RL78/G13

R20AN0399EJ0100
Rev.1.00
Feb 05, 2016

Usage Example of the Code Generator (Sample Program)

Abstract

This document describes using the sample program with the Code Generator for e^2 studio. The CC-RL is used as a compiler in the sample program.

Settings for MCU peripheral functions in the sample program are generated by the Code Generator. The sample program can operate on the e^2 studio simulator, so the operation of the peripheral functions can be confirmed without having the actual MCU.

Target Device

RL78/G13

Contents

1. Using the Sample Program

1.1 Development Environment

The sample program is operated under the following environment.

- Integrated development environment
  e^2 studio Version. 4.3.0.007
- Compiler
  CCRL v1.02.00
- Code Generator
  Code Generator for RL78/G13 V2.03.02.01 [15 May 2015]

1.2 Importing a Project

Start e[2] studio and import the sample program.

Select File >> Import (Figure 1).

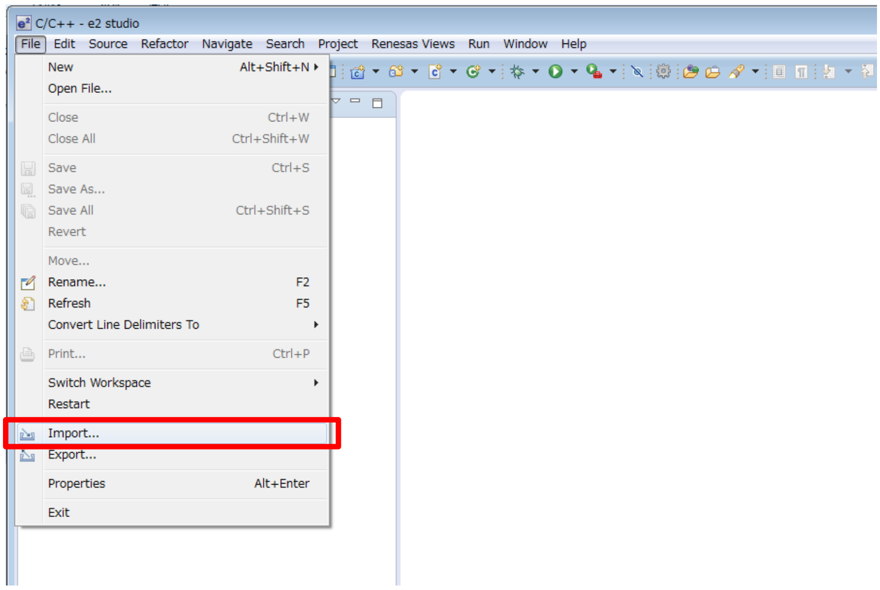


Figure 1 File Menu

The Import dialog box opens. Select 'Existing Projects into Workspace' and click the Next button (Figure 2).

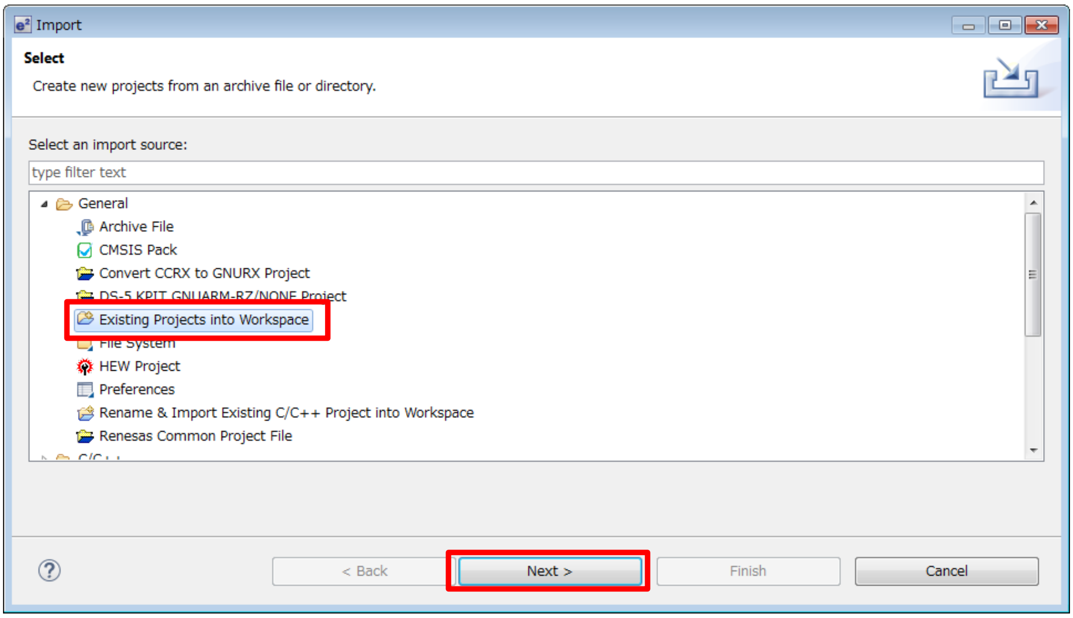


Figure 2 Import Dialog Box 1/2

Click the Browse button and select the folder where the sample program is stored.

The project will be displayed in the Projects section. Confirm that the project is ticked and click the Finish button (Figure 3).

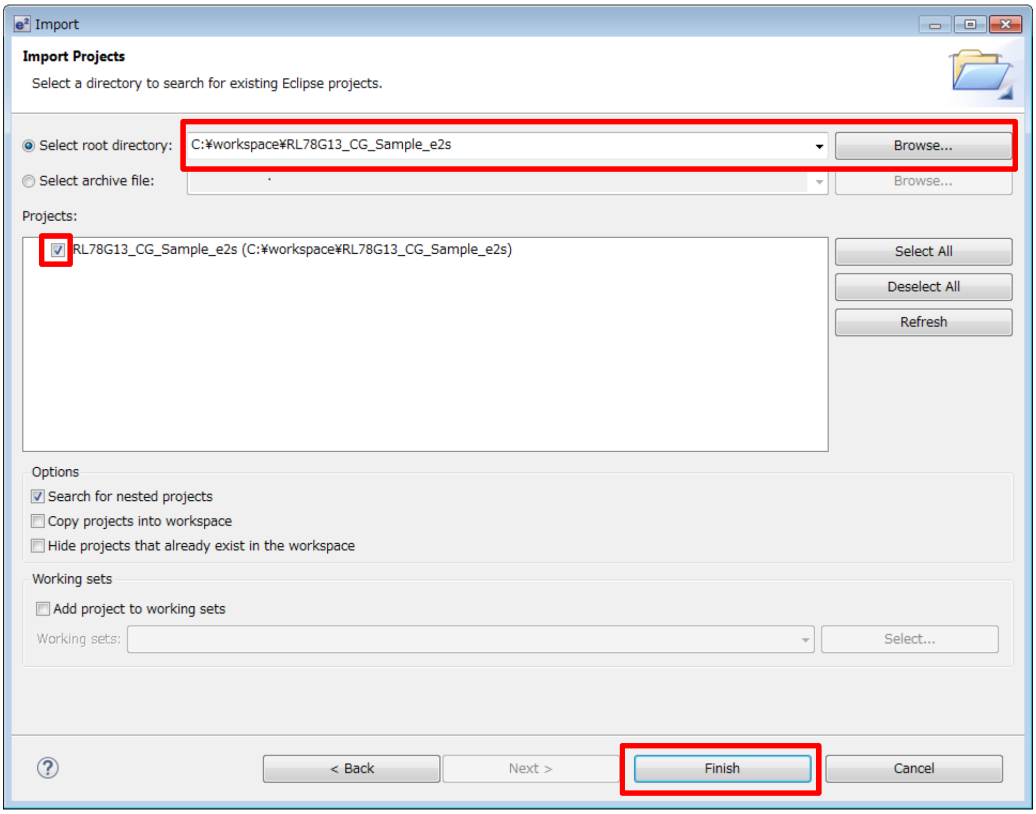


Figure 3 Import Dialog Box 2/2

The imported project is displayed in the Project Explorer (Figure 4).

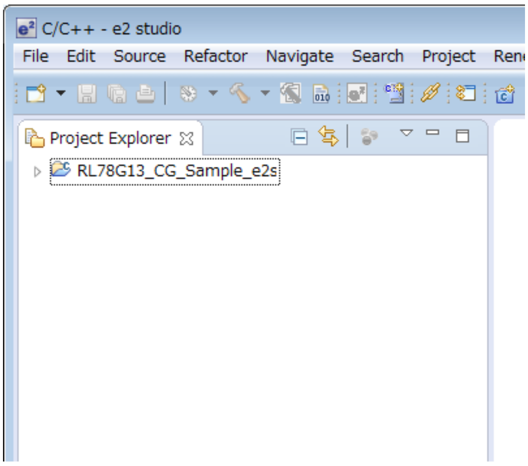


Figure 4 Project Explorer

1.3 Building the Project

Select Project >> Build Project (Figure 5).

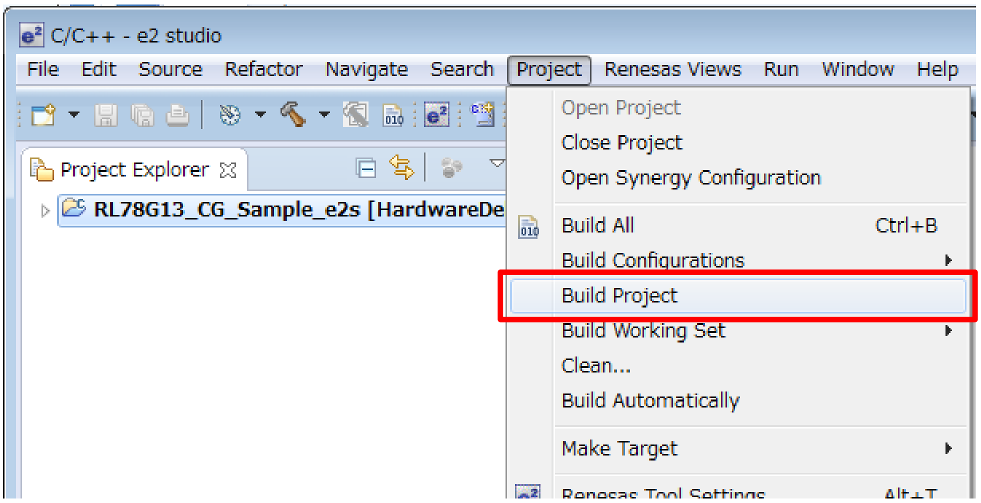


Figure 5 Building the Project

The project will be built and the process is shown in the Build Project dialog box (Figure 6).

