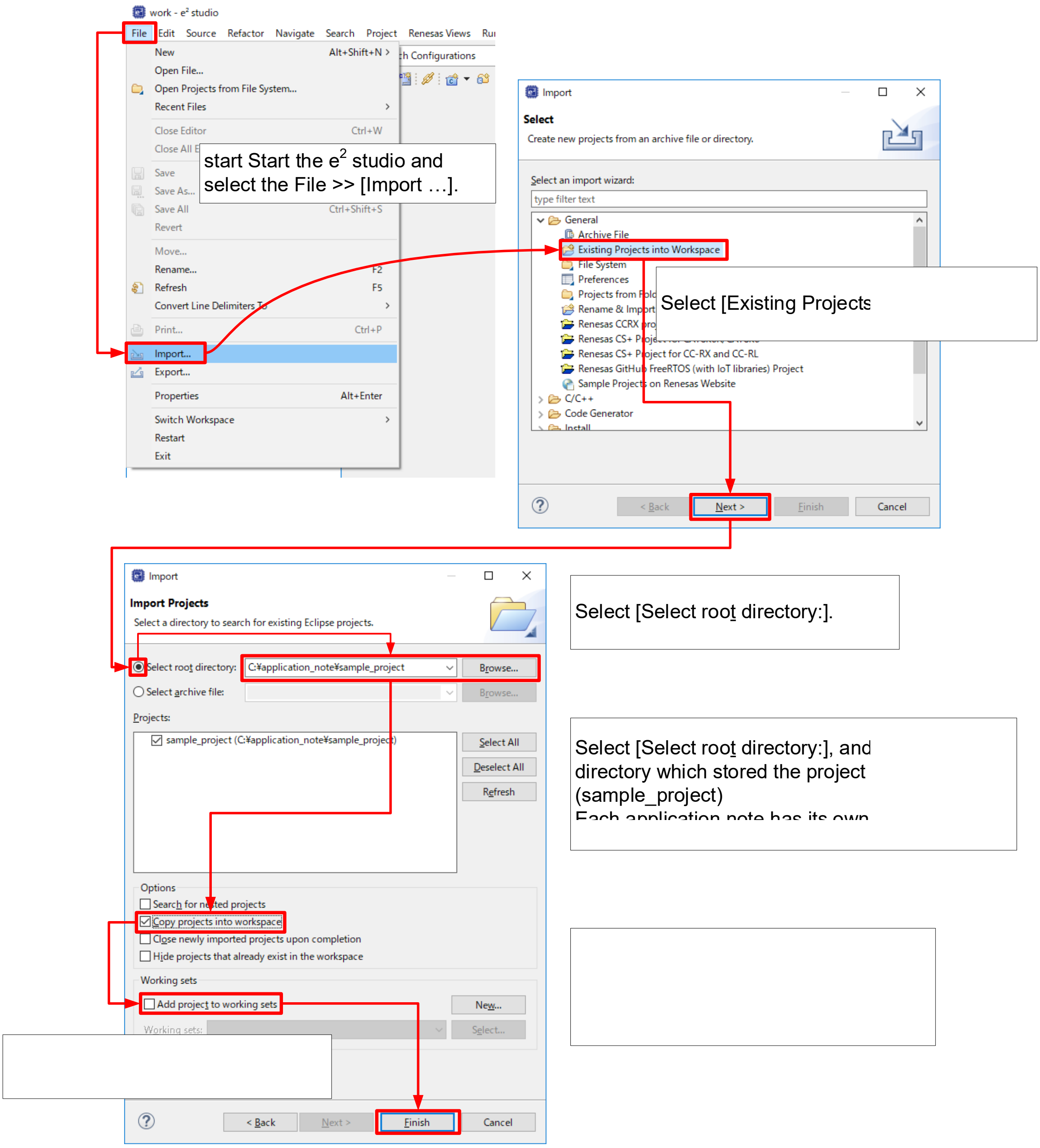


Figure 3.1 Importing a Project into e² studio

3.2 Procedure in CS+

To use sample programs in CS+, follow the steps below to import them into CS+. In projects managed by CS+, do not use space codes, multibyte characters, and symbols such as "$", "#", "%" in folder names or paths to them.

(Note that depending on the version of CS+ you are using, the interface may appear somewhat different from the screenshots below.)

