Smart Analog

Smart Analog Development Tutorial for TSA-IC500

R20AN0258EJ0100
Rev.1.00
Mar 31, 2014

Introduction

This tutorial manual is designed to provide the user with an understanding of the hardware functionality of the Smart Analog IC 500 evaluation board TSA-IC500 and its development tool chains. It is intended for users evaluating this Smart Analog IC platform and using this evaluation board.

Target Devices

Smart Analog IC 500 (RAA730500DFP) and RL78/G1A (R5F10ELEAFB)

Contents

1. Overview

This application note describes an example of developing a program that drives a phototransistor and controls LEDs in the Smart Analog development environment, by using the Smart Analog IC 500 evaluation board TSA-IC500 (TSA-IC500) on which a circuit configurable Smart Analog IC 500 and RL78/G1A are mounted.

1.1 Development environment

Use the latest version of development tools for the Smart Analog development environment.

Sensor Select Web Simulator	Renesas VA Version 2.0.0.0 or later
Analog Front End Design Tool	SA-Designer V1.02.00 or later
IDE	CubeSuite+ V2.00.00 or later

2. Required items

Prepare the items below before starting development by following the steps described in this document.

Required items

- Tessera Technology evaluation board TSA-IC500 incorporating Smart Analog IC
- Renesas on-chip debugging emulator E1
- New Japan Radio phototransistor NJL7502L
- LED × 2 (any color)
- 470 Ω resistor × 2 (1/8W)
- 1 kΩ resistor (1/8W)
- 50-pin female socket (2 × 25 pins) × 2 and 50-pin male header (2 × 25 pins) × 2
- Board × 2

2.1 Tessera Technology evaluation board TSA-IC500 incorporating Smart Analog IC

Smart Analog IC 500 (RAA730500DFP) and RL78/G1A (R5F10ELEAFB) are mounted on this board.

How to obtain TSA-IC500

Corporate customer - Place an order with a Renesas distributor.

Private customer - Purchase the product from the Tessera Technology website.

- Tessera Technology Inc.
  http://www.tessera.co.jp/tsa-ic500.html (Japanese website)

2.2 Renesas on-chip debugging emulator E1

In this document, the E1 emulator is connected to the evaluation board TSA-IC500 and used as a debugger/flash memory programmer.

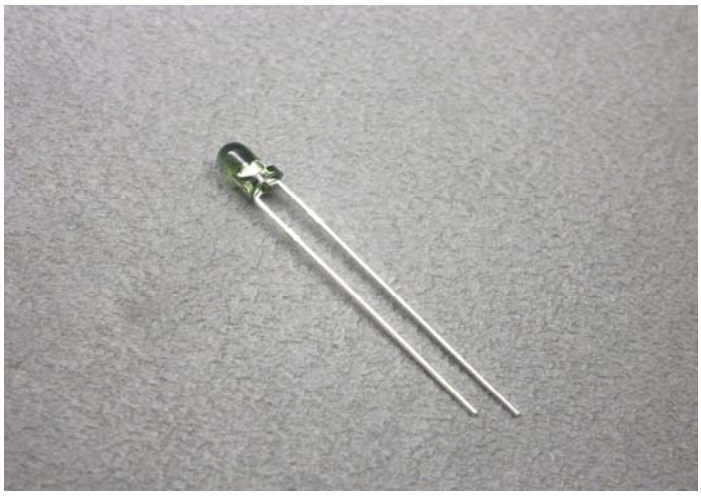
For all product sales, samples, status and availability information, please contact a Renesas Electronics distributor or sales representative in your area.

On-chip debugger E1

Part No.: R0E000010KCE00

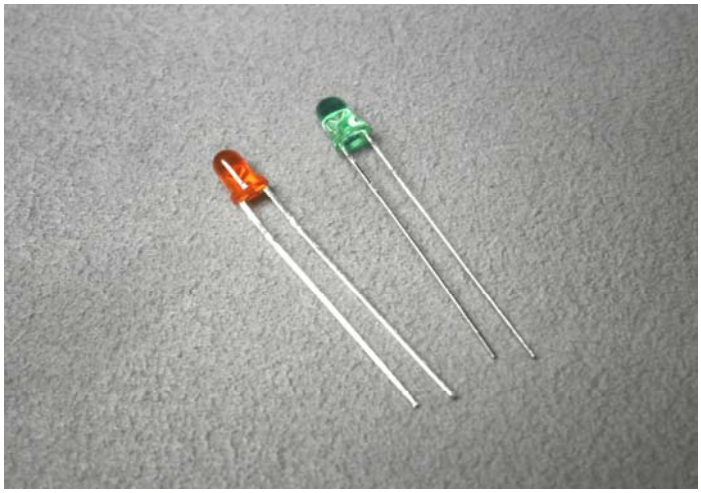
- Renesas Buy/Samples:
  http://www.renesas.com/buy/index.jsp?campaign=gn_buy

2.3 New Japan Radio phototransistor NJL7502L



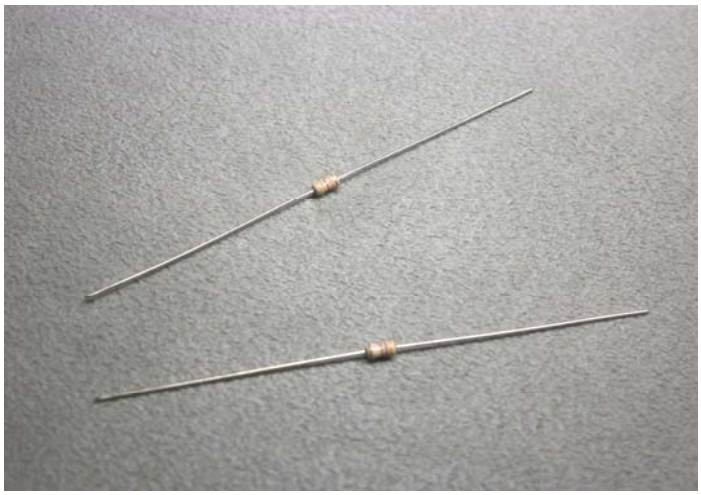
In this document, this phototransistor functions as a sensor driven by the Smart Analog IC. This product is available from parts suppliers or online.

2.4 LED × 2 (any color)



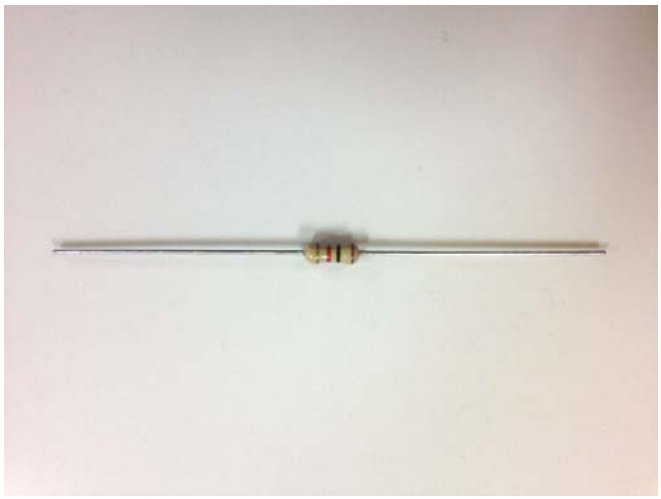
These LEDs are controlled from the RL78/G1A and blink to indicate the sensor operating status. This product is available from parts suppliers or online. You can choose any color.

2.5 470 Ω resistor × 2 (1/8W)



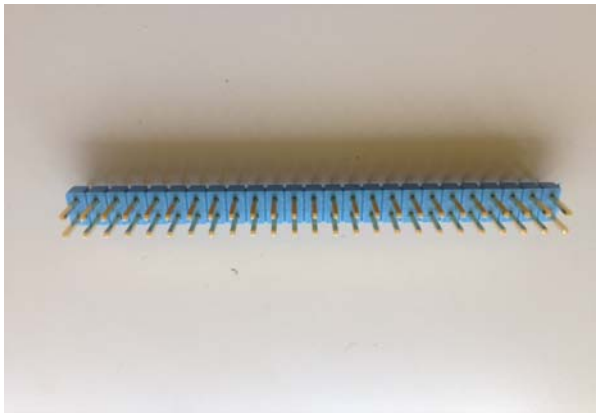
These resistors determine the current that flows when driving LEDs via the RL78/G1A ports. This product is available from parts suppliers or online.

2.6 1 kΩ resistor (1/8W)

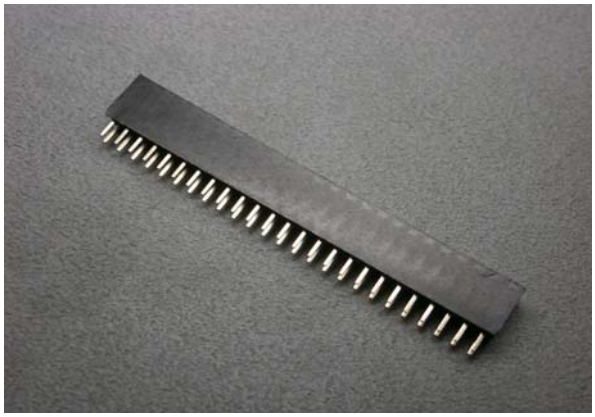


This resistor determines the current that flows when driving the sensor. This product is available from parts suppliers or online.

2.7 50-pin female socket (2 × 25 pins) × 2 and 50-pin male header (2 × 25 pins) × 2



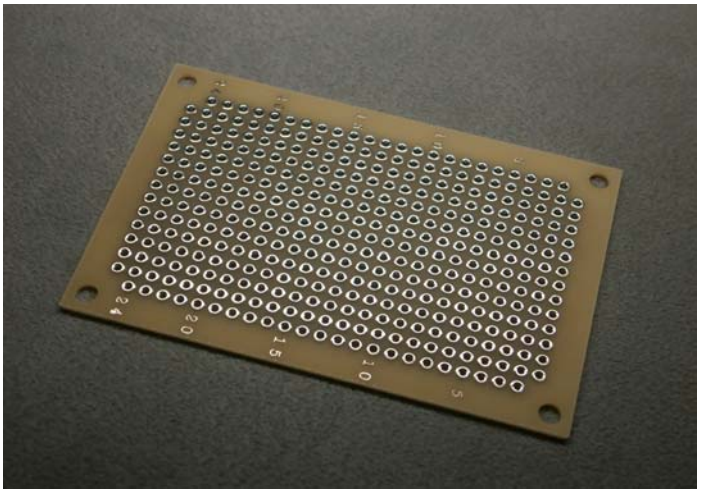
50-pin male header (2 × 25 pins)



50-pin female socket (2 × 25 pins)

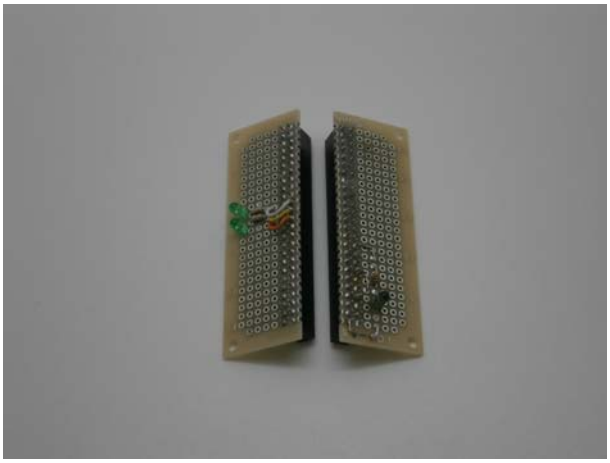
These connect the sensor to the LEDs on the evaluation board TSA-IC500. Prepare two of each. This product is available from parts suppliers or online.

2.8 Board × 2



These are the boards on which the female sockets are soldered. One is used to mount LEDs, and the other is used to mount the sensor. Prepare boards that have a space larger than a 25 × 2 matrix to connect the 50-pin sockets.