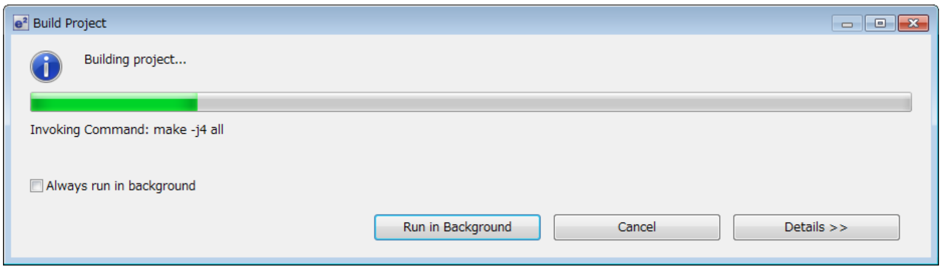


Figure 6 Build Project Dialog Box

When the build has finished, the result is shown in the Console tab (Figure 7).

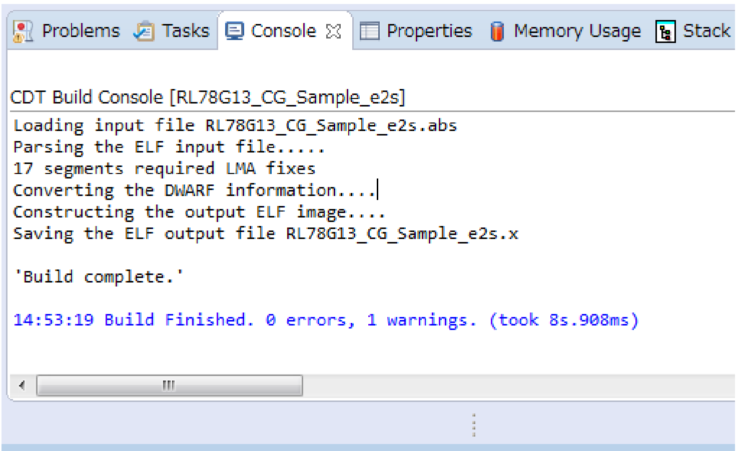


Figure 7 Build Finished

After the build has finished, if you click the Problems tab, a warning is displayed (Figure 8).

The warning appears since the on-chip debug setting is enabled to make on-chip debugging available.

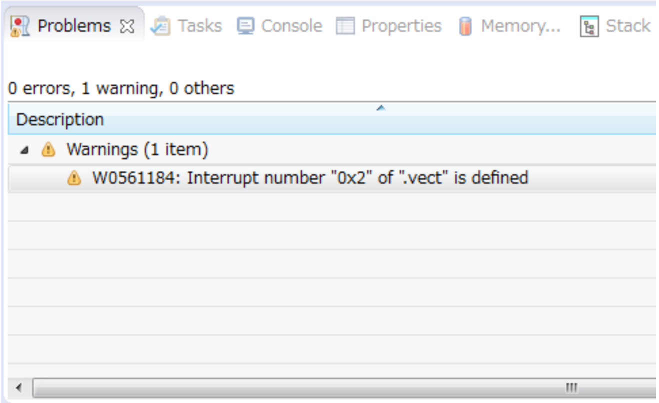


Figure 8 Warning on the Problems Tab

If the on-chip debug setting is disabled, the warning will disappear. Note that on-chip debugging is not available any more with the on-chip debug setting disabled.

<u>Procedure to disable the on-chip debug setting</u>

Expand the project folder in the Project Explorer and select Code Generator >> Peripheral Functions >> Clock Generator (Figure 9).

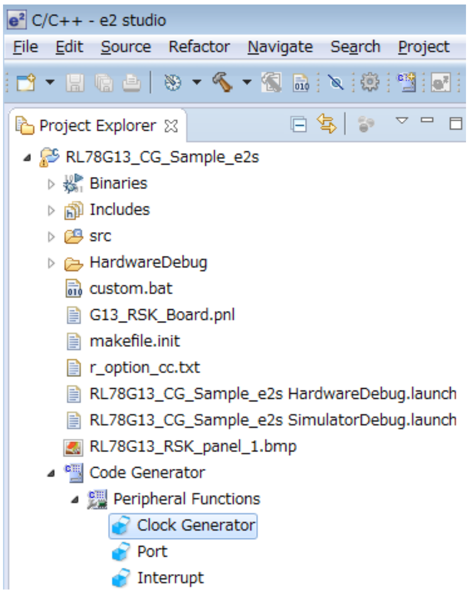


Figure 9 Clock Generator

The Peripheral Functions tab appears (Figure 10). Select the On-chip debug setting tab.

In the On-chip debug operation setting section, change the selection from 'Used' to 'Unused'. Then click the Generate Code button.

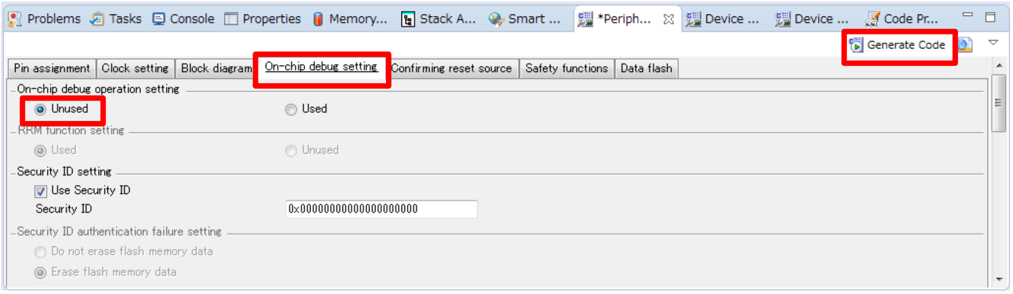


Figure 10 On-Chip Debug Operation Setting

If you rebuild the program now, no warning will appear in the Problems tab (Figure 11).

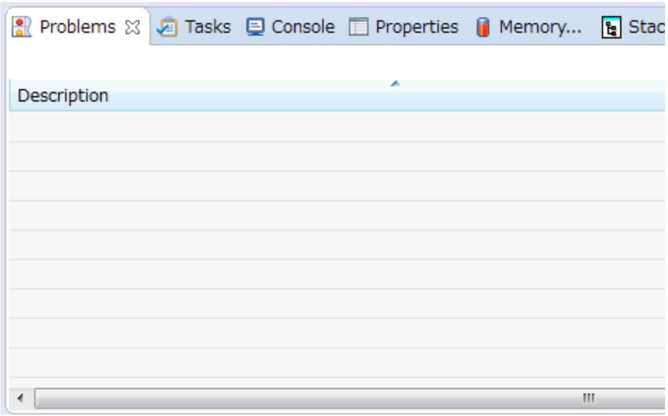


Figure 11 Problems Tab with No Warning

1.4 Debugging

Debugging of the sample program can be performed by the simulator.

Select Run >> Debug Configurations (Figure 12).

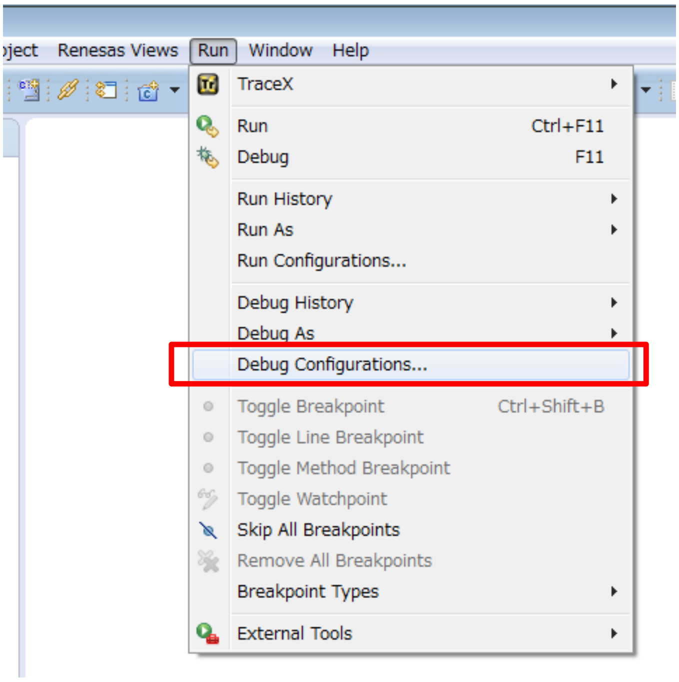


Figure 12 Run Menu

Select 'RL78G13_CG_Sample_e2s SimulatorDebug' and click the Debug button (Figure 13).

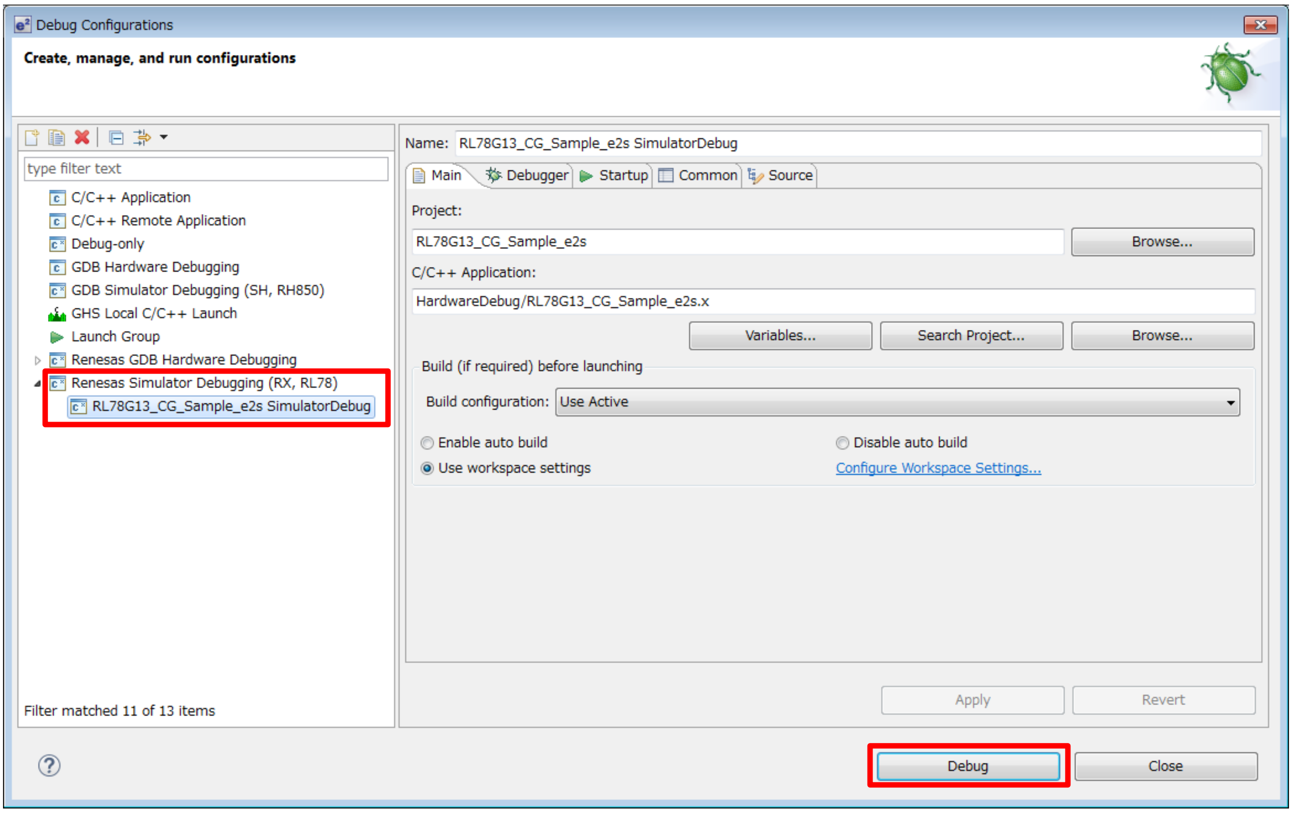


Figure 13 Debug Configuration

If the following dialog box appears (Figure 14), click Yes. The Debug perspective will appear (Figure 15).