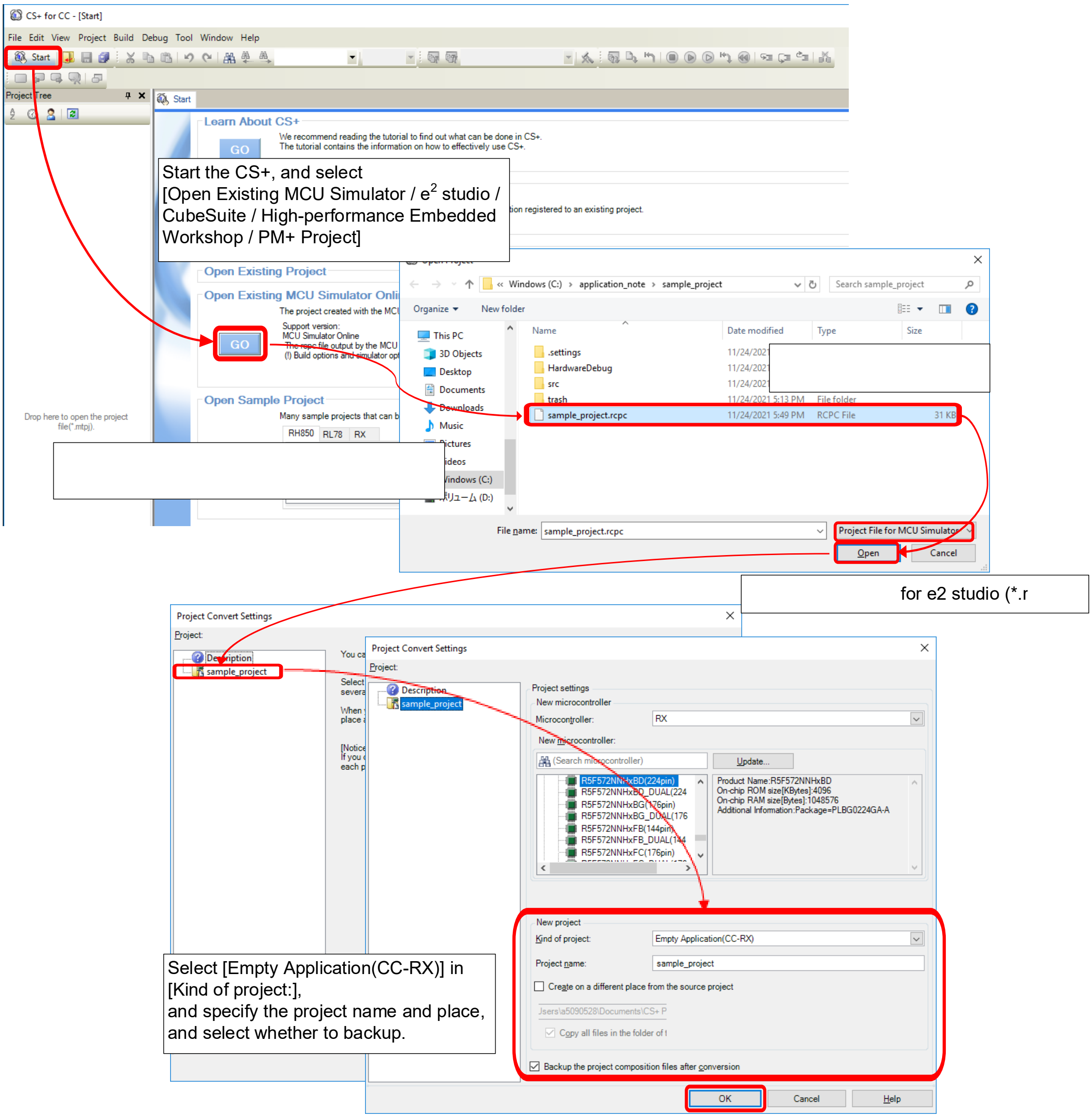


Figure 3.2 Importing a Project into CS+

4. Reference

4.1 Current Measurement Method

4.1.1 Using Target Board for RX140

In order to measure current consumption on the Target Board for RX140, it is necessary to make some modifications to the board. Table 4.1 lists the locations requiring modification.