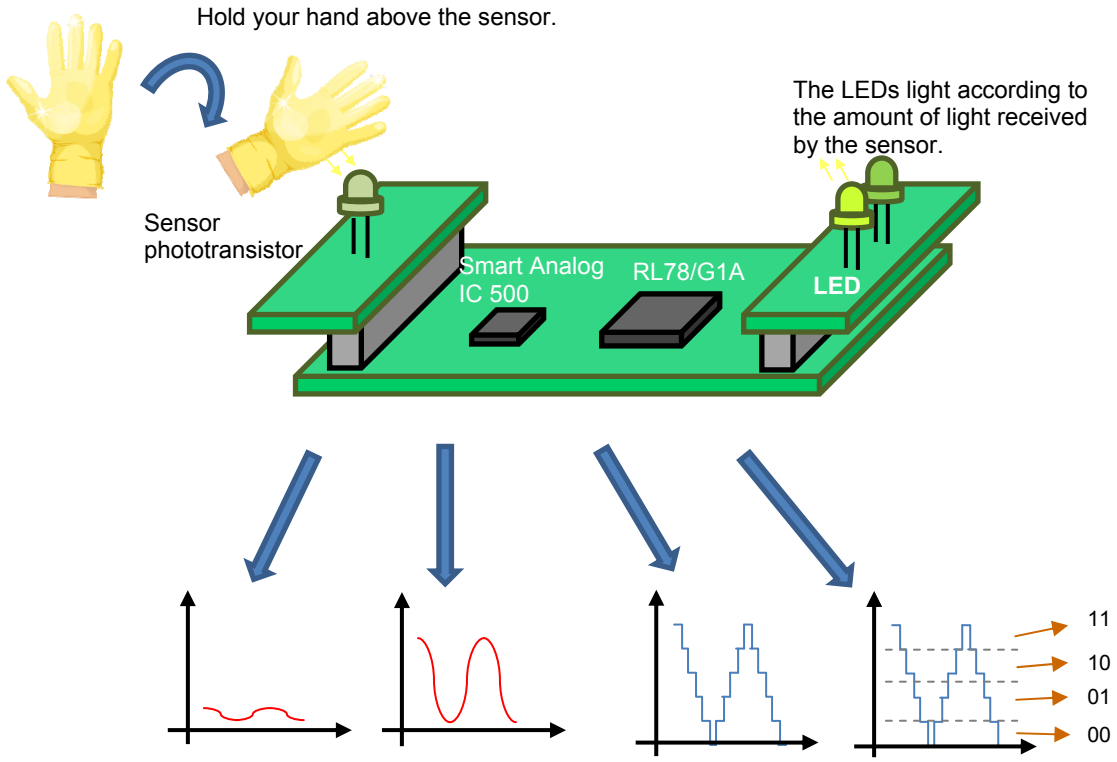
The photos below show the boards separated in two, onto which the pin sockets, LEDs, and sensor are mounted. Prepare the board as shown in these photos.

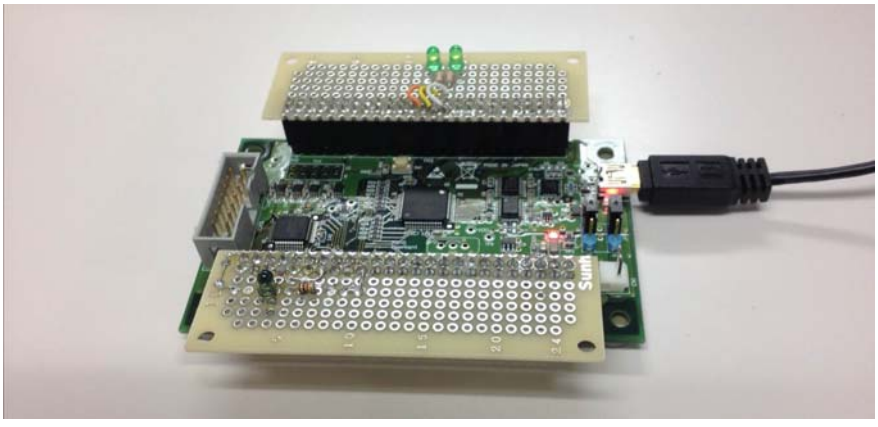


3. Overview of development

In this system, the phototransistor senses the amount of light and lights the LEDs accordingly. By using the Smart Analog IC, you can amplify the signal output from the phototransistor to adjust it to an optimum value, and then digitize the signal by using an A/D converter (ADC).

The ADC measurement range is divided into four ranges, and the LEDs are controlled to indicate "00", "01", "10", or "11", according to the signal output from the Smart Analog IC.





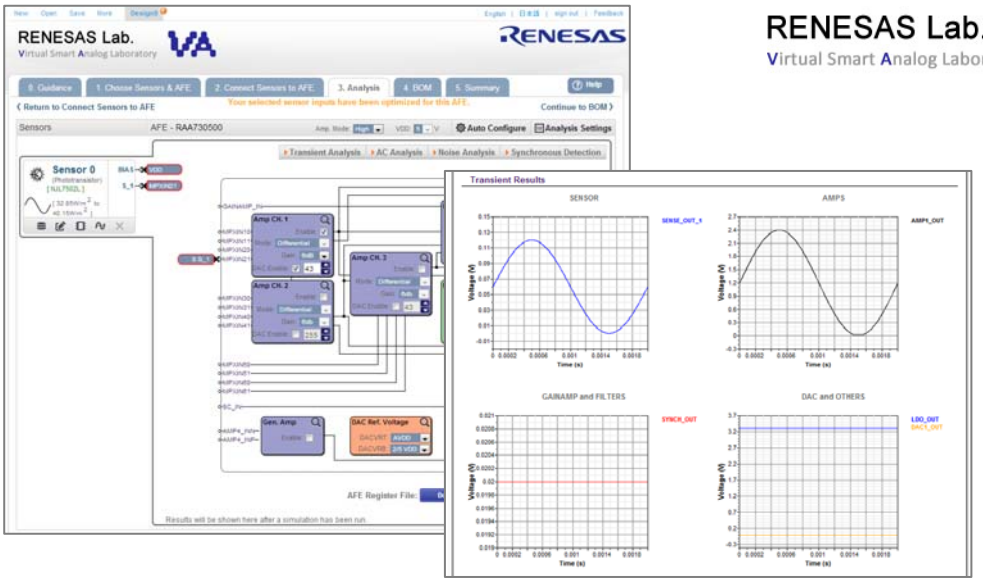
Actual system

4. Procedures

The development flow described in this document is described below.

4.1 Simulating the sensor operation by using Renesas VA

Simulate the sensor operation by using a simulator (Renesas VA).



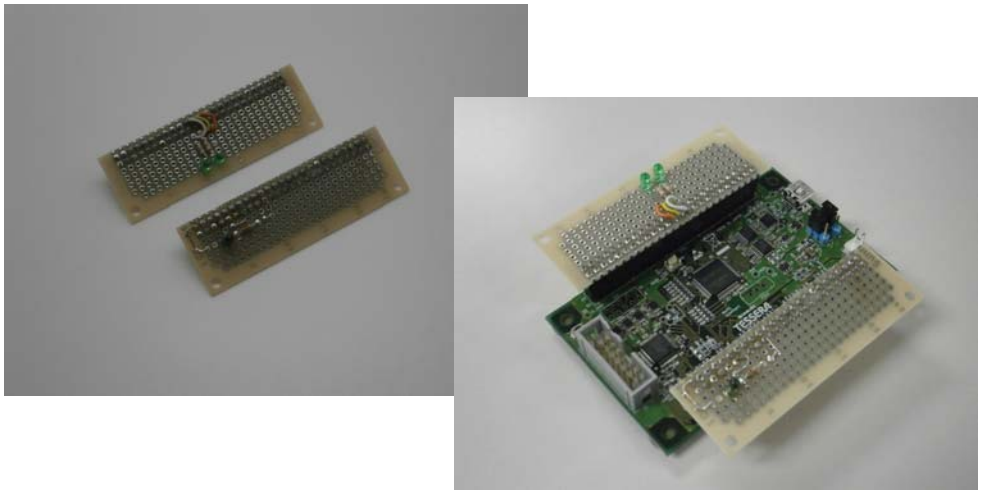
RENESAS Lab.
Virtual Smart Analog Laboratory

Result:

- Simulation result
- Smart Analog circuit parameters (AFE register file)

4.2 Creating an evaluation board

Create two boards, one to mount the sensor and one to mount the LEDs, and mount them on the evaluation board.

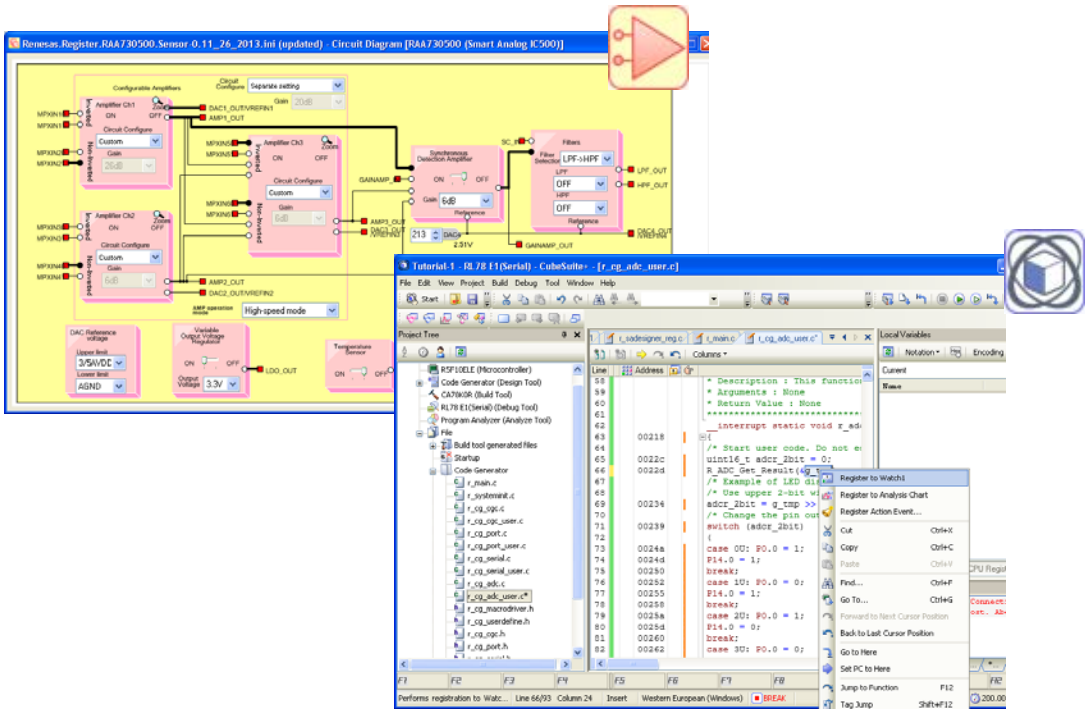


Result:

- LED board
- Sensor board

4.3 Reading a register file by using SA-Designer and loading the file to CubeSuite+ to create a program

Read the Smart Analog AFE register file created by Renesas VA, convert it to a C source file by using SA-Designer, load the parameters to CubeSuite+, and then create the program to drive the sensor and control the LEDs.

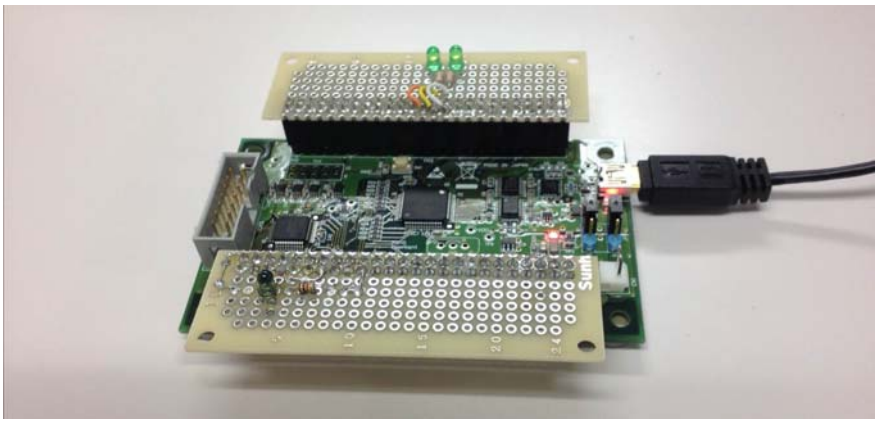


Result:

- Compiled program (load module)

4.4 Checking the operation on the evaluation board

Check whether the program created in section 4.3 operates as expected on the evaluation board.



Result:

- Sensor evaluation environment and program

5. Simulating the sensor operation by using Renesas VA

First, simulate the operation of the sensor and the Smart Analog IC by using Renesas VA, a web simulator provided by Renesas.

5.1 What is a web simulator

A web simulator enables analog circuits that incorporate a sensor and Smart Analog IC to be simulated and designed just by using a web browser. You do not need to download and install a special program. You can access the tool and simulate the operation any time you want, as long as you can connect to the Internet.

Renesas VA provides the following features:

- Circuit design
  You can combine a Smart Analog IC and a sensor to design a Smart Analog internal circuit.
- Transient analysis
  You can check the waveform generated from the analog signal converted by Smart Analog from the input sensor signal.
- AC analysis
  You can check the amplitude frequency characteristics of the waveform processed in the Smart Analog IC.
- Filtering effect analysis
  You can check the Smart Analog filter characteristics.
- Synchronous detector operation analysis
  You can check the operation of the synchronous detector incorporated in the Smart Analog IC.
- Reading and saving simulation details, and uploading and downloading the AFE register file
  You can read and save simulation results, and upload and download the Smart Analog AFE register file.

5.2 Using Renesas VA

Next, it is time to use Renesas VA.

Launch your web browser, and then access the Renesas website to obtain Renesas VA.