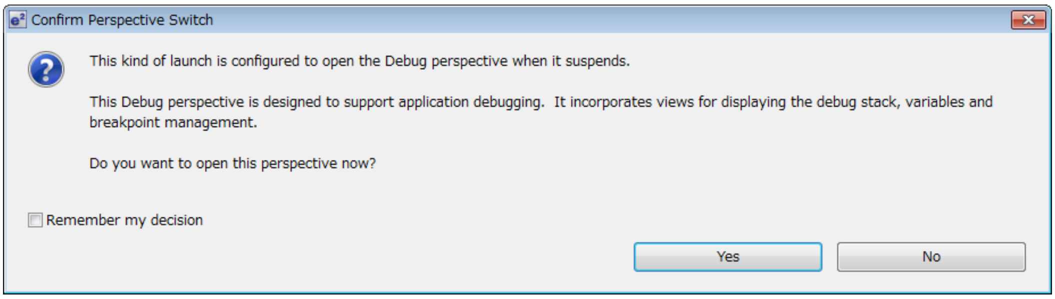


Figure 14 Perspective Switch Confirmation

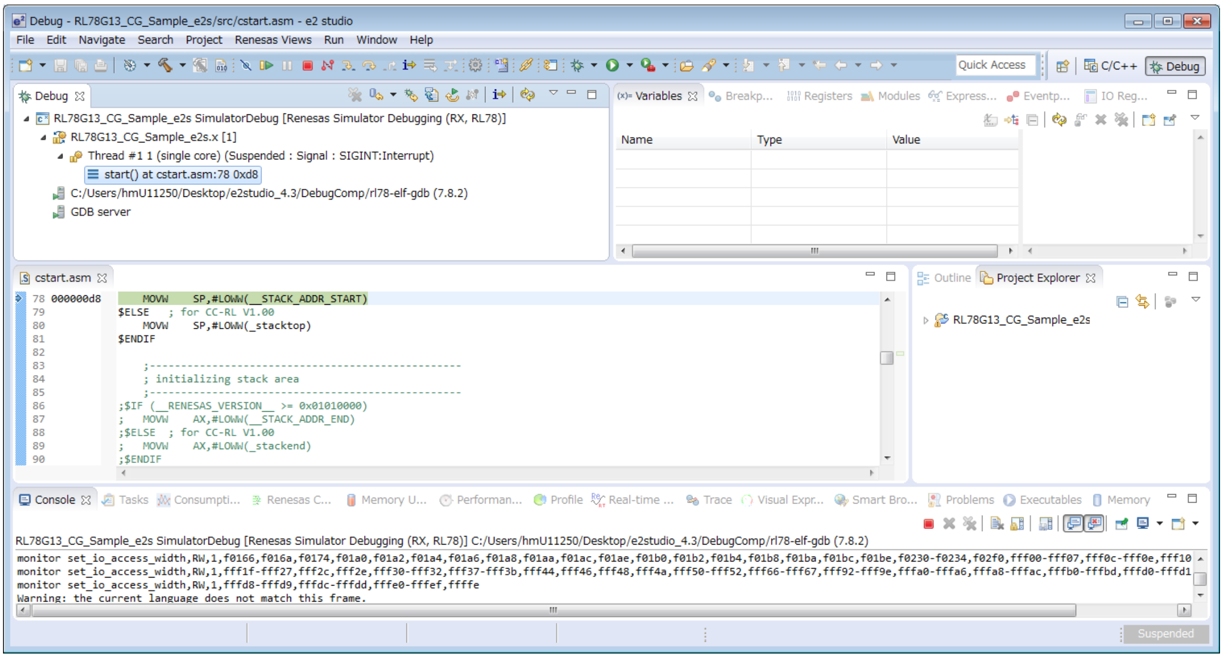


Figure 15 Debug Perspective

Also the Simulator GUI starts up separately (Figure 16). The MCU peripheral functions can be debugged in this window.

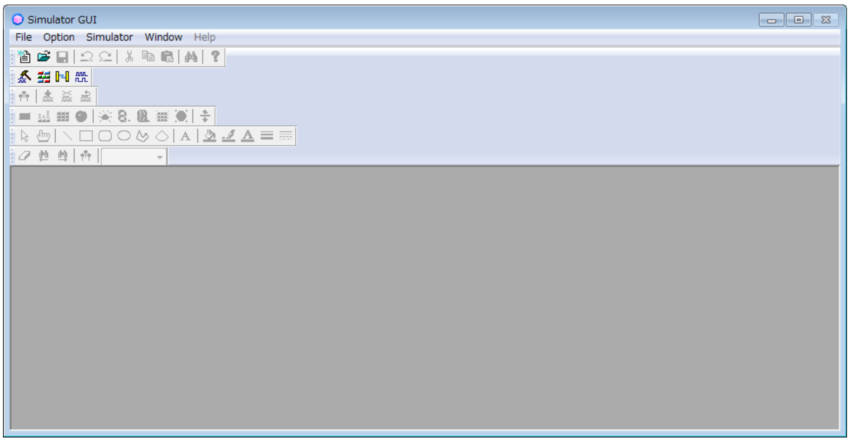


Figure 16 Simulator GUI

Select File >> Open in the Simulator GUI (Figure 17).

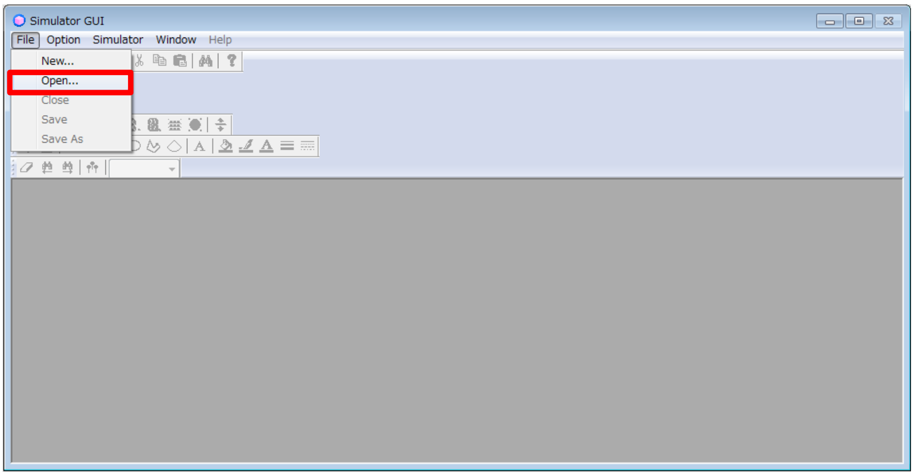


Figure 17 File Menu in the Simulator GUI

Select 'G13_RSK_Board.pnl' in your project folder and click the Open button (Figure 18).

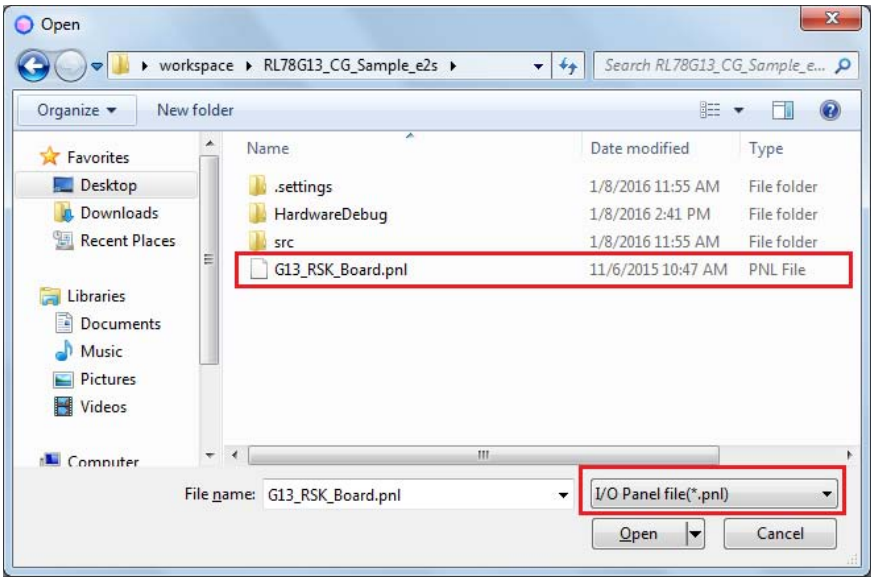


Figure 18 Open Dialog Box in the Simulator GUI

The I/O Panel window opens in the Simulator GUI window (Figure 19). This is the data generated specific to the RSKRL78G13 board. The following functions are available.

- SW1 to SW3 (input)
- Potentiometer (input)
- Reset button (input)
- LED0 to LED3 (output)