Table 4.1 Modifications Required for the Target Board for RX140

Item	Modifications	Description
Current measurement	J4: Mount header component. SS6: Cut pattern trace.	To enable measurement of the current consumption, cut the pattern trace at SS6 and connect an ammeter to header J4.
External power supply	J2, J3: Mount header component. SS3: Cut pattern trace.	Cut the pattern trace at SS3 and connect an external power supply to header J3. To return to USB power supply, short-circuit pins 1 and 2 of header J2.
Emulator reset	J5: Mount header component.	Short-circuit header J5 to force a reset of the on-board emulator. This allows the MCU to operate alone.

For details, refer to 5.6, External Power-Supply Header, 5.8, Current Measurement Headers, and 5.13, Emulator Reset Header, in RX140 Group Target Board for RX140 User's Manual (R20UT4893).

4.1.2 Using the Renesas Starter Kit for RX140

In order to measure the current consumption on the Renesas Starter Kit for RX140, it is necessary to make some modifications to the board. Table 4.2 lists the locations requiring modification.

Table 4.2 Modifications Required for the Renesas Starter Kit for RX140

Item	Modifications	Description
Current measurement	J9: Mount header component. R268: Cut pattern trace.	To enable measurement of the current consumption, cut the pattern trace at R268 and connect an ammeter to header J9.

For details, refer to 6.4, Power Supply Configuration in the RX140 Group Renesas Starter Kit for RX140 User's Manual (R20UT5026).

4.1.3 Using EK-RX261

In order to measure current consumption on the EK-RX261, it is necessary to make some modifications to the board. Table 4.3 lists the locations requiring modification.

Table 4.3 Modifications Required for the EK-RX261

Item	Modifications	Description
Current measurement	J5: Remove header component. J6: Remove header component.	To enable measurement of the current consumption, remove the header J5 and J6 and connect an ammeter to TP1 and TP2.
Emulator reset	J24: Mount header component.	Short-circuit header J24 to force a reset of the on-board emulator. This allows the MCU to operate alone.

For details, refer to 7.2, MCU Current Measurement, and 7.3, MCU Operating Modes, in RX261 Group Evaluation Kit for RX261 Microcontroller Group EK-RX261 User's Manual (R20UT5351).

4.2 Notes on Low-Power Timer (LPT)

The LPT is the peripheral function most commonly used in snooze mode. The following notes apply whenever it is used, and they should be kept in mind when using the LPT as a cycle timer or PWM output timer.

- The LPT registers are covered by the register write protection setting. Therefore, it is necessary to cancel write protection by means of the PRC2 bit in the protect register (PRCR) before writing to the LPCMR0 register with a DTC transfer. In addition, when using FIT modules, the R_BSP_RegisterProtectDisable and R_BSP_RegisterProtectEnable API functions of the BSP FIT module should be used to disable and enable write protection, respectively. This will prevent the protect setting from being overwritten by an API function call to the LPT FIT module.
- In order to generate the LPT compare match 1 interrupt, it is necessary to enable interrupts in the interrupt controller and also to clear the ELC's module stop setting bit (MSTPCRB.MSTPB9) to 0 (module stop state disabled).
- The LPT FIT module does not set the IEN bit for compare match 1. It is therefore necessary to set the IEN bit by means of a user program in order to use LPT compare match 1 to trigger DTC transfers or interrupts.

4.3 Sample Program Consumption Current

4.3.1 Hardware and Software Used for Measurement

Table 4.4 lists the hardware and software items that were used to measure the consumption current of the sample program.

Table 4.4 List of Hardware and Software Items Used

Category	Item	Description
Digital multimeter	Manufacturer: Keithley Model: DMM7510	Measures the consumption current.
Stabilized power supply × 2	Manufacturer: KENWOOD Model: PA18-3A	Supplies power to the board.
Software	Manufacturer: Keithley Model: KickStart	Obtains the consumption current measurement results from the DMM7510 and outputs them to a log file.

4.3.2 Measurement Settings

Figure 4.1 shows the settings specified in Keithley KickStart.