- Renesas VA

  http://www.renesas.com/renesas_va
  Renesas home page → **Products** → **Smart Analog** → **Renesas VA**

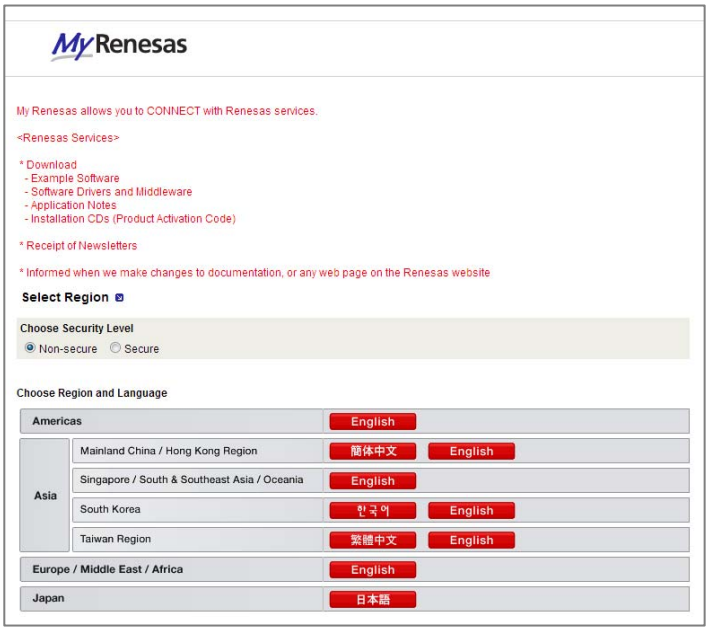
Click **Try Now** on the top right of the screen.

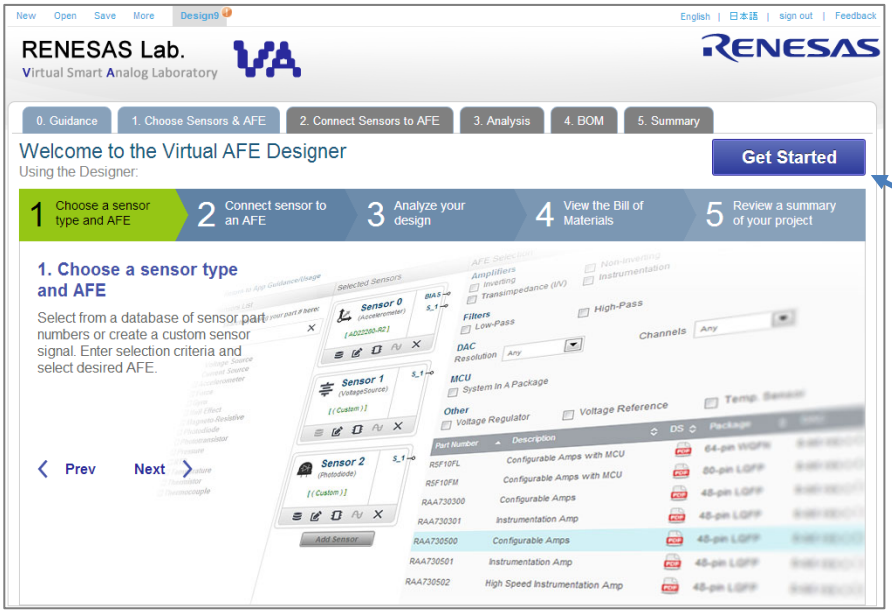
Click **Agree** on the bottom of the disclaimer screen.

A screen for inputting your My Renesas ID and password appears.

To use Renesas VA, you need a My Renesas ID and password. If you do not have an ID, click **New user? Click here to register now** to create an ID (free of charge).

Note At registration, if you select "Smart Analog" from **Category** under **Subscribe to content / service**, Smart Analog is added to your service list. You will then be able to receive newsletters related to Smart Analog products such as Renesas VA and SA-Designer.

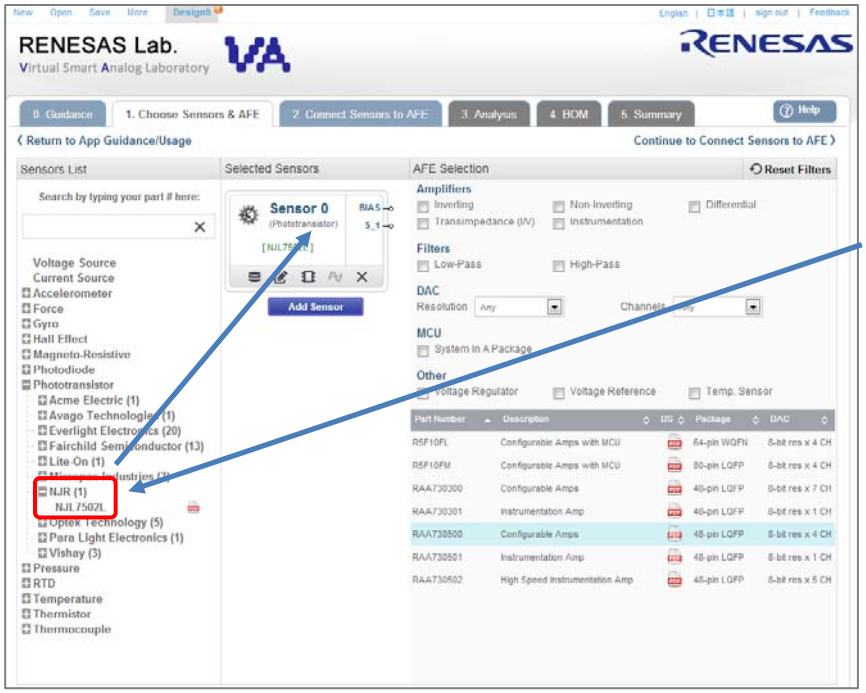




Click here.

Type your My Renesas ID and password. The Renesas VA screen appears.

Click **Get Started** on the top right of the screen.

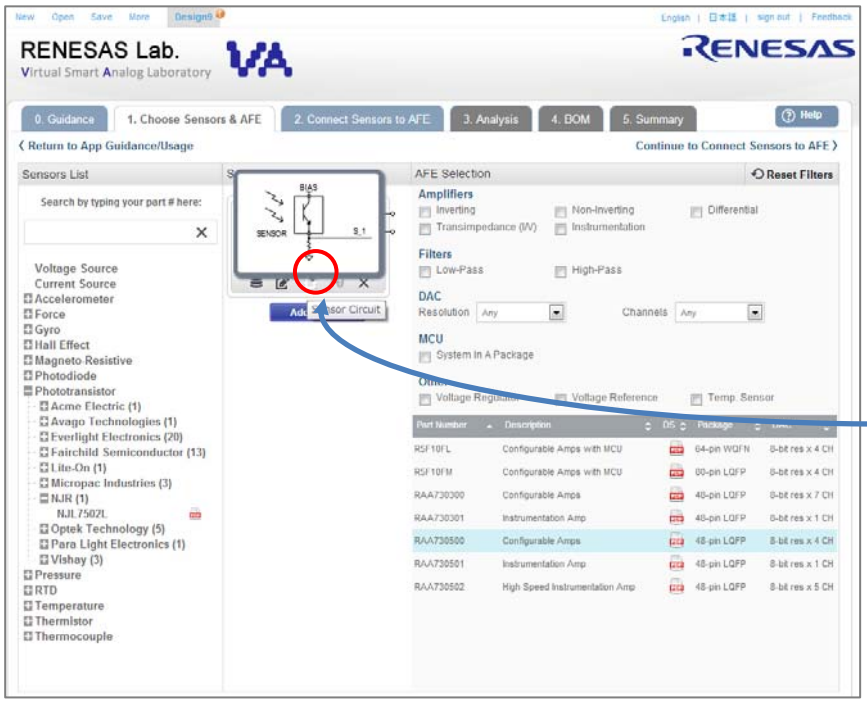


In the **Sensors List** area, click **Phototransistor**, **NJR**, and then drag **NJL7502L** into the **Selected Sensors** area.

Part #: NJL7502L
Manufacturer: NJR
Description: Ambient Light Sensor
Intensity (W/m2): 1n to 73
λ^P: 5.6e-007

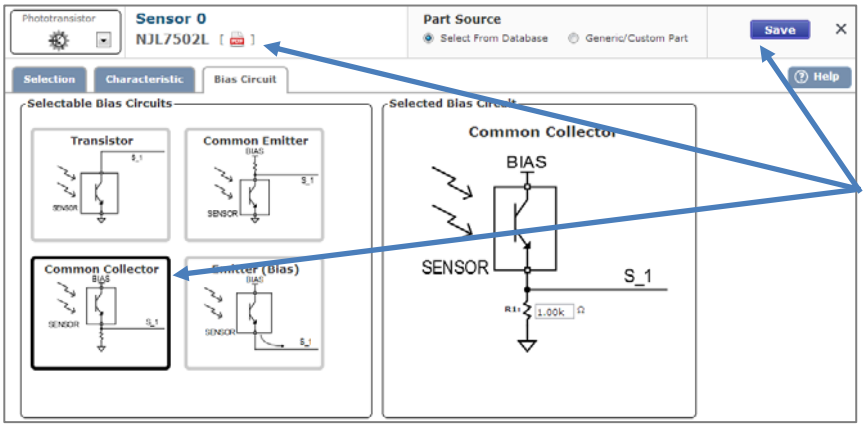
Select the sensor to be connected to the Smart Analog IC.

Here, you are using a phototransistor, so click **Phototransistor** to expand the tree, click **NJR** to expand the tree, and then drag **NJL7502L** into the **Selected Sensors** area to specify the model number.



Click **Sensor Circuit** to select the bias circuit.

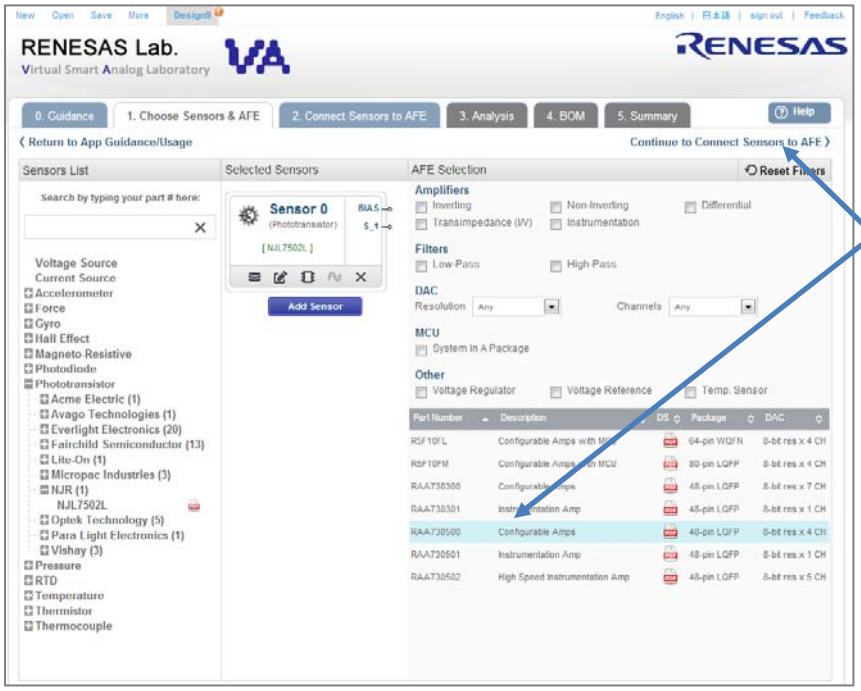
Select how to connect the sensor to the Smart Analog IC.



Click this icon to download the data sheet.
Select **Common Collector** as the bias circuit type, and then click the **Save** button on the top right of the screen.

The PDF icon next to **NJL7502L** on the top left of the screen is linked to the NJL7502L data sheet. Click this icon to download the data sheet.

There are four types of bias circuit connections available for NJL7502L and the Smart Analog IC. Here, select the common collector type, which allows you to obtain the widest dynamic range when the circuit is driven at 5 V.