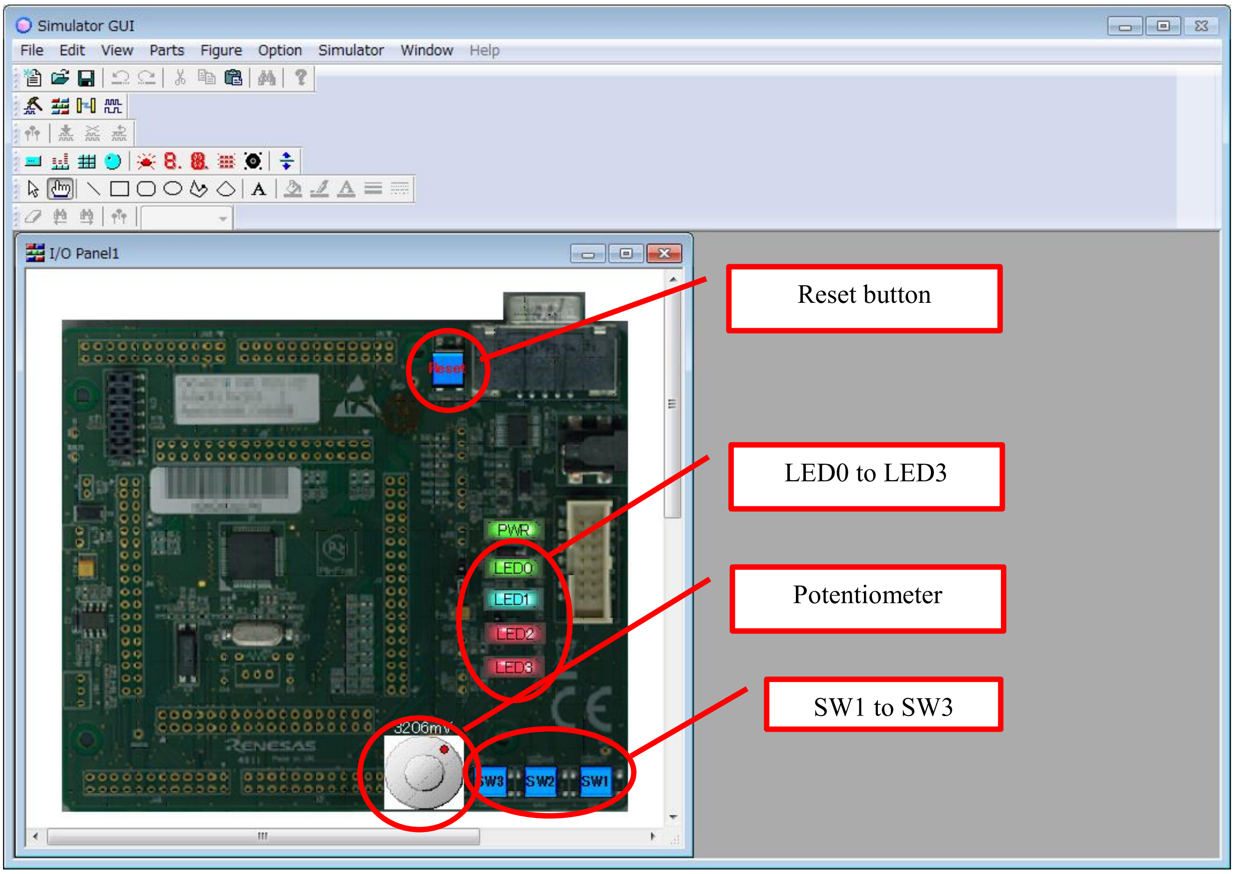


Figure 19 I/O Panel

Go back to the e^2 studio and click the Resume button (Figure 20).

The sample program will be executed.

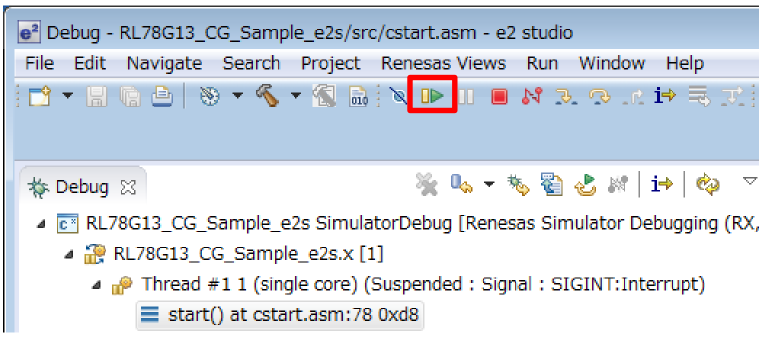


Figure 20 Resume Button

When the program is executed, LED0 starts flashing. LED2 and LED3 are turned off (Figure 21). The flashing speed of LED0 can be adjusted with the potentiometer.

The execution speed of the simulator is slower than the actual MCU. It may take several seconds for the flashing speed to change after the speed has been changed with the potentiometer.

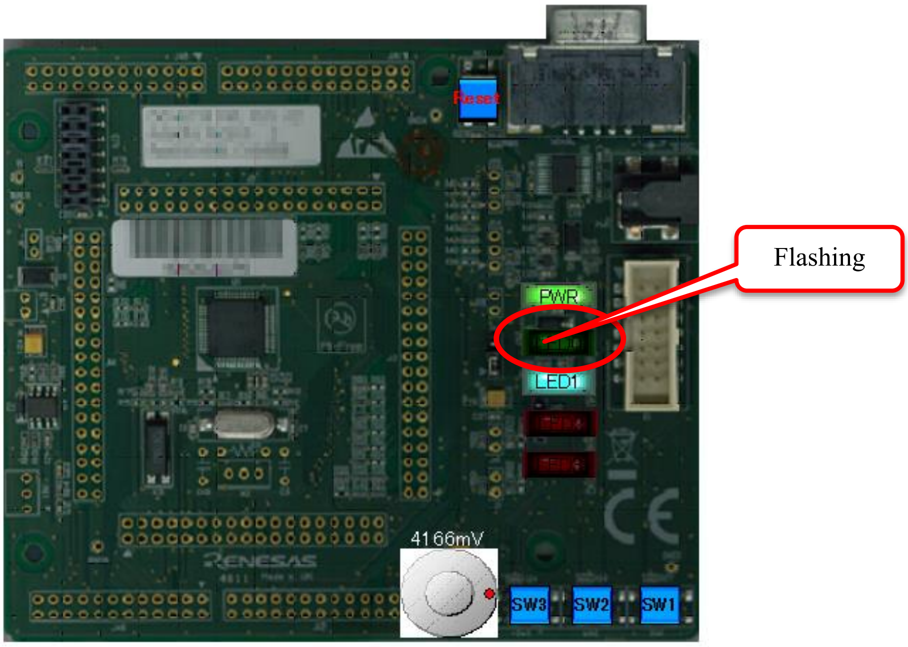


Figure 21 LED Flashing

The other functions are described in 2.3 Detailed Specifications.

1.5 Creating the Configuration File for the Simulator

The configuration file for the simulator is not automatically created, so the user has to make it.

In this sample program, ‘RL78G13_CG_Sample_e2s SimulatorDebug.launch’ is prepared in the project folder as the configuration file for the simulator.

When creating a new configuration file, open the Debug Configurations window and right click on ‘Renesas Simulator Debugging (RX, RL78)’, then select New from the popup menu (Figure 22).

In an example below, the file name is ‘RL78G13_CG_Sample_e2s HardwareDebug (1)’. The file name can be changed to any name.

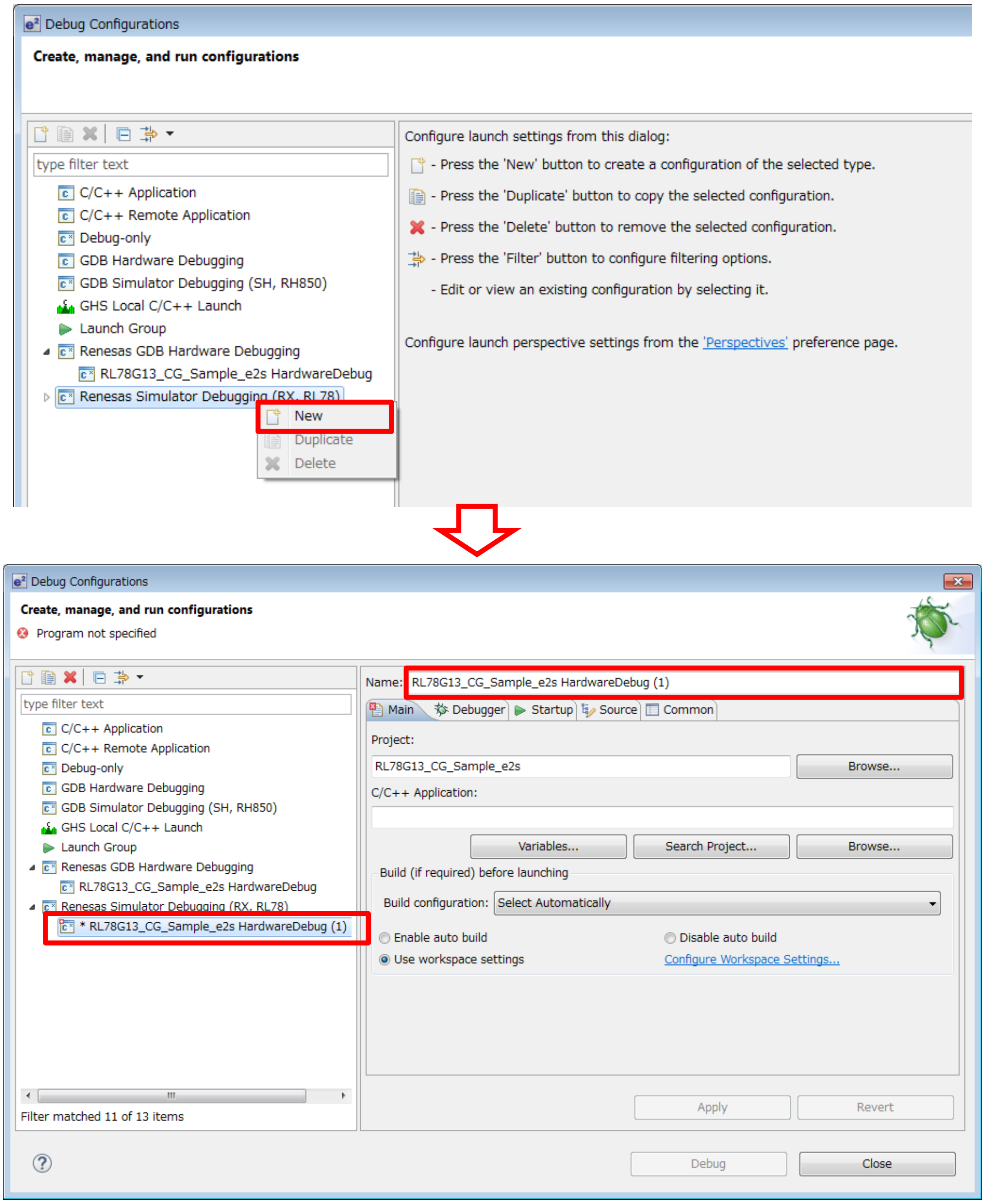


Figure 22 Creating the Configuration File

Some settings need to be specified for the configuration file.

Specify the C/C++ application to be executed in the Main tab (Figure 23).

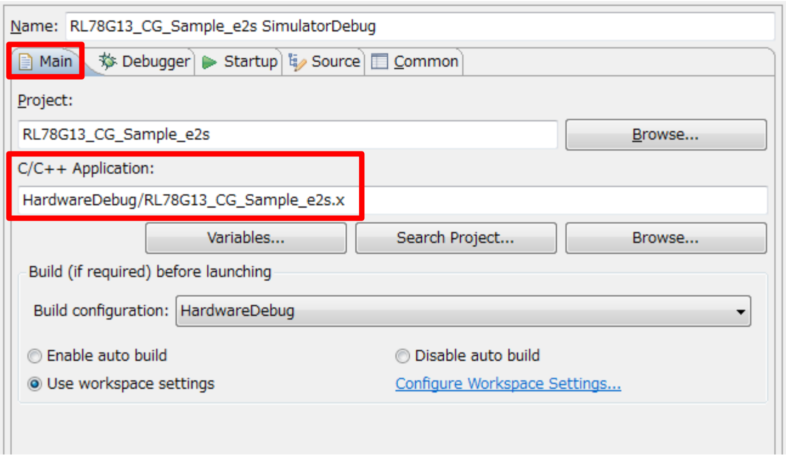


Figure 23 File Configuration 1/3

In the Debugger tab, choose RL78 Simulator in the Debug hardware and specify the Target Device and the Clock (Figure 24).