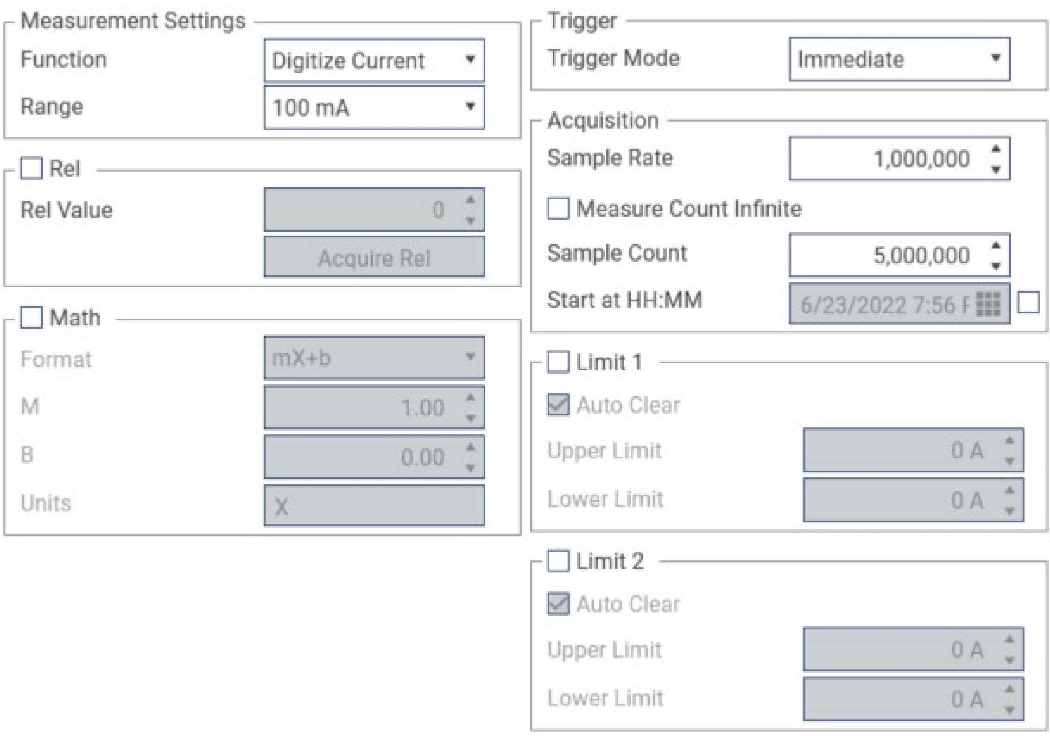


Figure 4.1 Consumption Current Measurement Settings in Keithley KickStart

4.3.3 Measurement Environment

Figure 4.2 shows the consumption current measurement environment.