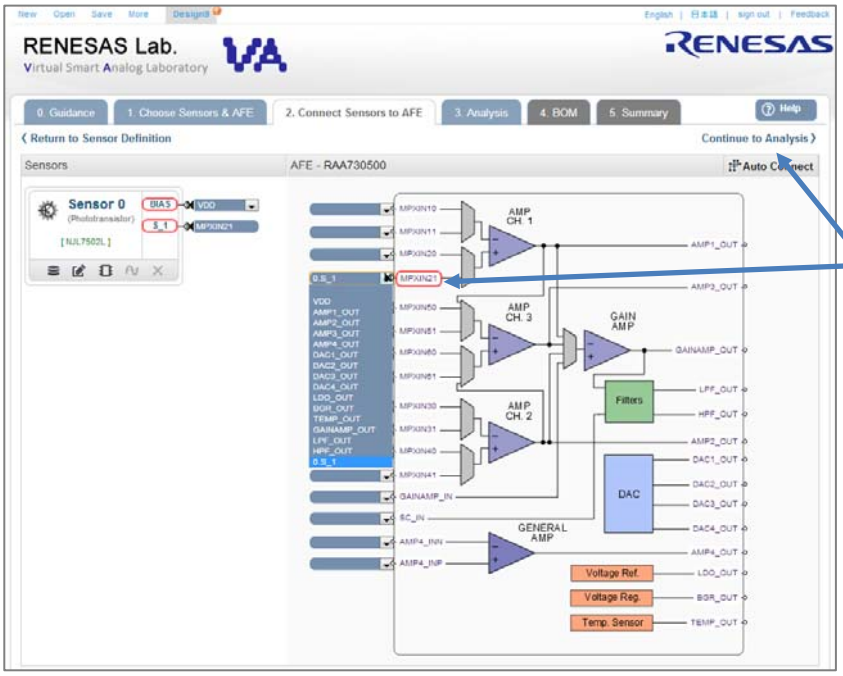


Select "RAA730500" under **Part Number**, and then click the **Continue to Connect Sensors to AFE** link on the top right of the screen.

Select the Smart Analog product to use for simulation.

The current version of Renesas VA supports Smart Analog MCU (SiP that packages an AFE and MCU) and Smart Analog IC (AFE only).

Here, you are using an evaluation board on which a Smart Analog IC (AFE only) is mounted, so select a Smart Analog IC (RAA730500).



From the drop-down list for the MPXIN21 pin, select "0.S_1" as the signal input from the sensor, and then click the **Continue to Analysis** link on the top right of the screen.

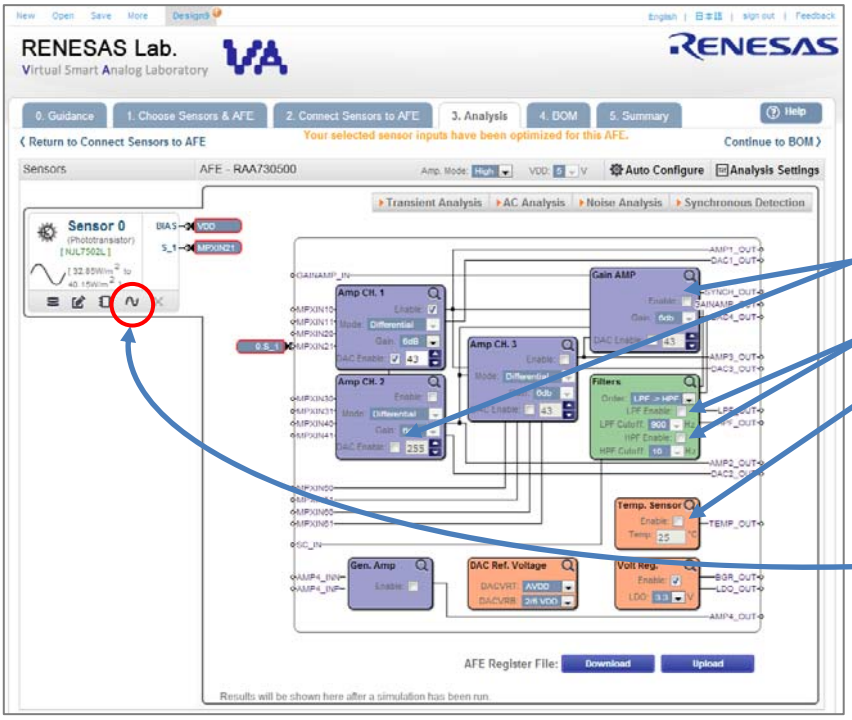
A screen that shows the connection of the sensor and Smart Analog IC appears.

The sensor is connected on the evaluation board in the later steps according to the connection between the sensor and the Smart Analog IC specified here. Although the sensor is automatically connected to the MPXIN20 pin on the screen, expand the drop-down list for the MPXIN21 pin and select "0.S_1" as the signal input from the sensor so that the sensor is connected to MPXIN21.

Here, when you attempt to change the sensor pin from MPXIN20 to MPXIN21, the message below will be displayed. Click **OK**.

Message output:
Custom sensor to AFE connections may not be supported by the Auto Configure function. To return to default connections select "Auto Connect".

Click here.



For **Amp CH.2**, clear the **DAC Enable** check box.
For **Gain Amp**, clear the **Enable** and **DAC Enable** check boxes.
For **Filters**, clear the **LPF Enable** and **HPF Enable** check boxes.
For **Temp Sensor**, clear the **Enable** check box.

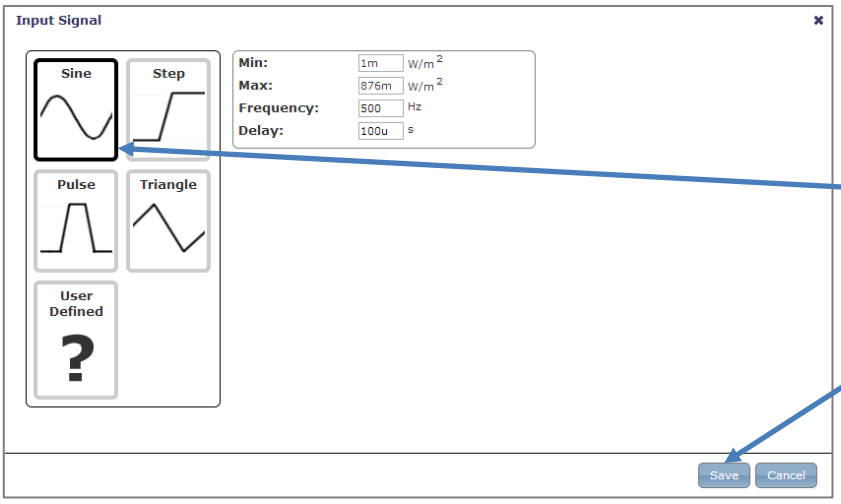
Click **Sensor Input**.

On this screen, configure settings such as the Smart Analog internal circuits, external signals input to the sensor, and the simulation type.

The default values are already set as the Smart Analog IC constants. You can start simulation with these settings. Here, to build the simplest configuration, use Amp CH.1 only. Clear the following check boxes that are selected by default: **DAC Enable** in **Amp CH.2**, **Enable** in **Gain AMP**, **DAC Enable** in **Gain AMP**, **LPF Enable** and **HPF Enable** in **Filters**, and **Enable** in **Temp. Sensor**.

The **Volt Reg** setting is used as the reference voltage for the A/D converter, so leave the setting as is (**Enable** selected).

Next, select the signal to input to the sensor.



For **Input Signal**, select "Sine".
Min: **1m** W/m^2
Max: **876m** W/m^2
Frequency: **500** Hz (default)
Delay: **100u** s (default)

After specifying the above, click the **Save** button on the bottom right of the screen.

Verify the settings by using the sine wave.

Here, specify 876 mW/m^2 for the maximum power of the signal input to the sensor, based on *Relationship between W/m2 and lux (Lx)* below.

Relationship between W/m^2 and lux (Lx)

The unit of the signal input to the photodiode (used as the sensor in this procedure) is defined by using W/m^2.

To make this easier to understand, we will convert the unit.

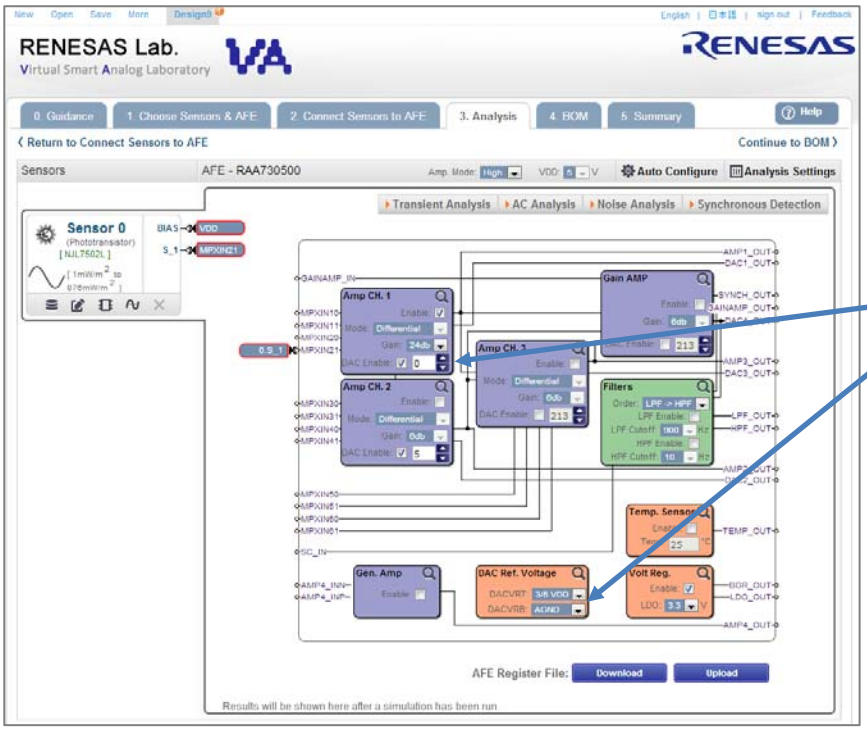
Lux (Lx) is generally used to indicate the luminance of light. The relationship between the lux (Lx) and W/m^2 is expressed as follows:

1 Lx = 1.46 mW/m^2 Note

Therefore, estimate the luminance in lux first and then convert it to W/m2 by using the above formula.

Note To be exact, the condition "where the middle wave length of the visible light spectrum is 555 nm" applies to this formula.

The luminance in standard homes and offices is approximately 150 to 400 lux. (The value differs depending on the location.) In this tutorial, holding your hand above the sensor to shut down the room light is tested as the signal input condition, so estimate the power of the signal input to the sensor to be 400 lux (Lx) = 584 mW/m^2. To make allowances for the environmental conditions, specify 600 lux (Lx) = 876 mW/m^2 as the maximum value.

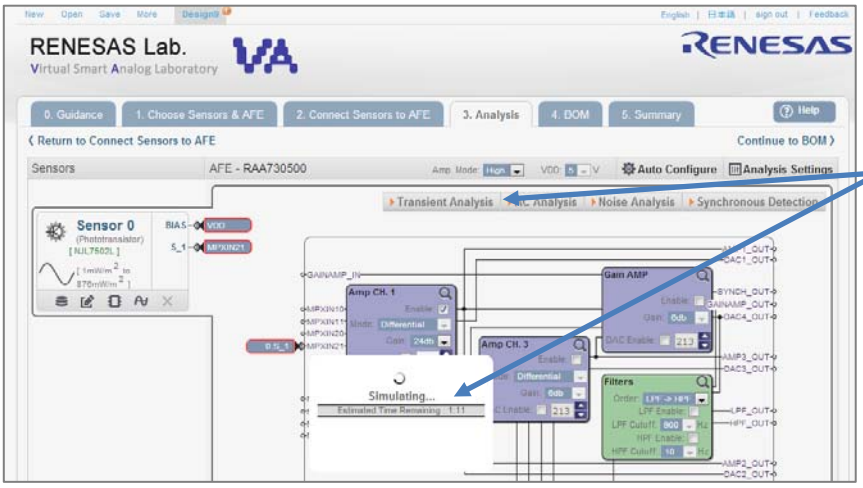


For **Amp CH.1**, set the DAC1 value to "0".
For **DAC Ref. Voltage**, set **DACVRB** to "AGND".

Finally, specify the Smart Analog amplifier gain and configure the bias circuit.

For **Gain** in **Amp CH.1**, use the value optimized by Renesas VA (24 dB) as is, and set the DAC1 value, which determines the bias voltage, to "0".

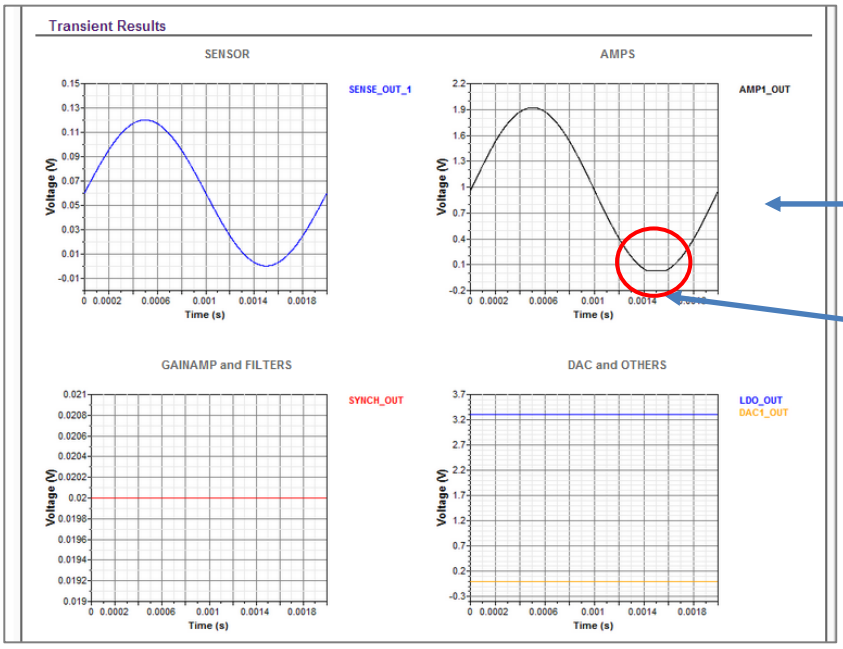
For **DAC Ref. Voltage**, set **DACVRB**, the lower limit for determining the D/A converter reference voltage, to "AGND".