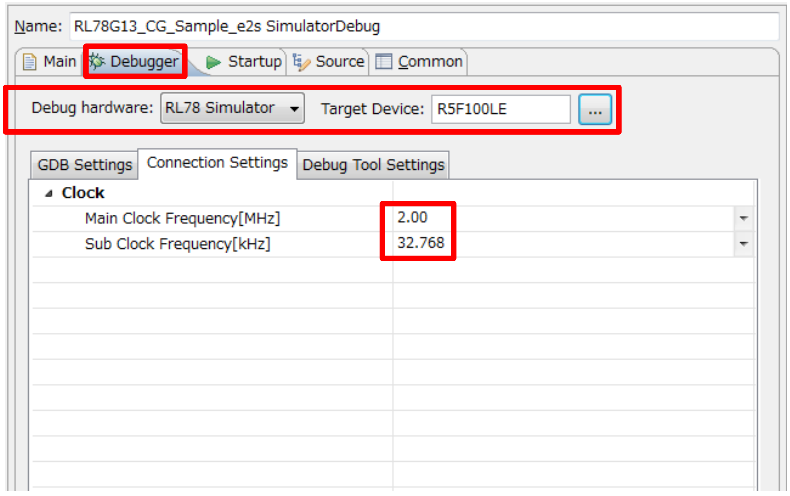


Figure 24 File Configuration 2/3

If you want to save the configuration file in the project folder, select “Shared file” for the Save as option in the Common tab. This does not affect the simulator operation (Figure 25).

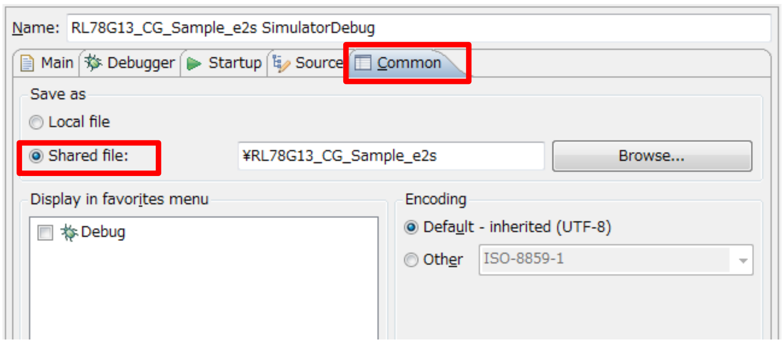


Figure 25 File Configuration 3/3

2. Specifications of the Sample Program

2.1 Overview

The sample program has been designed based on the hardware configuration of RSKRL78G13.

For the MCU, RL78/G13 (64-pin R5F100LE) is used and it operates with the following clocks:

- High-speed system clock: 20 MHz
- High-speed on-chip oscillator: 12 MHz (for SNOOZE mode)
- Subsystem clock: 32.768 kHz

The sample code has the following functions:

1. LED flashing (12-bit interval timer)
2. PWM output
3. Square wave output
4. Counter for the number of times the switch button is pressed (event count in a timer)
5. UART communication
6. CSI communication
7. IIC communication
8. DMA transfer
9. Standby function (HALT, STOP, SNOOZE)

2.2 Pin Assignment

Table 1 Pin Assignment

Pin Number and Name	Pin Function
50 P26/ANI6	AD input (RV1)
21 P31/TI03/TO03/INTP4/(PCLBUZ0) 31 P05/TI05/TO05 40 P16/TI01/TO01/INTP5/(SI00)/(RXD0)	Timer • Event count (TI03) • Square wave output (TO05) • PWM output (TO01)
35 P52/(INTP10) 36 P53/(INTP11) 19 P62 20 P63	LED0 LED1 LED2 LED3
32 P30/INTP3/RTC1HZ/SCK11/SCL11	RTC
44 P12/SO00/TxD0/TOOLTxD/... 45 P11/SI00/RxD0/TOOLRxD/SDA00/...	SAU/UART (serial connecter)
58 P04/SCK10/SCL10 59 P03/ANI16/SI10/RxD1/SDA10 60 P02/ANI17/SO10/TxD1 41 P15/SCK20/SCL20/... 42 P14/RxD2/SI20/SDA20/... 43 P13/TxD2/SO20/...	SAU/CSI • CSI10 (slave reception) • CSI20 (master transmission)
17 P60/SCLA0 18 P61/SDAA0	IIC • IICA0 (master transmission)
33 P50/INTP1/SI11/SDA11 34 P51/INTP2/SO11 21 P31/TI03/TO03/INTP4/(PCLBUZ0)	SW1 SW2 SW3
2 P43 3 P42/TI04/TO04 4 P41/TI07/TO07	Selection of the standby function
22 P77/KR7/INTP11/(TXD2) 23 P76/KR6/INTP10/(RXD2) 24 P75/KR5/INTP9/SCK01/SCL01 25 P74/KR4/INTP8/SI01/SDA01 26 P73/KR3/SO01 27 P72/KR2/SO21 28 P71/KR1/SI21/SDA21 29 P70/KR0/SCK21/SCL21	DMA transfer • RAM to P7

Note: Yellow indicates the pin function used.
Green indicates components on the RSKRL78G13.
Connections such as for the power supply and oscillator are omitted.

2.3 Detailed Specifications

2.3.1 LED Flashing

LED0 flashing is performed using the 12-bit interval timer. The LED0 flash interval can be changed with RV1.

The default setting of LED0 flashing is turned off (port output = high). The port output is inverted every time a 12-bit interval timer interrupt (INTIT) occurs.

The cycle of the 12-bit interval timer changes according to the A/D conversion value. The cycle range is from 10 ms to 100 ms (0V: 10 ms, VCC: 100 ms). The A/D conversion value is stored in the gADC_Result variable (uint16_t).

Hardware trigger wait mode (INTRTC) is used as the A/D conversion trigger and A/D conversion is performed at 0.5-second intervals. 0.5 seconds of the RTC can be checked with the RTC1HZ pin.

Table 2 Settings of the ADC

Item	Settings
Resolution	8 bits
Trigger mode	Hardware trigger wait mode INTRTC specified
Conversion operation mode	One-shot conversion mode ANI06 (RV1 connected) is selected.

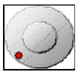
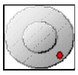
<u>Procedure to simulate LED flashing</u>

If a program is running, halt the program.

Select File >> Open in the Simulator GUI and open the I/O panel file ‘G13_RSK_Board.pnl’ in the project folder.

When the program is restarted, LED0 flashes.

Use the potentiometer to adjust the flash interval. The potentiometer works as follows:

 10 ms in 0 mV  100 ms in 5000 mV

2.3.2 PWM Output

Timer channels 0 and 1 are used to output PWM on the TO01 pin.

The output period is 1 kHz (1-ms period) and the duty range is from 10 to 90%. Duty varies by 10%, and duty 0% and 100% are not used. The initial duty value is 10% and the value is increased by 10% every time SW2 is pressed. When the duty is 90%, the value returns to 10% by pressing SW2. The reference value is duty 10% which is specified in the Code Generator.

INTP2 is used for SW2. LED2 is inverted every time the INTP2 interrupt occurs.

<u>Procedure to view the PWM waveform</u>

If a program is running, halt the program.

Select Simulator >> Timing Chart or click .

The Timing Chart window appears (Figure 26). Select Edit >> Select Pin, enter “P16/TI01/TO01/INTP5” in the Pin Name section and click OK.

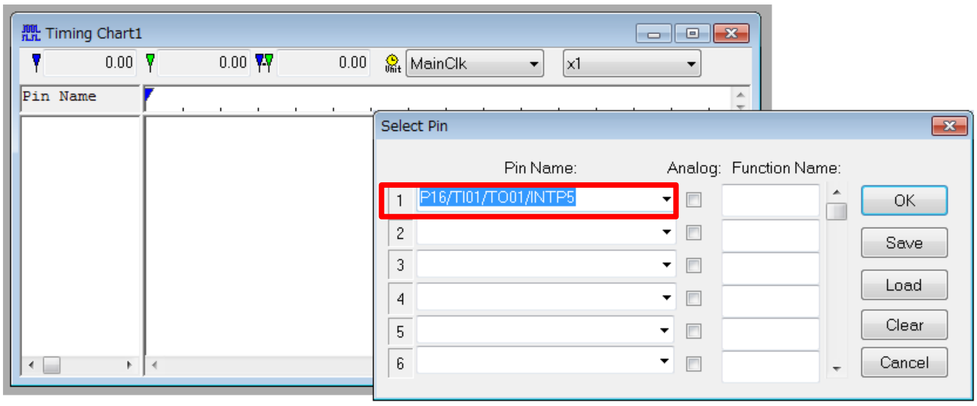


Figure 26 Timing Chart Window

When the program is restarted, the waveform is displayed in the Timing Chart window. Halt the program and measure the period.

Right click on the window where the waveform is displayed and select Move markerA (blue) and Move markerB (green) from the popup menu to place markers. Then check whether the period is 1 ms.
The marker positions can be moved to a rising edge or a falling edge by selecting Edit >> Search.

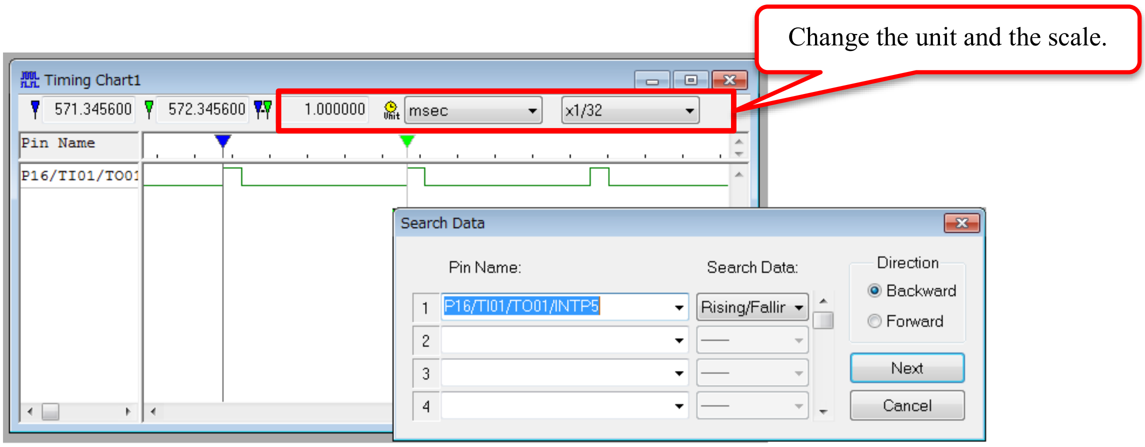


Figure 27 PWM Output Waveform

2.3.3 Square Wave Output

Timer channel 5 is used to output a square wave from the TO05 pin.

The output period is 1 kHz (1-ms period).

Procedure to view square waveforms

The procedure is the same as the one to view PWM waveforms. Pin to be registered is 'P05/TI05/TO05'.