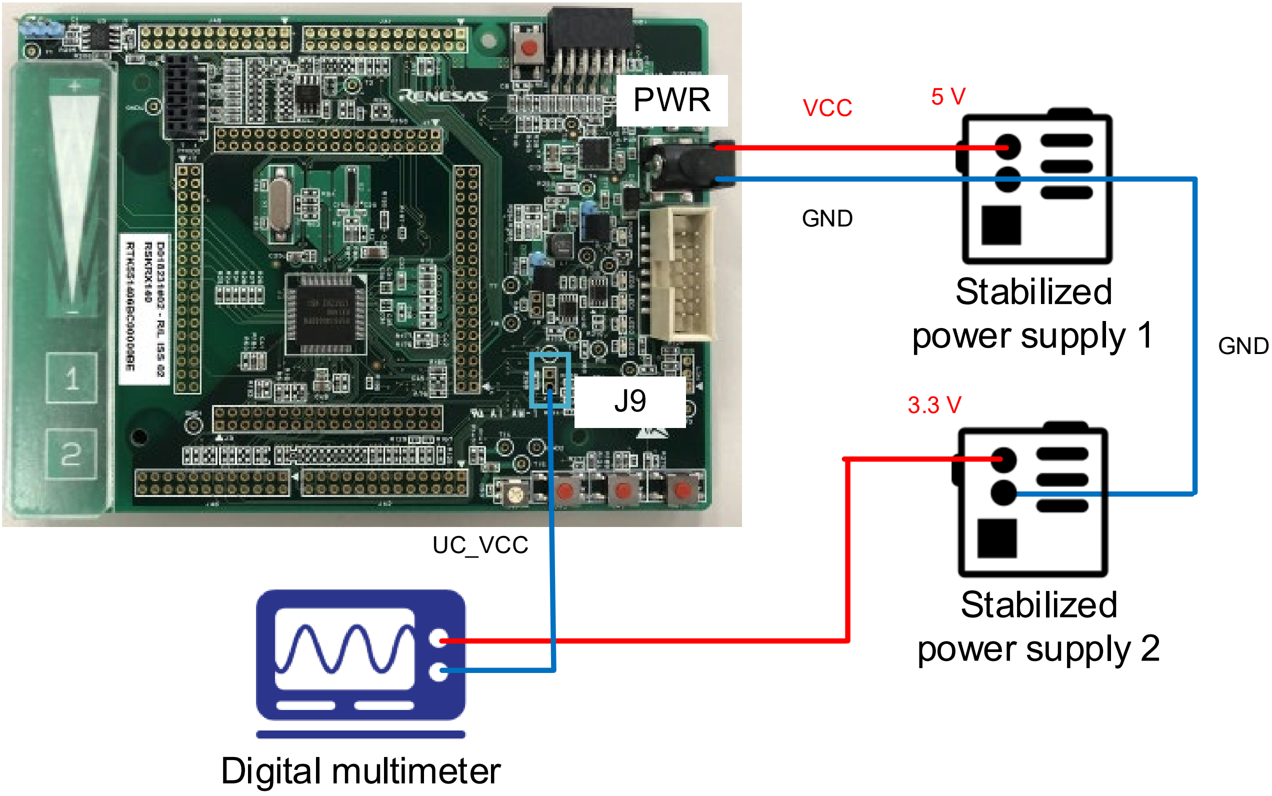


Figure 4.2 Consumption Current Measurement Environment for the Renesas Starter Kit

4.3.4 Consumption Current Measurement Results

This section shows the actual consumption current measurement results of the sample program for reference. The current values shown in this section are not nominal values. The information in this section does not guarantee the values of currents consumed during operation in snooze mode.

Figure 4.3 shows the consumption current waveform for a sequence of operations in transition from software standby mode to snooze mode (for SCI operations). Figure 4.4 shows a magnified image of the consumption current waveform for the first reception operation. Figure 4.5 shows the waveform in sleep mode (for SCI operations).