Click the **Transient Analysis** button. Simulation starts and the estimated time it will take is displayed.

You have now finished setting up and can start simulation.

Click the **Transient Analysis** button on the center top of the screen to start simulation.



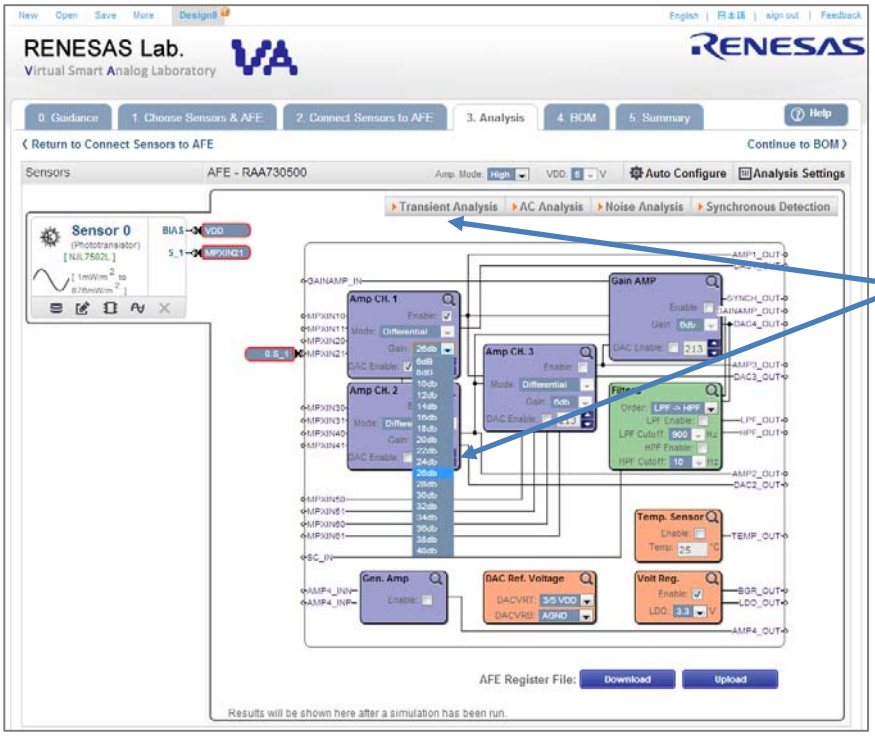
The waveform resulting from simulation is displayed on the bottom of the circuit screen.

Check that a signal of 0 V or lower is not output; that is, that the signal has been clipped.

When simulation finishes, the **Result** link to the simulation results appears immediately below the **Transient Analysis** button, and the waveforms resulting from simulation are displayed on the bottom of the circuit screen.

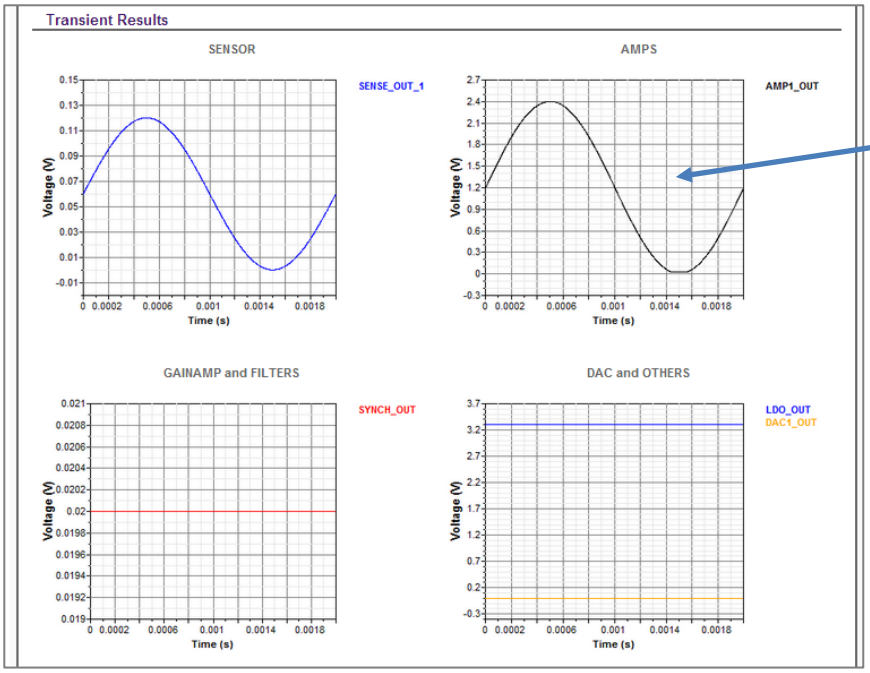
Based on the simulation results, you can see that the lower level of the AMP1 output is clipped. This is because the AMP1 bias voltage has been set to 0 V and thus the operating point of AMP1 is 0 V.

Based on the results, AMP1 output at 600 Lx is approximately 1.921 V, which seems to be low, so raise the gain.



For AMP1, set **Gain** to "26 dB". Click the **Transient Analysis** button.

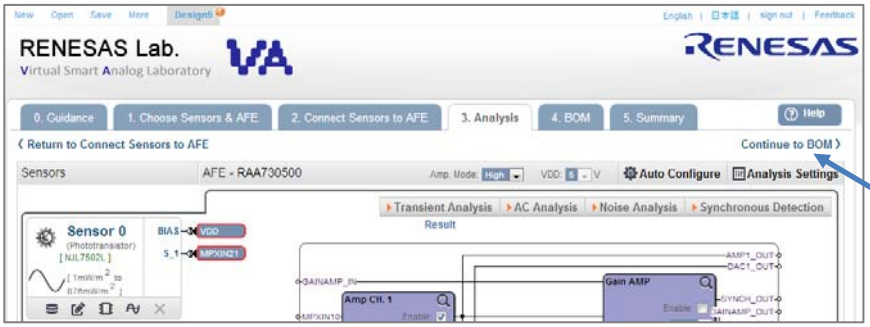
Set the gain for AMP1 to 26 dB, and then click the **Transient Analysis** button to perform simulation again.



Check the waveform resulting from simulation.

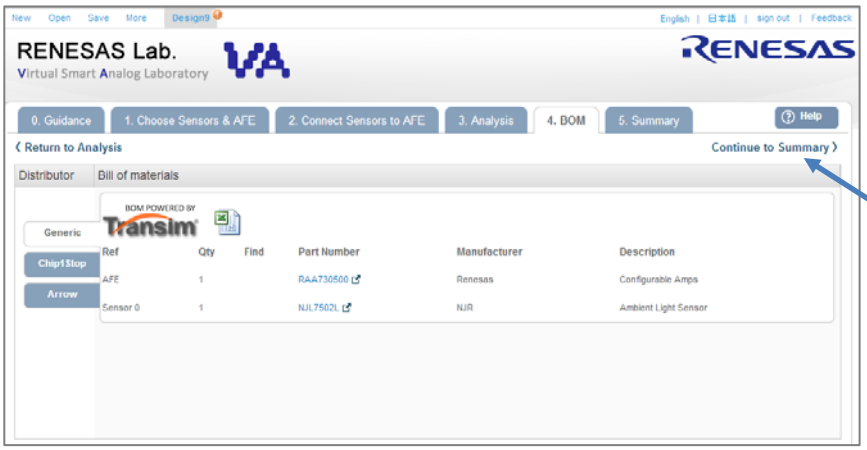
Based on the simulation results, you can see that the AMP1 output has been amplified to approximately 2.4 V.

You have confirmed that sufficient output is obtained, so use this parameter.



Click the **Continue to BOM** link on the top right of the screen.

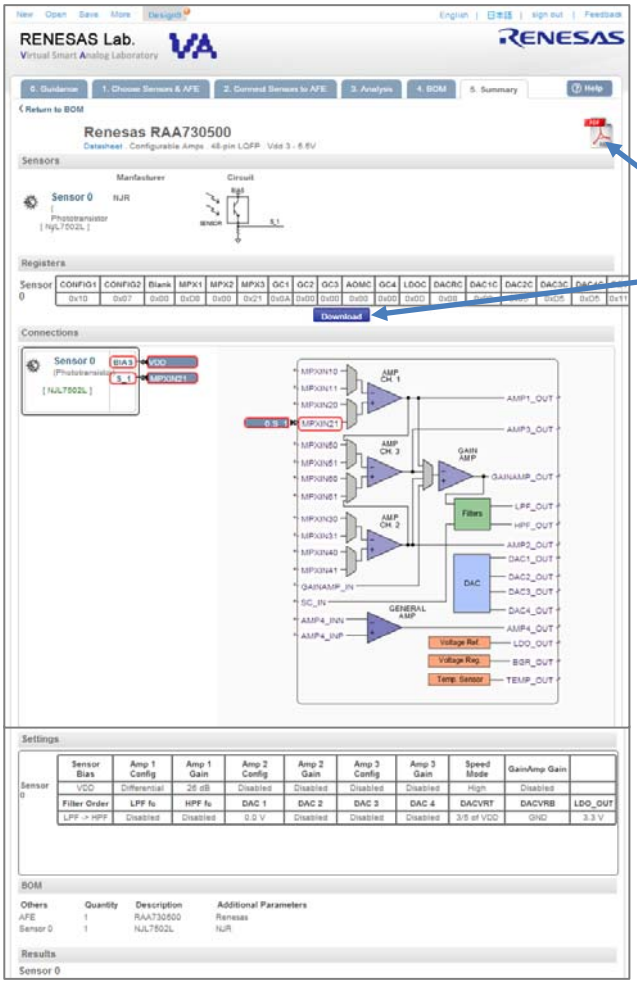
Click the **Continue to BOM** link on the top right of the screen to move to the next screen.



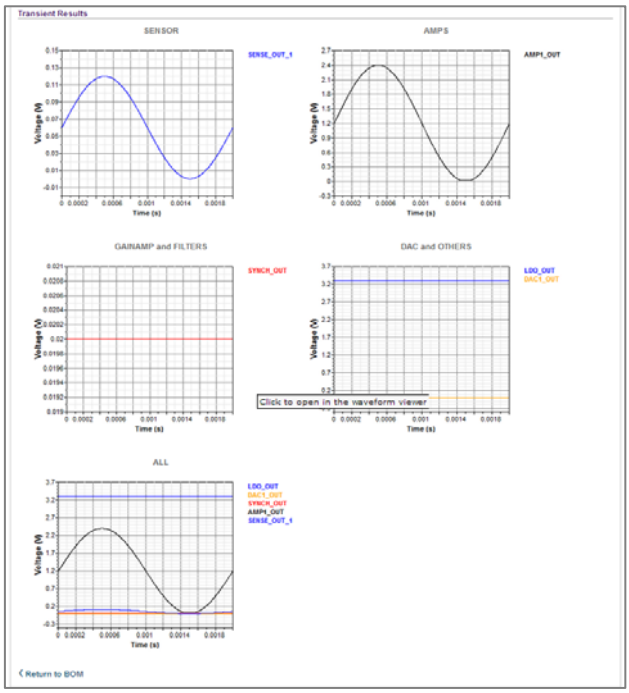
Click the **Continue to Summary** link on the top right of the screen.

This screen shows a list of parts used for this simulation.

Click the **Continue to Summary** link on the top right of the screen to move to the next screen.



Click **Download** to save the results.



The simulation results are displayed.

If you click the pdf icon on the top right of the screen, you can download `Renesas_RAA730500.pdf`, which contains the simulation results shown on this screen.

Under **Registers**, click the **Download** button. You can download and save `Renesas.Register.RAA730500.Sensor-XX_XX_XXXX.ini` on your local disk.

This file contains the Smart Analog circuit configuration parameters (AFE register file) that were obtained as a result of simulation.