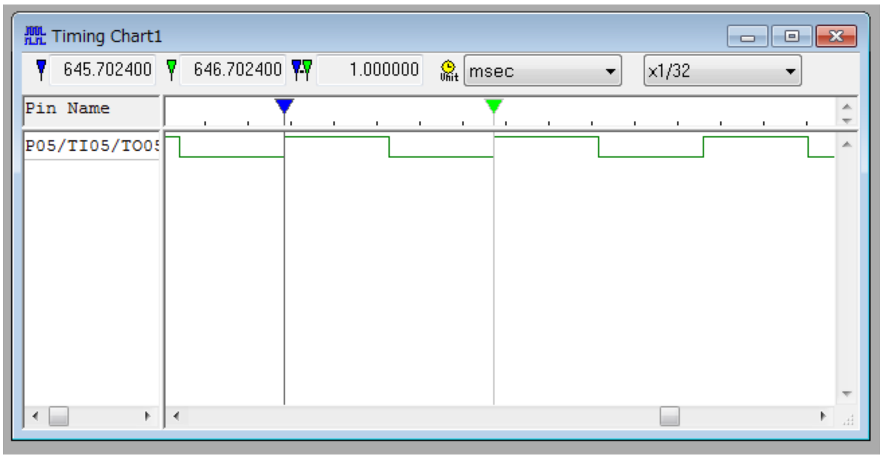


Figure 28 Square Waveform Output

2.3.4 Counting the Number of Times the Switch Button is Pressed

Channel 3 of the timer is used to count the number of times SW3 (TI03) is pressed.

The count value is set to 1 and the timer interrupt (INTTM03) occurs every time SW3 is pressed. The variable for the count is incremented in the interrupt handler. The count value is stored in the gEventCount variable (uint16_t). When the counter overflows, the value is returned to 0 (overflow not controlled).

LED3 is inverted every time the counter is incremented.

Procedure to check the event count

If a program is running, halt the program.

Select File >> Open in the Simulator GUI and open the I/O panel file 'G13_RSK_Board.pnl' in the project folder.

If SW3 is pressed on the I/O panel while the program is running, the variable is incremented and LED3 is inverted.

2.3.5 UART Communication

UART0 is used for UART communication.

In UART communication, one character is transmitted every 0.5 seconds using INTRTC and 12 characters (‘0’ to ‘9’, ‘\r’, and ‘\n’) are repeatedly transmitted one by one.

If the received data is ‘z’, the transmission is paused. When data other than ‘z’ is received, the transmission is restarted.

Table 3 Settings of UART0

Item	Setting
Data length	8 bits
Transfer direction	LSB
Parity setting	None
Stop bit	1
Baud rate	19200 bps

<u>Procedure to simulate UART communication</u>

If a program is running, halt the program.

Select Simulator >> Serial or click .

The Serial window appears (Figure 29). Select ‘UART0’ on the top left of the window. Once selected, it cannot be changed.

Click the Format button to configure settings for UART0.

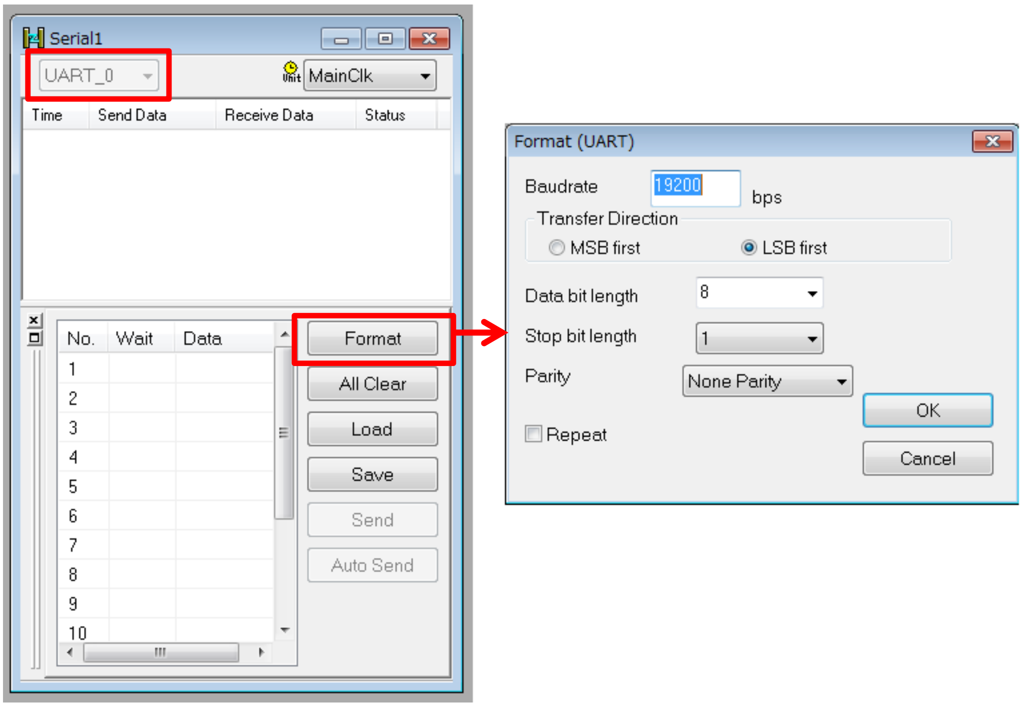


Figure 29 Serial Window (UART0) 1/2

When the program is restarted, the data transmitted by the MCU are displayed in the Serial window. The data are received data in the serial window side, thus they are displayed under the Receive Data (Figure 30).

To transmit 1-byte data to the MCU, input the data to be transmitted in the bottom left window and click the Send button.

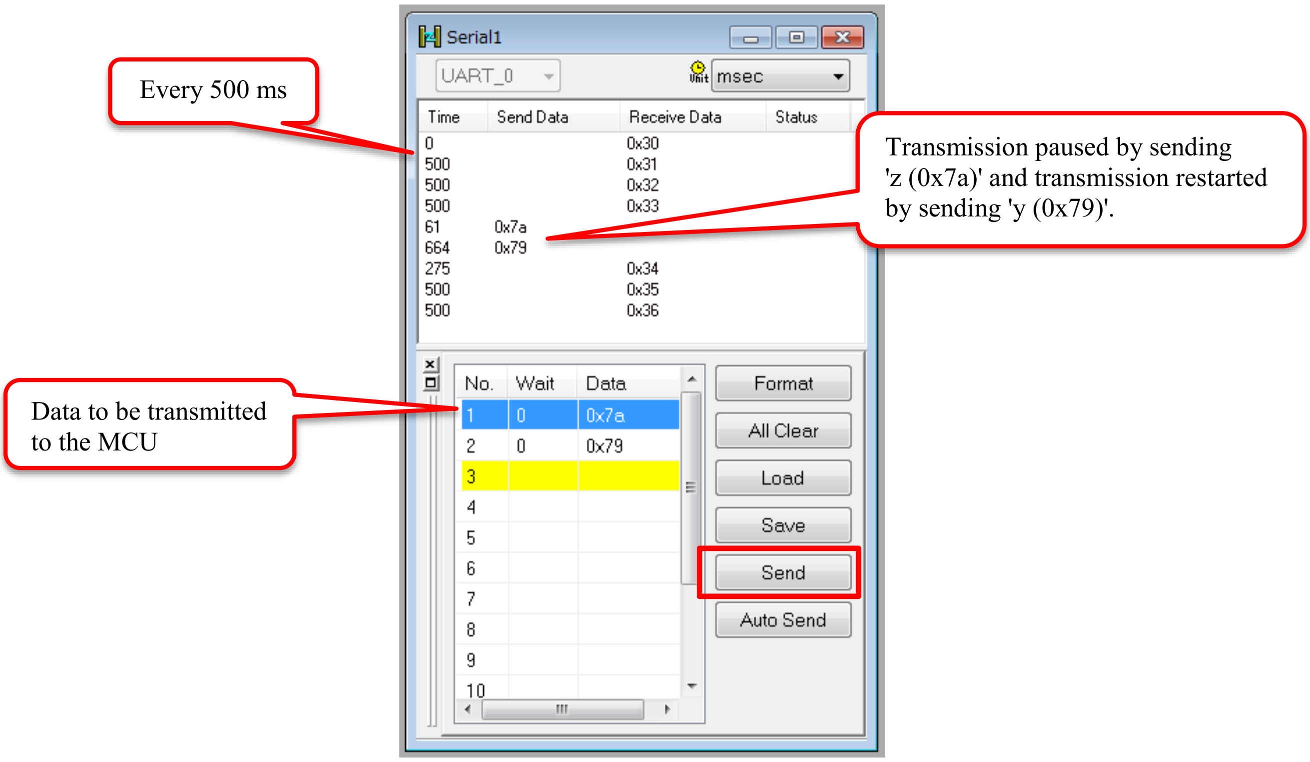


Figure 30 Serial Window (UART0) 2/2

2.3.6 CSI Communication

CSI10 is used for master transmission and CSI20 is used for slave reception.

CSI10 transmits data which is stored in the buffer “gCSI10_TXBuffer (16-byte array)” with the initial value. It transmits 16-byte data every time the switch button is pressed. The initial vale of the 16-byte data is “CSI10 transmit. (15 characters + 0x00 (end of data))”.

CSI20 stores the received data in the buffer “gCSI20_RXBuffer (16-byte array)”. If the received data exceeds 16 bytes, the data is overwritten from the beginning of the buffer repeatedly.

Table 4 Settings of CSI10 and CSI20

Item	Setting of CSI10	Setting of CSI20
Transfer mode	Single transfer mode	
Data length	8 bits	
Transfer direction	LSB	
Data transmit/receive timing	Type 1	
Clock mode	Internal clock (master)	External clock (slave)
Baud rate	10000 bps	—

Procedure to simulate CSI communication

If a program is running, halt the program.

Select Simulator >> Serial or click .

The Serial window appears (Figure 31). Select ‘CSI_10’ on the top left of the window. Once selected, it cannot be changed.

Click the Format button to configure settings for the CSI10. Then select ‘Repeat’ on the bottom left in the Format dialog box.