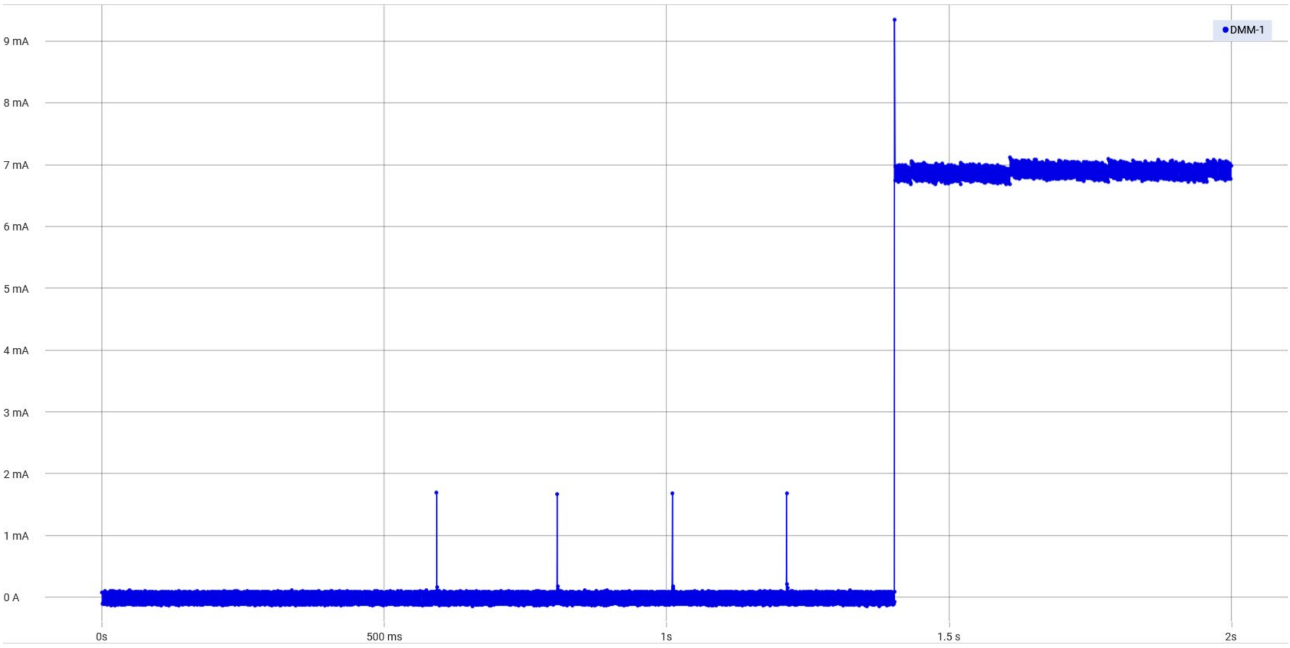


Figure 4.3 Current Value in Software Standby Mode and Snooze Mode (for SCI Operations)

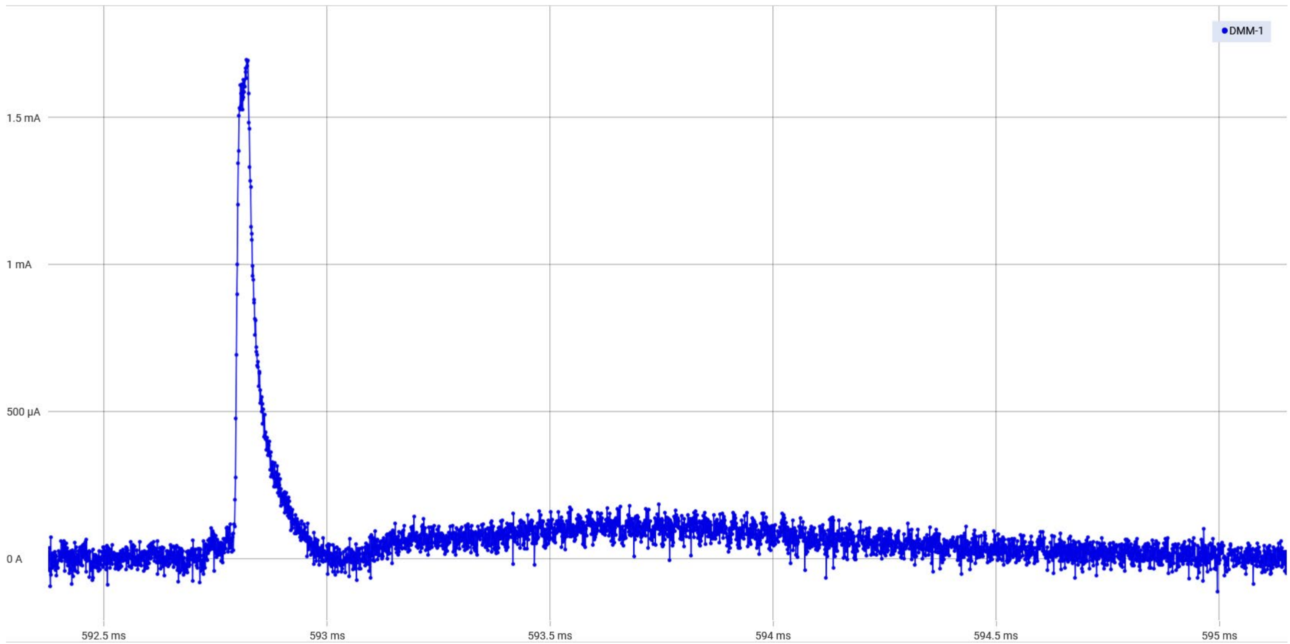


Figure 4.4 Current Value in Snooze Mode (for SCI Operations) (Magnified Image for the 1st Reception Operation)