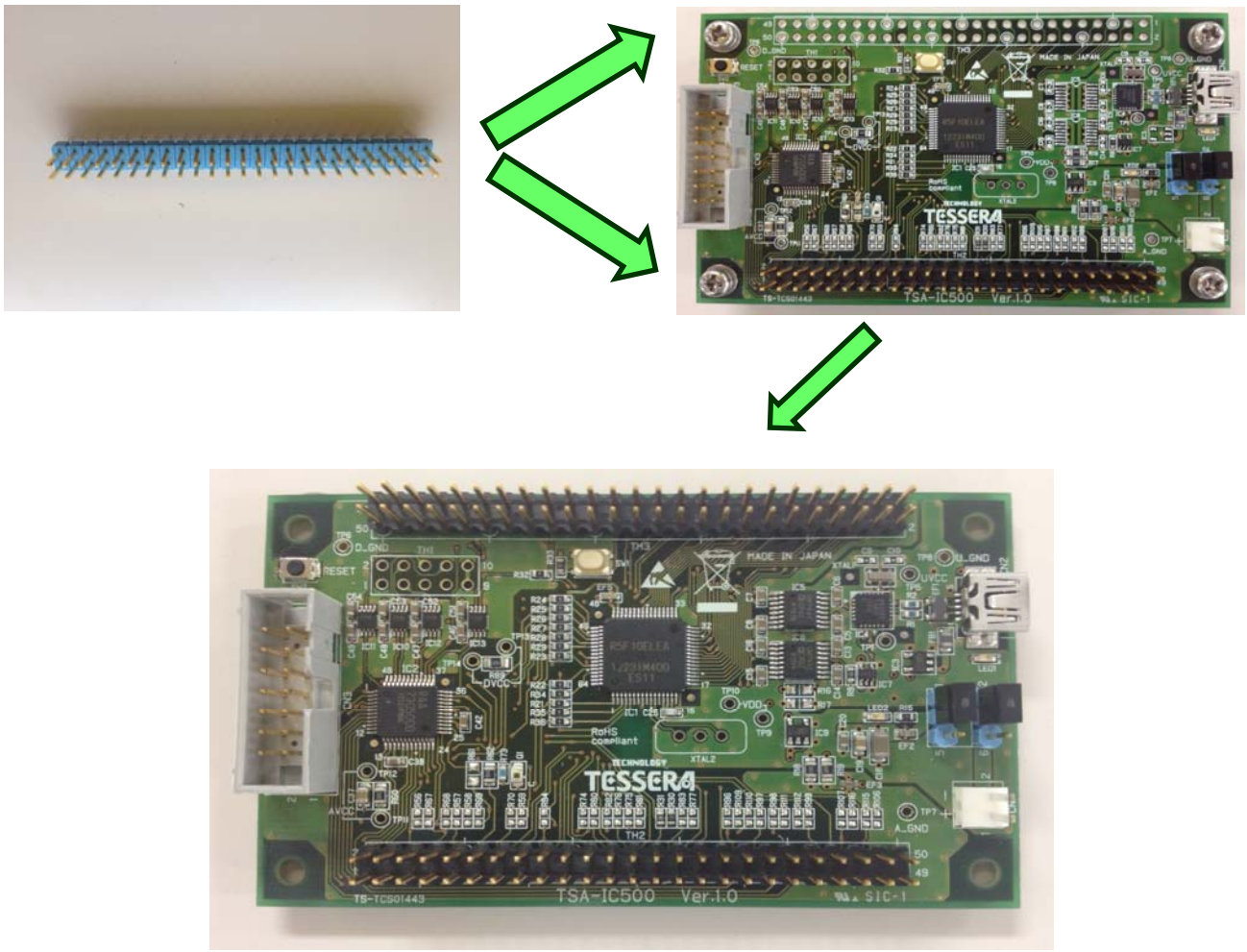
Simulation is now complete. Next, start development on the evaluation board.

6. Creating the sensor board and LED board

The following describes how to create the sensor board and LED board to be mounted on the evaluation board.

6.1 Soldering the 50-pin male headers (2 × 25 pins) on the evaluation board

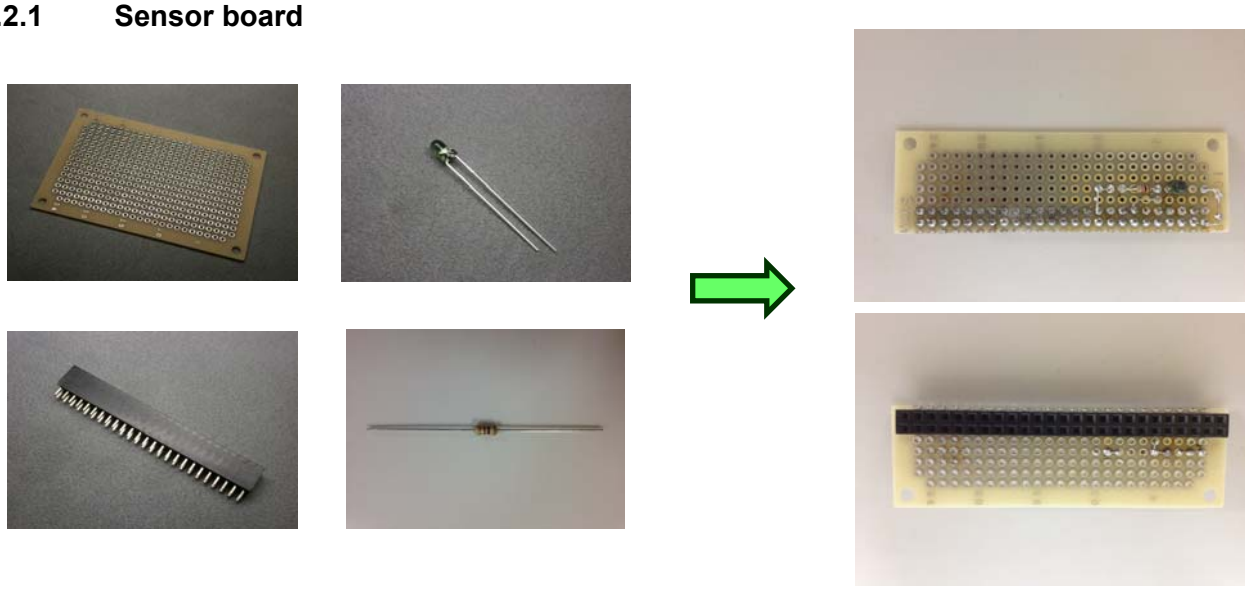
Solder the 50-pin male headers on the top and bottom of the evaluation board TSA-IC 500.



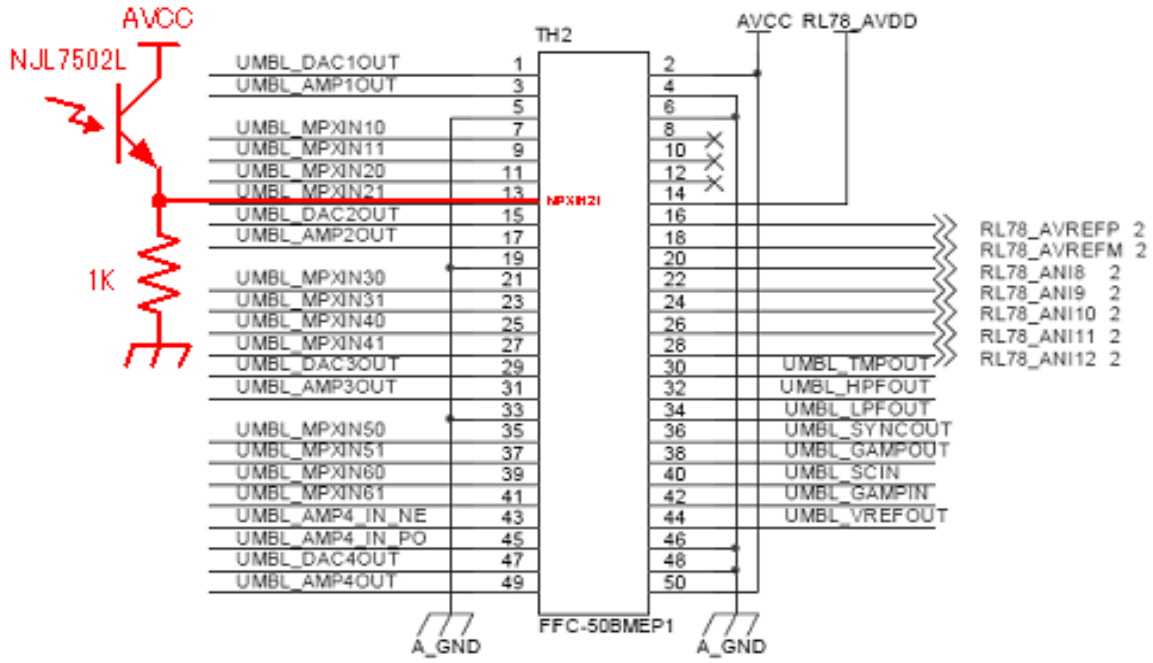
6.2 Creating the sensor board and LED board

Create the sensor board and LED board to be mounted on the evaluation board TSA-IC 500.

6.2.1 Sensor board

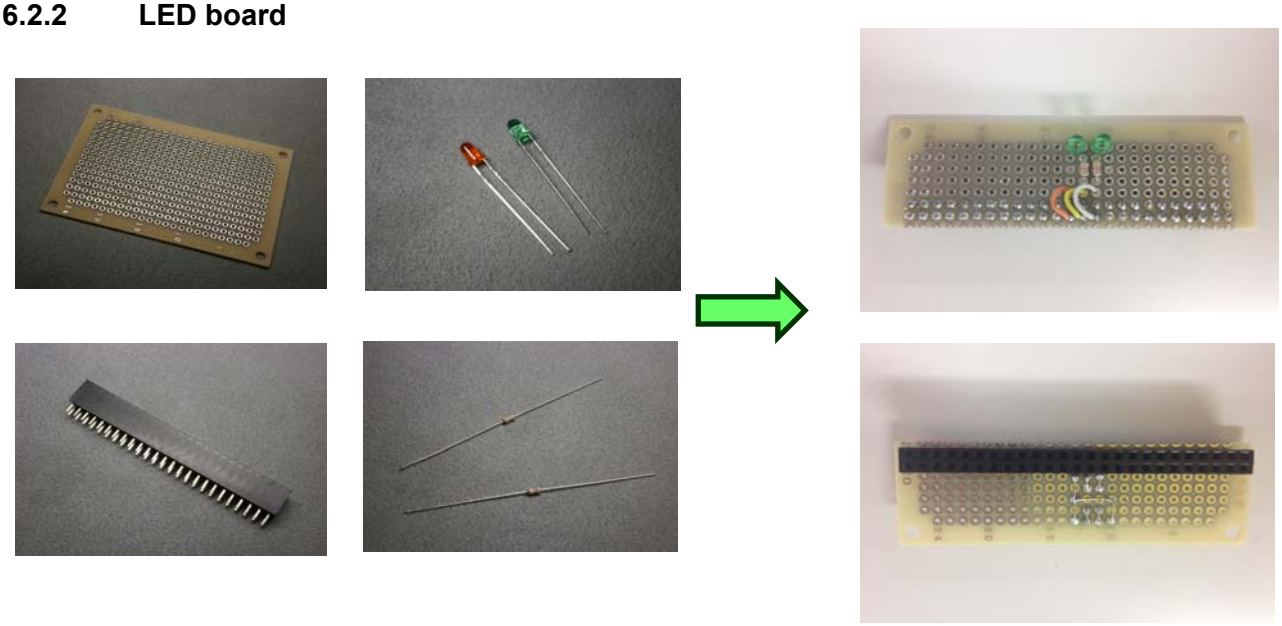


Solder the circuits on the sensor board as shown below.