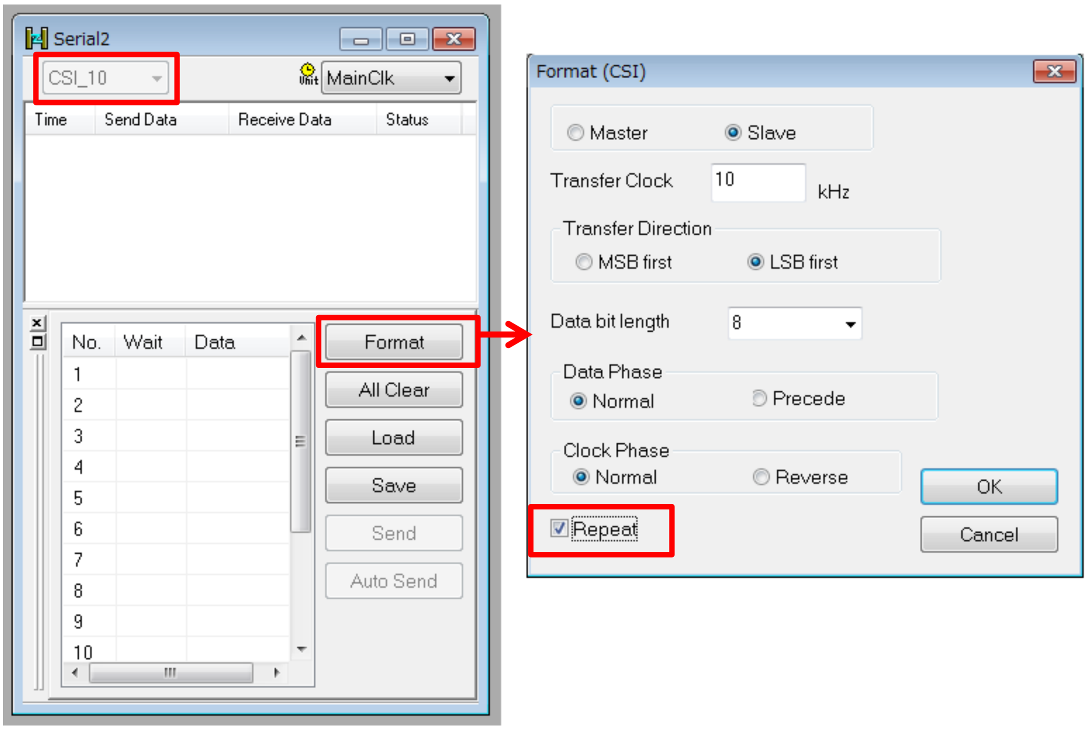


Figure 31 Serial Window (CSI10) 1/2

Set 1-byte transmit data (any value) in the bottom left of the Serial window. Restart the program and click the Auto Send button once in the Serial window.

At this stage, the data transmitted by the MCU can be received by pressing SW2 on the I/O panel. Then, the transmit data set previously is transmitted to the MCU at the same time. In this sample program, the data received by CSI10 are not used.

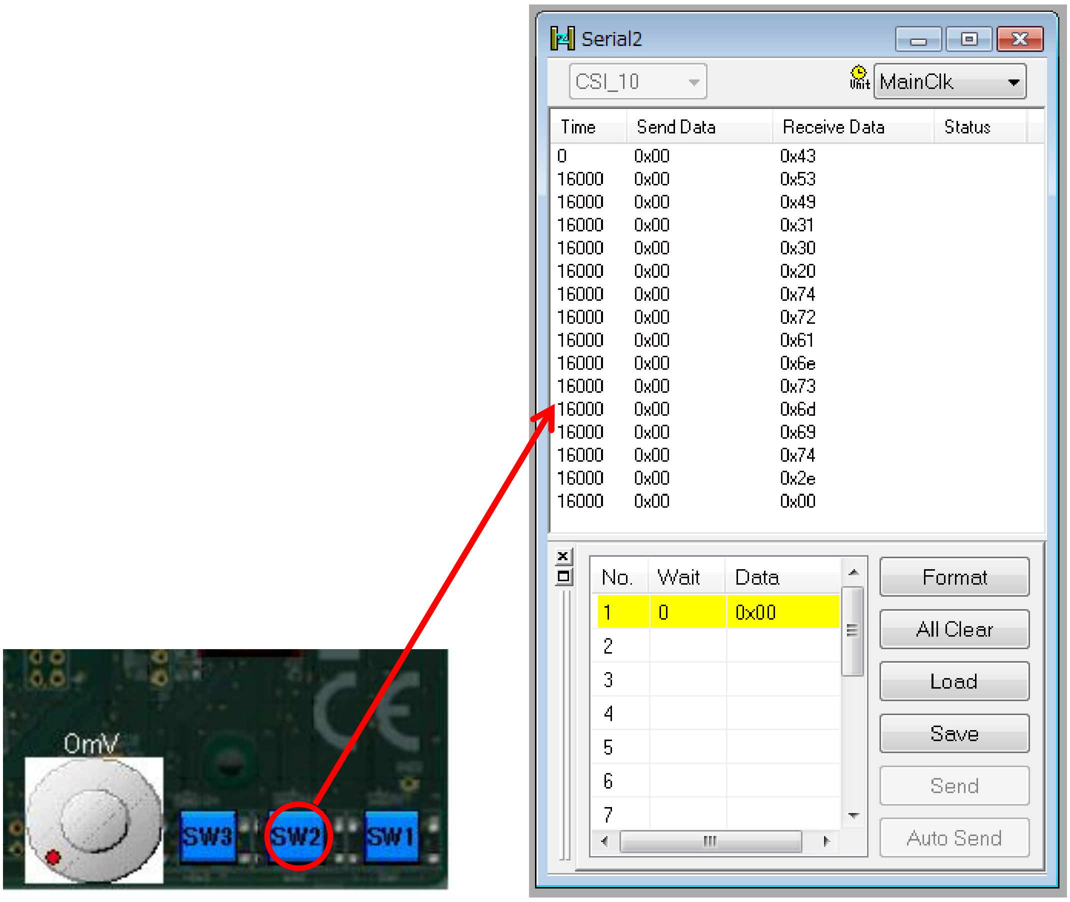


Figure 32 Serial Window (CSI10) 2/2

Open the Serial window to select ‘CSI_20’ and click the Format button to configure settings for CSI20 (Figure 33).

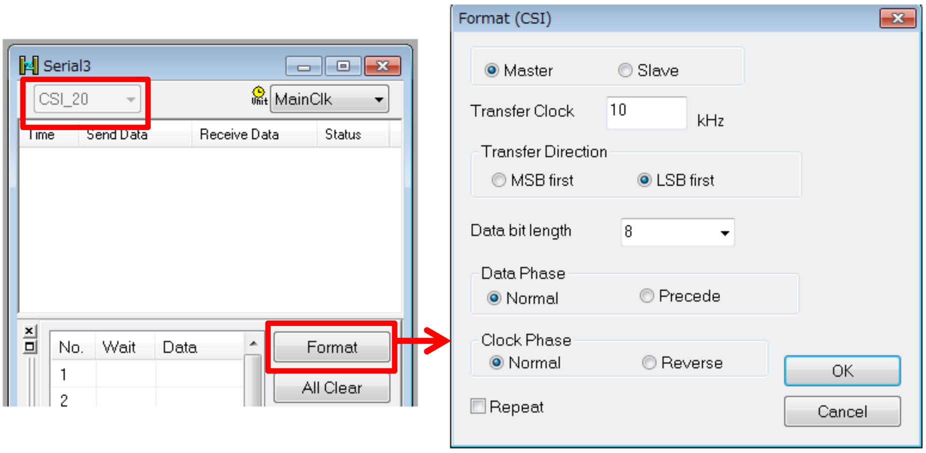


Figure 33 Serial Window (CSI20) 1/2

Input the transmit data in the bottom left of the Serial window and click the Send button to transmit the data in 1 bytes. If you click the Auto Send button, then all data will be transmitted (Figure 34).

The data transmitted from the Serial window can be confirmed in the receive buffer for CSI20 (gCSI20_RXBuffer). Now go back to the e^2 studio, add ‘gCSI20_RXBuffer’ to the Expression and confirm the value.

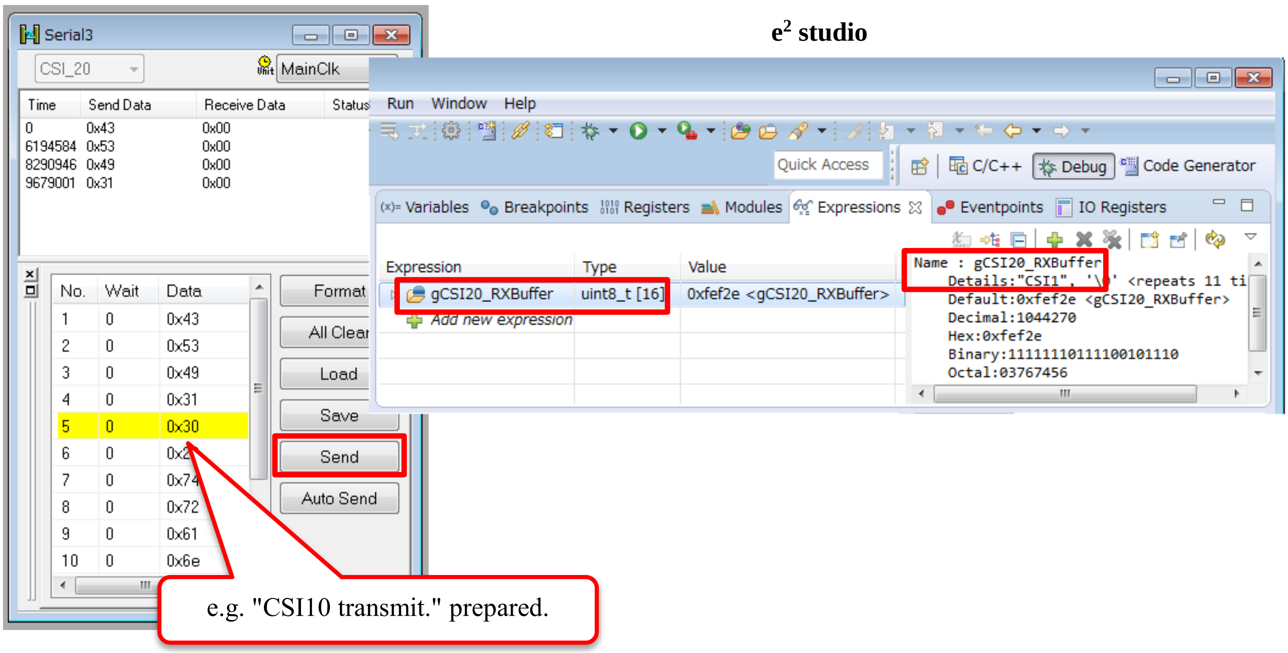


Figure 34 Serial Window (CSI20) 2/2

2.3.7 IIC Communication

IICA0 is used for single master transmission.

8-byte data is transmitted to the slave address 0xA0 every time SW2 is pressed. The content of the 8-byte data is as follows: 0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77.

Table 5 Settings of IICA0

Item	Setting of CSI10
Transfer mode	Single master
Local address	0x10
Transfer clock	100000 bps

<u>Procedure to simulate IIC communication</u>

If a program is running, halt the program.

Select Simulator >> Serial or click .

The Serial window appears (Figure 35). Select 'IICA_0' on the top left of the window. Once selected, it cannot be changed.

Click the Format button to configure settings for IICA0. Then select 'Repeat' on the bottom left in the Format dialog box.