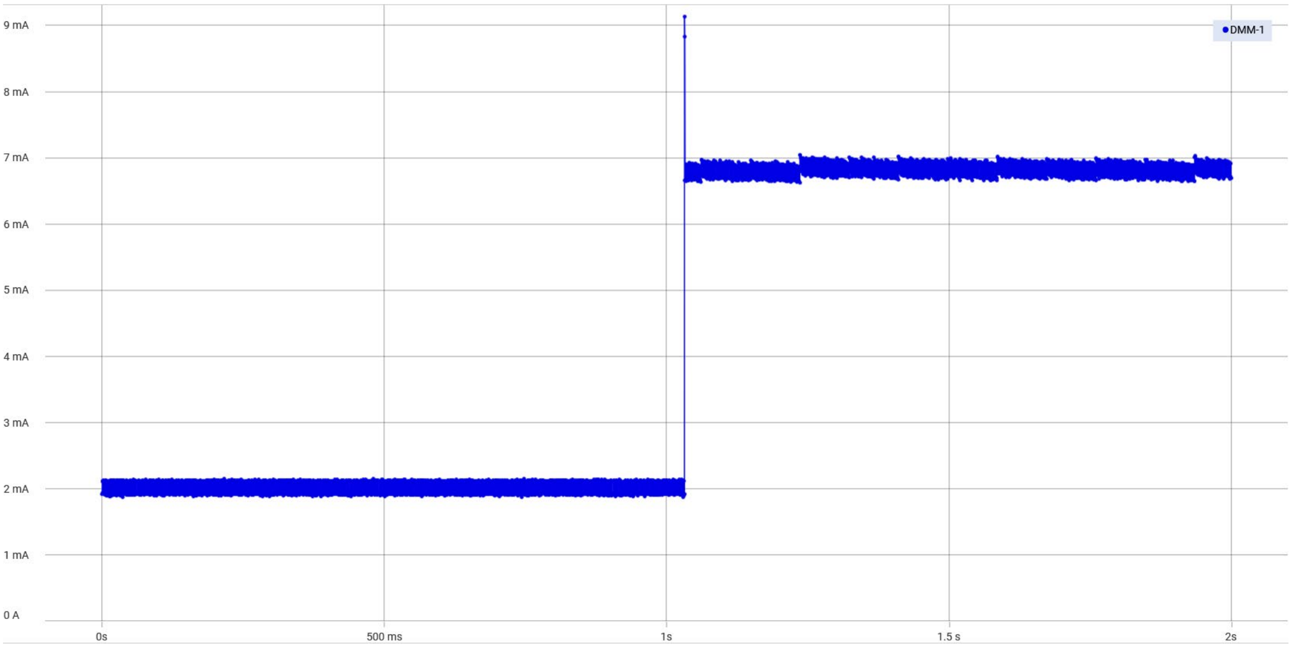


Figure 4.5 Sleep Mode (for SCI Operations)

Table 4.5 shows the actual consumption current measurement results of the sample program that uses the SCI. See also Table 4.6 to compare these results with the actual consumption current measurement results in sleep mode.

Table 4.5 Current Value for snooze_sci5_rx140

Operation Mode	Actual Measured Value (Average)	Measurement Period
Software standby mode	About 0.25 μA/ms	Time period during which the board is in software standby mode
Snooze mode (when the SCI is operating)	About 84 μA/ms (About 1.7 mA, max.)	Approximately 2.5 ms time period before the current value becomes stable after the operation mode changes from snooze mode back to software standby mode

Table 4.6 Current Value of a Reception Operation in Sleep Mode and Asynchronous Mode on the Serial Communications Interface (SCI)

Operation Mode	Actual Measured Value (Average)	Measurement Period
Sleep mode	About 2.15 mA/ms	Time period during which the board is in sleep mode

In sleep mode, because the peripheral functions are already operating, the board is continuously operating at a current of about 2.15 mA with little variation. If snooze mode is used, when the SCI begins to operate and the operation mode changes from software standby mode to snooze mode, the consumption current temporarily rises from 0.25 μA to a maximum of about 1.7 mA. The operation mode then changes back to software standby mode.

The sample program allows the user to receive data at any time. The following figure shows the average current consumed when the SCI performs a reception operation at 100 ms intervals.