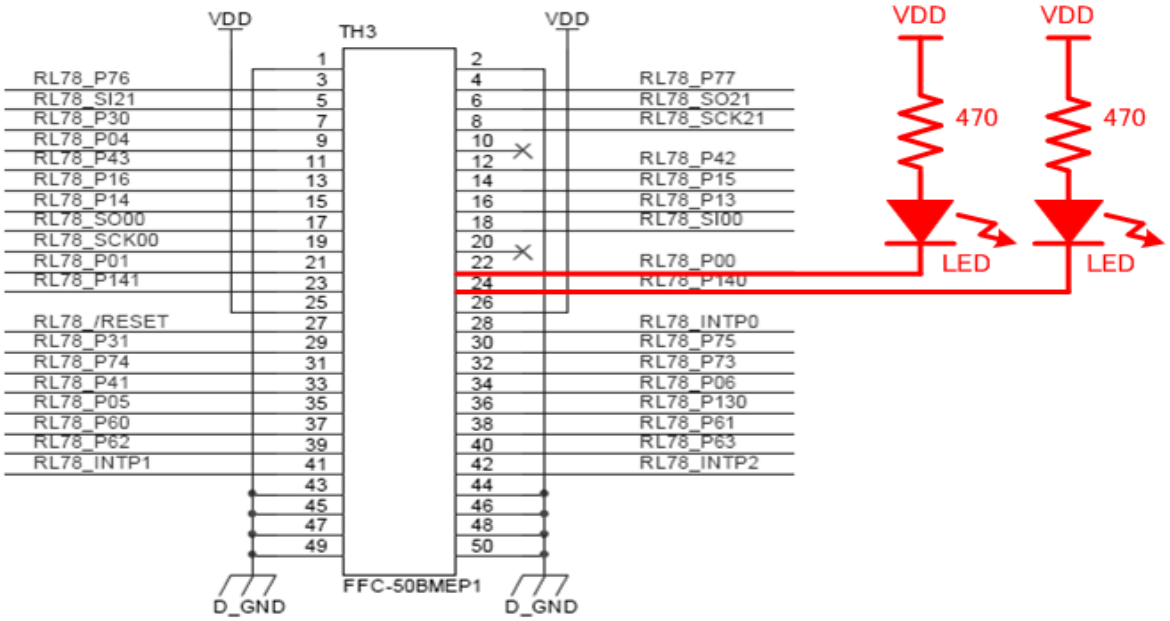


Sensor board circuit schematic

6.2.2 LED board



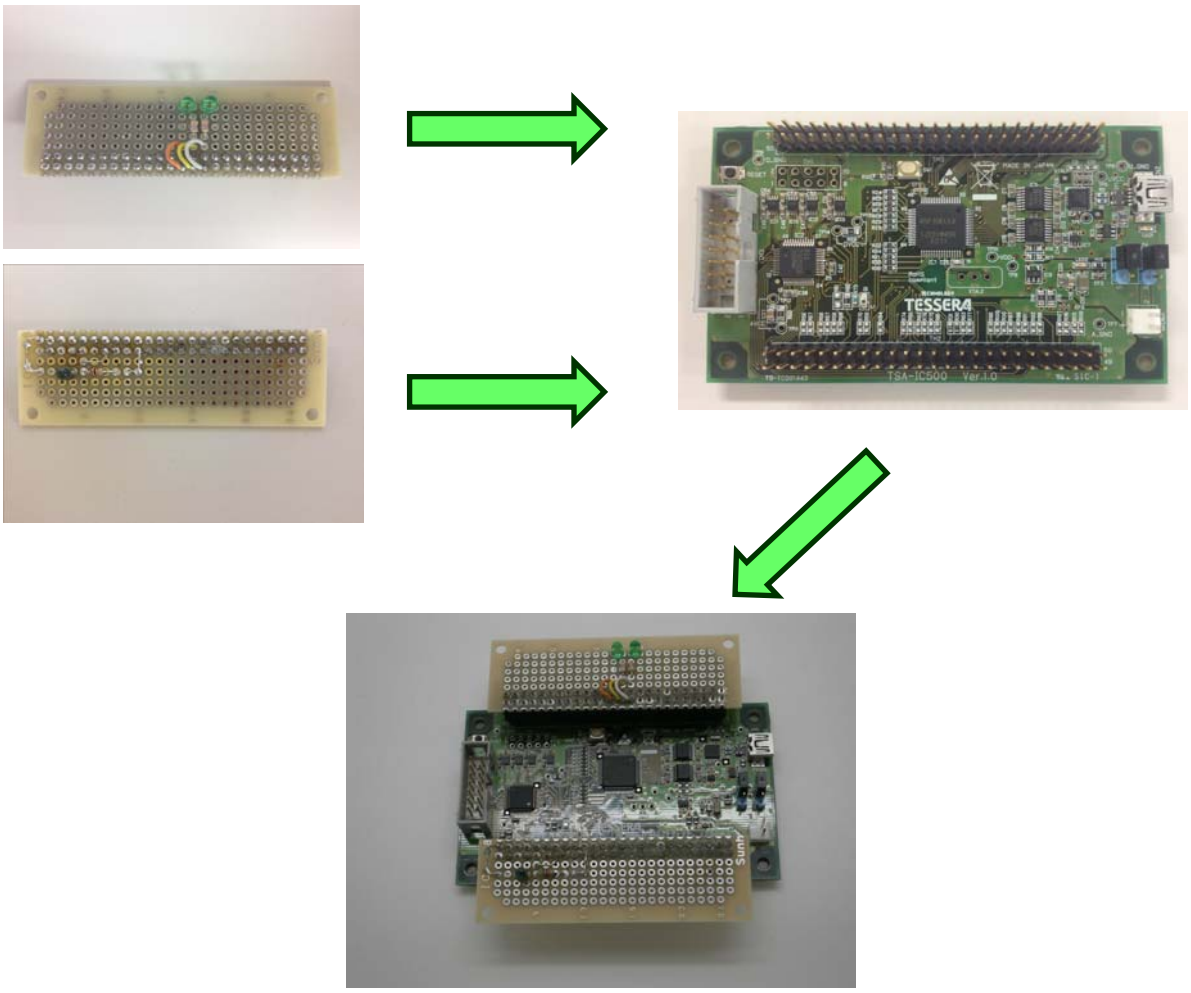
Solder the circuits on the LED board as shown below.



LED board circuit schematic

6.3 Mounting the sensor board and LED board on the evaluation board

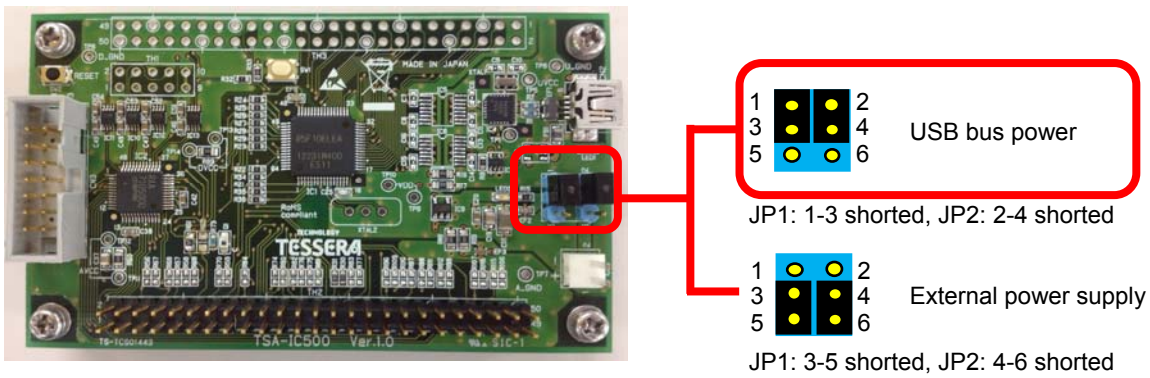
Mount the sensor board and LED board on the evaluation board.



Note Mount the LED board on the top side of the evaluation board (the side facing up in the photograph) and mount the sensor board on the bottom side.

6.4 Configuring the power supply on the evaluation board

You can select an external power supply or USB bus power as the power supply for the evaluation board TSA-IC 500. Here, we are using USB bus power, so check that the jumper pins on the TSA-IC 500 are set as below.



Creation of the evaluation board is now complete.

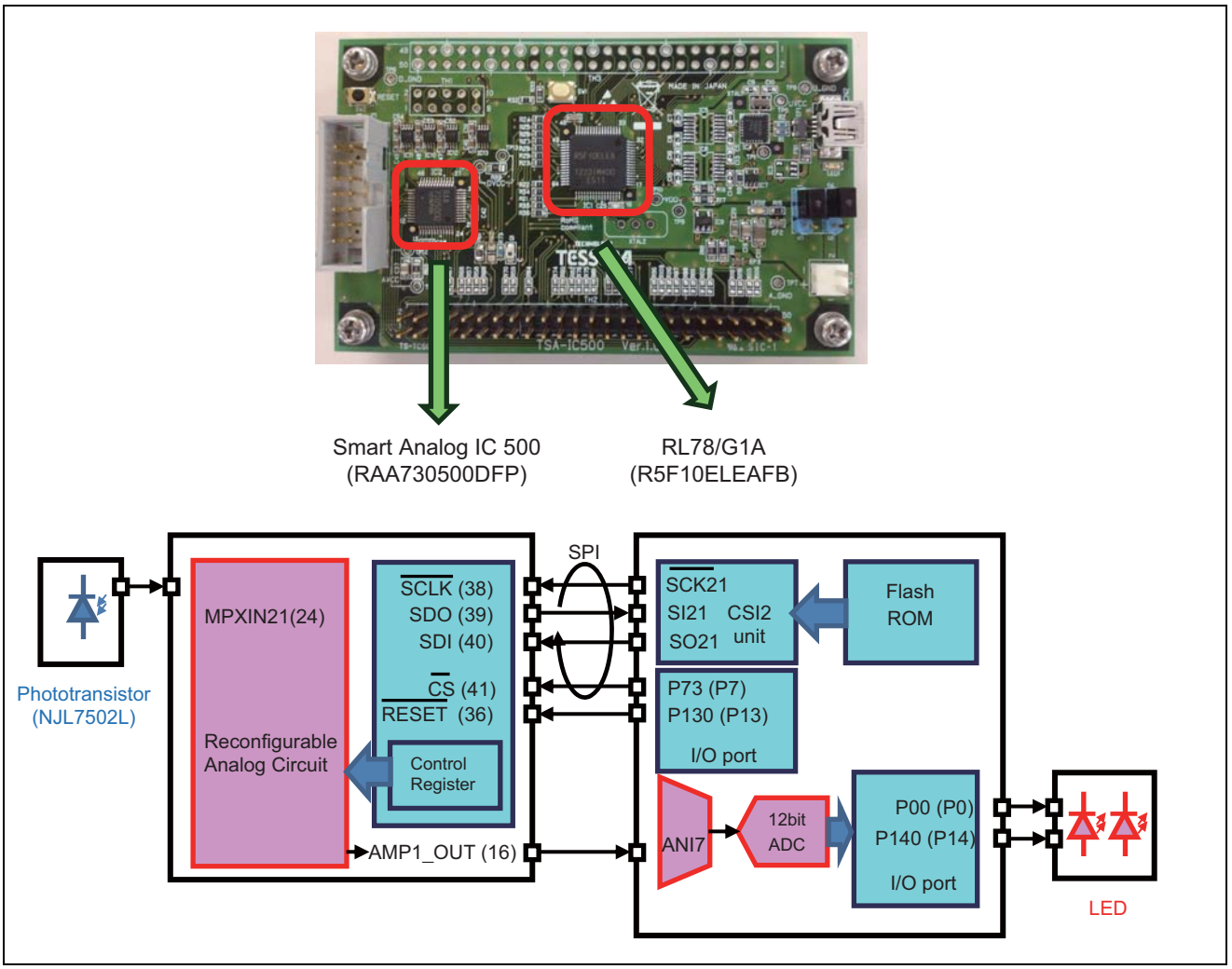
Next, create a program by using this evaluation board.

7. Reading the AFE register file by using SA-Designer and creating a program by using CubeSuite+

Create a program that runs on the evaluation board by using the AFE register file for the Smart Analog IC created in section 5 and the sensor board, LED board, and evaluation board created in section 6.

7.1 The IC mounted on the evaluation board, connection to the microcontroller, and the program to create

Before developing a program, note the following concerning the Smart Analog IC mounted on the evaluation board, the microcontroller configuration, and the pins.



The figure above shows the connection of the Smart Analog IC 500 (RAA730500DFP) and RL78/G1A (R5F10ELEAFB) pins and a diagram of the internal circuits. Smart Analog IC 500 is connected to and controlled by RL78/G1A via the SPI.

Copy the register values in the AFE register file to the flash memory in the microcontroller, and create a program that drives the Smart Analog IC by transferring the register values from the SPI module in the microcontroller to the Smart Analog IC.

As part of this program, create a program that lights the LEDs by controlling the LED port signals based on the sensor signals sent to the A/D converter in RL78/G1A by the Smart Analog IC and A/D converted.

7.2 Installing SA-Designer

Install SA-Designer before creating the program.

7.2.1 What is SA-Designer?

SA-Designer is used to design analog front-end circuits for Smart Analog products and output the circuit data as C source code.

In conventional analog front-end circuit development, it took a long time to prepare hardware because parts needed to be procured for each sensor operation or each time the parts were changed. Using Smart Analog and SA-Designer resolves this problem, because circuits can be designed simply by using a mouse on the SA-Designer screen.

Using SA-Designer, you can import AFE register files that are created by the web simulator Renesas VA and that can be used to select the sensor, and use SA-Designer in association with integrated development environment CubeSuite+. As a result, you can complete all processes including sensor selection, circuit design, and program development just by using software, enabling efficient system development.