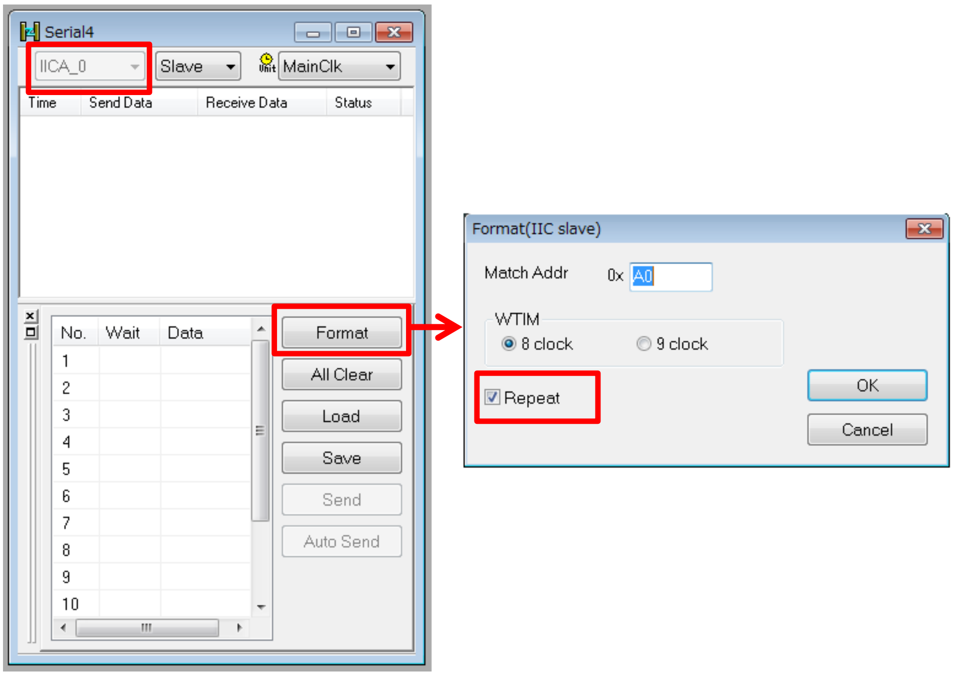


Figure 35 Serial Window (IICA0) 1/2

When the program is restarted and SW2 is pressed on the I/O panel, the MCU transmits the slave address and data.

When the slave addresses match, an ACK response is returned and the IIC bus state becomes bus-hold state. If the Auto Receive button is clicked at this point, the subsequent data are received (Figure 36).

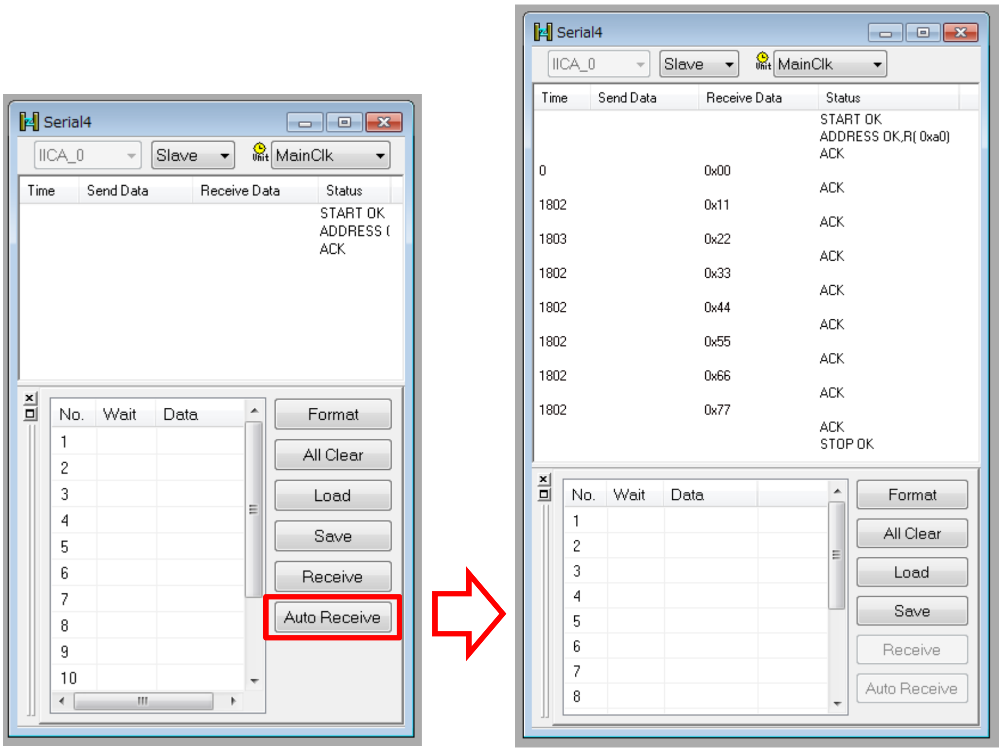


Figure 36 Serial Window (IICA0) 2/2

2.3.8 DMA Transfer

Data is transferred from the RAM to the SFR using the DMA.

The RAM address of the transfer source is reset to the start address of the 16-byte array declared in the program. The initial value of 16-byte data in the RAM is as follows: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f.

Channel 2 of the timer is used as the interval timer and the interval time is set to 1000 ms. The INTDMA0 interrupt occurs after 16-byte data is transmitted. Then the start address of the transfer source address is set back to the start address of the array and 16-byte transfer is repeatedly configured by the user software.

Table 6 Settings of DMAC

Item	Setting
RAM address	0xfeff0 (temporary setting for generating code)
SFR address	P7 - 0x000fff07
Number of transfers	16
Trigger signal	INTTM02

<u>Procedure to confirm P7</u>

Add '*(char *)0xFF07' to the Expression in the e[2] studio. Select the added line and right click on it, and then select Real-time Refresh from the popup menu. If 'R' is shown on the left, the real-time refresh is enabled (Figure 37).

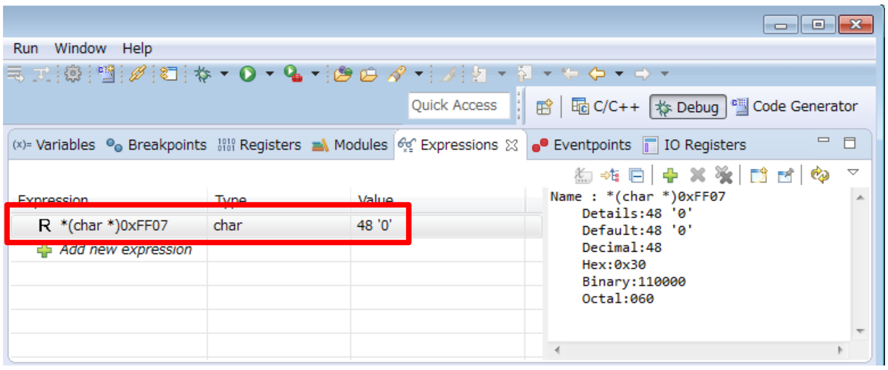


Figure 37 P7 Value

2.3.9 Standby Function

Pressing SW1 (INTP1) causes transition to one of the standby function modes.

The mode to be transitioned to is determined with a combination of P41 to P43.

Table 7 Transition Mode

P41	P42	P43	Transition Mode	Condition for Canceling the Mode
H	L	L	HALT mode	SW1 pressed
L	H	L	STOP mode	SW1 pressed
L	L	H	SNOOZE mode	Result of A/D conversion is 200 or greater.
Combination other than above			Invalid (transition is not made even if the switch button is pressed.)	

HALT and STOP modes are canceled by pressing SW1 (INTP1 interrupt occurs).

During SNOOZE mode, A/D conversion is performed using the INTRTC interrupt as the trigger (CPU remains stopped). When the conversion result is 200 or greater, an interrupt is generated and SNOOZE mode is canceled in the interrupt handler.

LED1 is turned on during normal operation. When transitioning to a standby function mode, LED1 is turned off. Also, LED0, LED2 and LED3 are all turned off and operations described from sections 2.3.1 to 2.3.8 are halted.

Procedure to confirm the standby function

If a program is running, halt the program.

Select Simulator >> Signal Data Editor or click  button.

The Signal Data Editor window appears (Figure 38). Select Edit >> Select Pin, enter “P41/TI07/TO07”, “P42/TI04/TO04”, and “P43” in the Pin Name section and click OK.

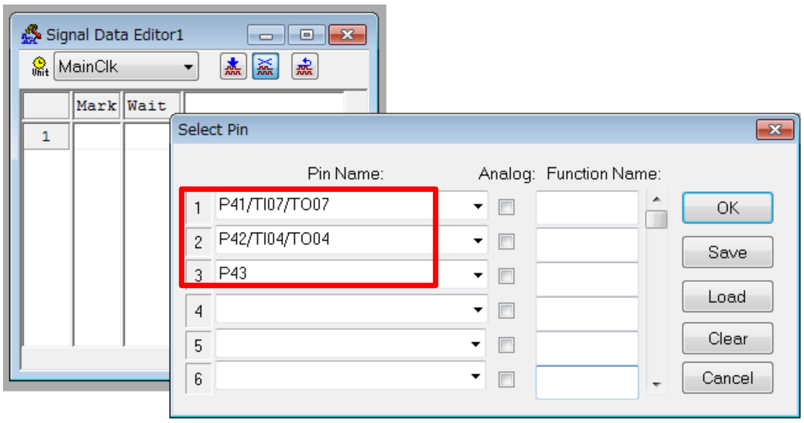


Figure 38 Signal Data Editor Window 1/2

Specify ‘100’ from P41 to P43 in order and click the Start Signal Input button (Figure 39).

If SW1 is pressed in the I/O panel when ‘100’ is input for P41 to P43, the mode to be transitioned to is HALT mode. When the value input for P41 to P43 is ‘010’, the mode to be transitioned to is STOP mode, and when the value is ‘001’, the mode to be transitioned to is SNOOZE mode.

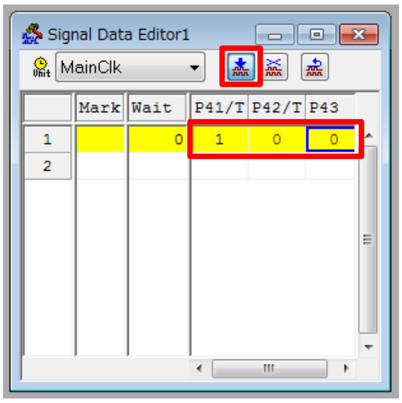


Figure 39 Signal Data Editor Window 2/2

Halt the program and check where the program counter stopped to confirm the mode entered (Figure 40).

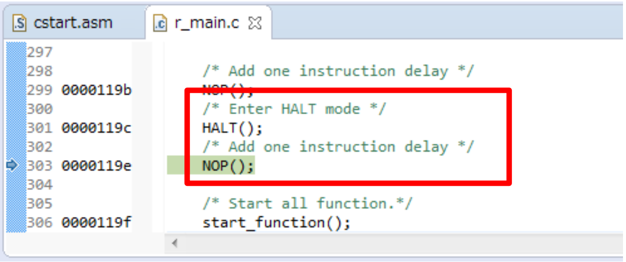


Figure 40 Where the program halted (when transitioning to HALT mode)

Website and Support

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

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Feb 05, 2016	--	First edition issued.

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

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

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

RENESAS

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

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

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

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

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

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

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

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

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

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

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

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL II Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 5.0