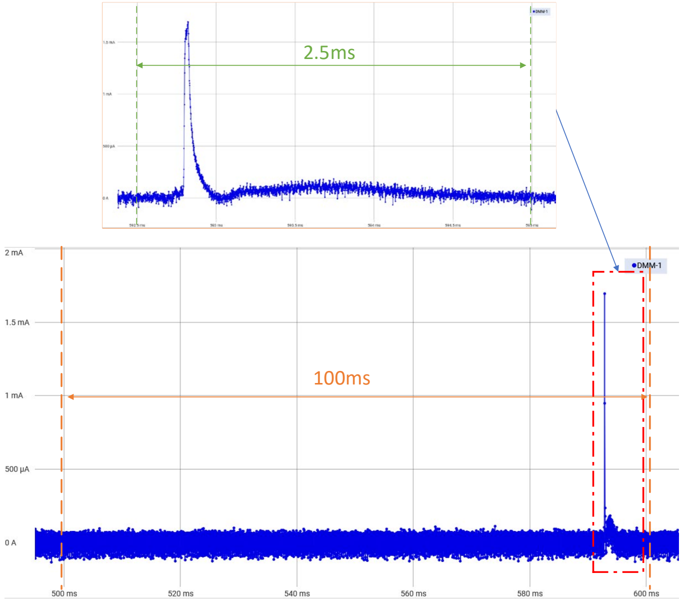


Figure 4.6 Example of Periodic Reception Operations by the SCI

Table 4.7 Average Consumption Current for Periodic Reception Operations by the SCI

Operation Mode	**Period**	**Consumption Current**
Software standby mode	100 - 2.5 = 97.5 ms	0.25 µA/ms × 97.5 ≈ 24.375 µA
Snooze mode (when the SCI is operating)	2.5 ms	84 µA/ms × 2.5 ≈ 210 µA
Total	100 ms	234.375 µA
Average consumption value for periodic reception operations (100 ms) = 234.375 µA / 100 ms ≈ 2.34 µA		

5. Obtaining the Development Environment

5.1 e² studio

Visit the following URL and download e² studio.

https://www.renesas.com/jp/en/software-tool/e-studio

This document assumes that version 2022-04 or later of e² studio is used. If a version earlier than 2022-04 is used, some e² studio functions may not be supported. Make sure to download the latest version of e² studio on the website.

5.2 Compiler Package

Visit the following URL and download the RX Family C/C++ Compiler Package.

https://www.renesas.com/jp/en/software-tool/cc-compiler-package-rx-family

6. Additional Information

6.1 Notes on Using the Evaluation Version of C/C++ Compiler Package for RX Family

The evaluation version of C/C++ Compiler Package for RX Family can only be used for a limited duration and other usage limitations apply. When the evaluation period expires, the size of linkable objects is reduced to 128 KB or less, which may cause incorrect generation of the load module.

For details, refer to the following software tool page for evaluation versions on the Renesas website:

https://www.renesas.com/jp/en/software-tool/evaluation-software-tools

7. Reference Documents

- RX140 Group: User's Manual: Hardware (R01UH0905)
- RX261 Group: User's Manual: Hardware (R01UH1045)
- RX Family: Board Support Package Module Using Firmware Integration Technology (R01AN1685)
- RX Family: LPC Module Using Firmware Integration Technology (R01AN2769)
- RX Family: ADC Module Using Firmware Integration Technology (R01AN1666)
- RX Family: LPT Module Using Firmware Integration Technology (R01AN2571)
- RX Family: ELC Module Using Firmware Integration Technology (R01AN3066)
- RX Family: DTC Module Using Firmware Integration Technology (R01AN1819)
- RX Family: SCI Module Using Firmware Integration Technology (R01AN1815)
- RX Family: BYTEQ Module Using Firmware Integration Technology (R01AN1683)
- RX Smart Configurator: User's Guide: e² studio (R20AN0451)
- Target Board for RX140 User's Manual (R20UT4893)
- Target Board for RX140 Schematic (R20UT4897)
- Renesas Starter Kit for RX140 User's Manual (R20UT5026)
- Renesas Starter Kit for RX140 CPU Board Schematics (R20UT5025)
- Evaluation Kit for RX261 Microcontroller Group EK-RX261 User's Manual(R20UT5351)
- EK-RX261 - Design Package

The latest version can be downloaded from the Renesas Electronics website.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Dec. 3. 21	—	First edition issued
1.10	Aug. 26. 22	—	Support for Renesas Starter Kit for RX140 was provided.
		60-64	"4.3 Sample Program Consumption Current" was added.
1.20	Oct. 15. 24	—	Support for EK-RX261 was provided.

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 Electrostatic 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. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
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.

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

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/.