SA-Designer provides the following features:

- Designing and customizing circuits using a circuit schematic GUI
- Listing, saving and restoring created circuit data (such as register values)
- Reading the circuit design (AFE register file) created by using Renesas VA
- Creating C source files for circuit design
- Registering projects to CubeSuite+ and building, downloading, and debugging using CubeSuite+
- Tutorial

7.2.2 Installing SA-Designer

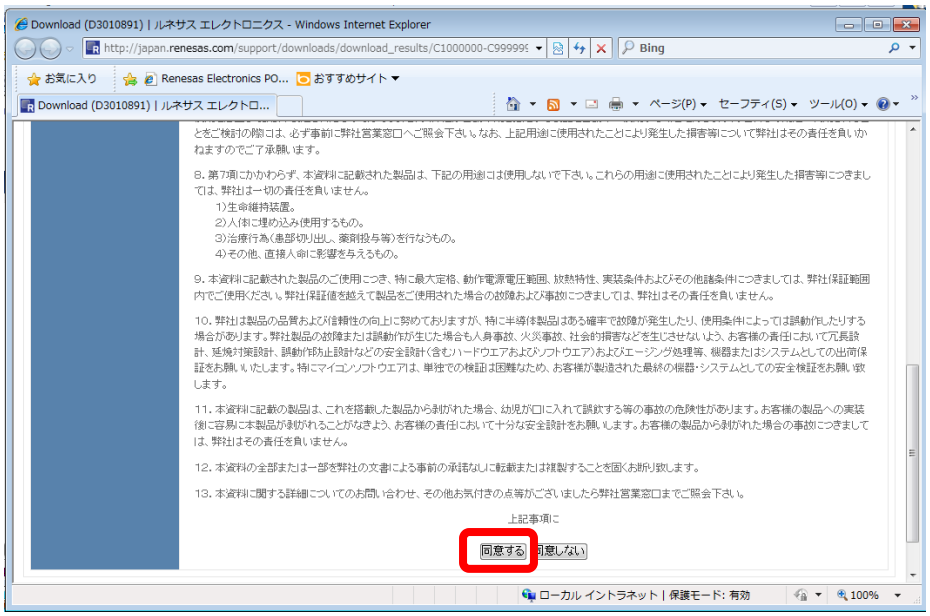
Launch your web browser, and then access the Renesas website > **Development Tools** > **Smart Analog Tools** > **SA-Designer**.

- SA-Designer

  http://www.renesas.com/sa_designer

  Renesas home page → **Development Tools** → **Smart Analog Tools** → **SA-Designer**

On the **Downloads** tab, click **Analog Front-End Design Tool SA-Designer V*x.xx.xx***.

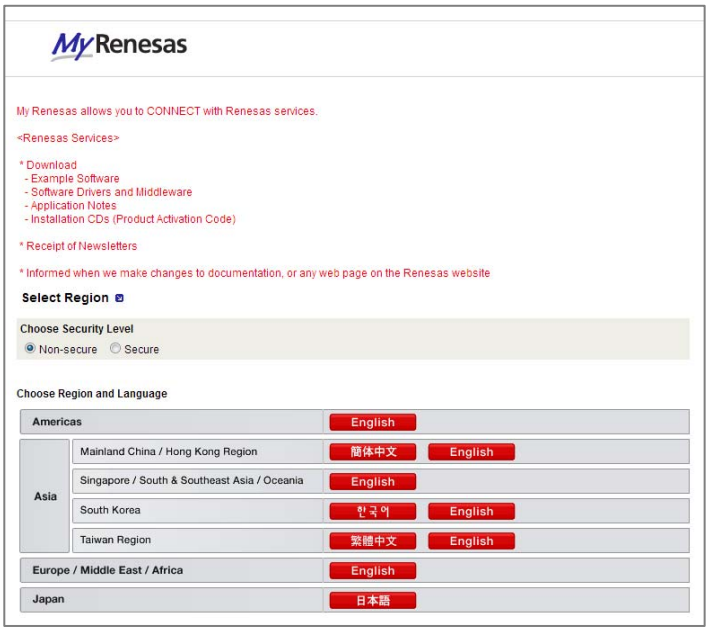


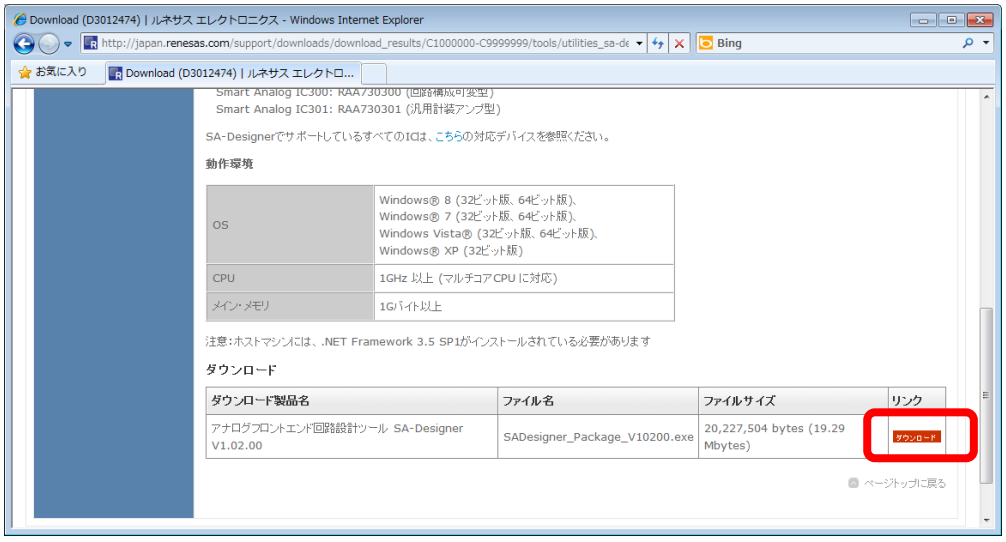
Click **Agree** on the disclaimer screen.

A screen for inputting your My Renesas ID and password appears. Input your ID and password.

If you do not have an ID, click **New user? Click here to register now** to create an ID (free of charge).

Note At registration, if you select "Smart Analog" from **Category** under **Subscribe to content / service**, Smart Analog is added to your service list. You will then be able to receive newsletters related to Smart Analog products such as Renesas VA and SA-Designer.

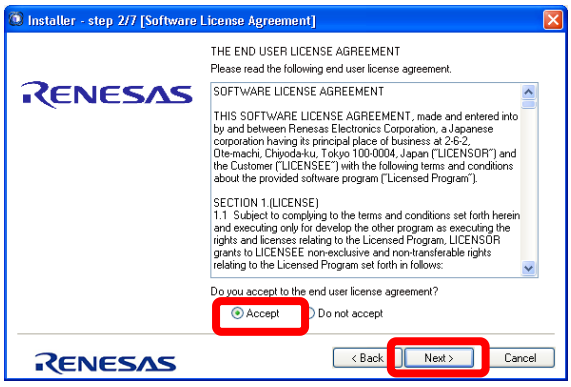




Click the **Download** button on the bottom right of the screen to download and save the file on your local disk.

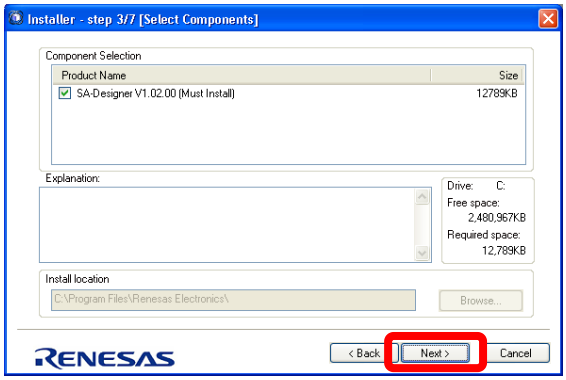
After you have downloaded the file `SADesigner_Package_V`*`XXXXX`*`.exe`, double-click it to launch the installer.

Click **Next**.

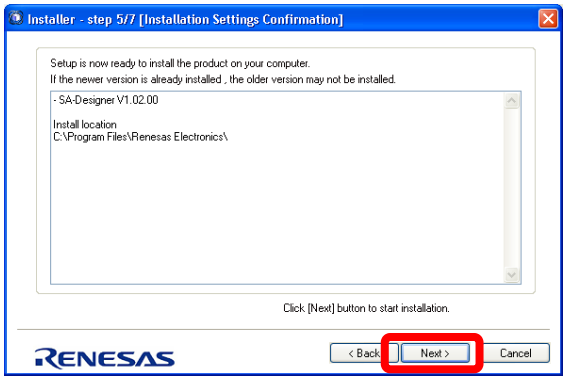


The software license agreement screen appears.

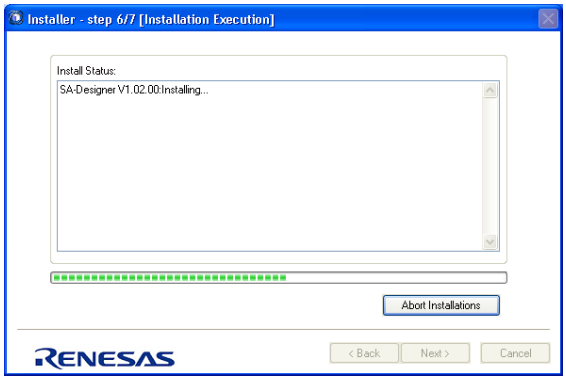
Click **Accept**, and then click **Next.**



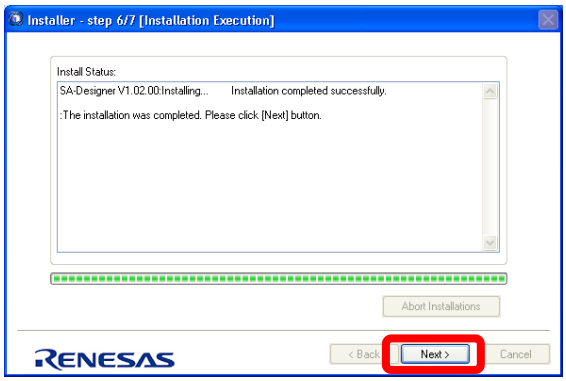
The location where SA-Designer is to be installed is displayed. Check the location, and the click **Next**.



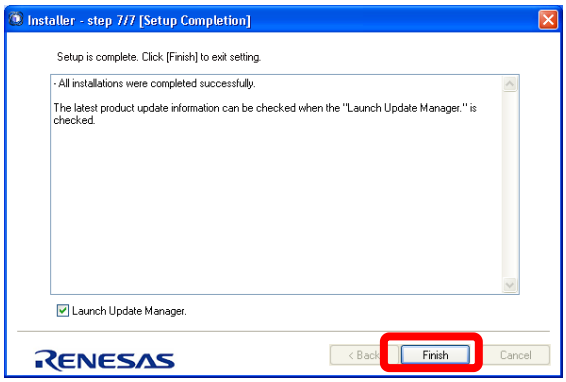
Click **Next**.



Installation starts.



Click **Next**.



Click **Finish**. This completes installation of SA-Designer.

7.3 Installing CubeSuite+

Installing CubeSuite+

If you have already installed CubeSuite+, skip this procedure and go to section 7.4.

Note If you are using a CubeSuite+ version earlier than Ver. 1.02, upgrade it the latest version.
To use CubeSuite+ in association with SA-Designer, you need to install CubeSuite+ Ver. 1.02 or later.

7.3.1 How to install CubeSuite+

See the following webpage for the features of CubeSuite+ and the environments in which CubeSuite+ can be used.

- Integrated development environment CubeSuite+
  http://www.renesas.com/cubesuite+
  Renesas home page → **Development Tools → IDEs and Project Managers → CubeSuite+**

Click the **Downloads** tab, click **[Evaluation Software] CubeSuite+ V*x.xx.xx*** (Single Download) or (Multipart Download) to download and install CubeSuite+.

Note Install the latest version of CubeSuite+ (Ver. 1.02 or later).

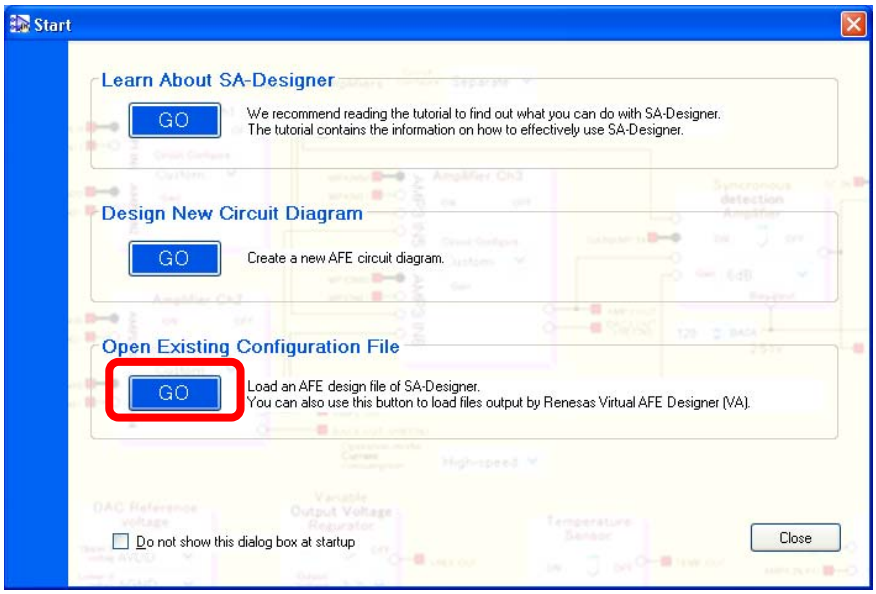
7.4 Starting SA-Designer

To start programming, start SA-Designer.

Start SA-Designer from the Start menu.

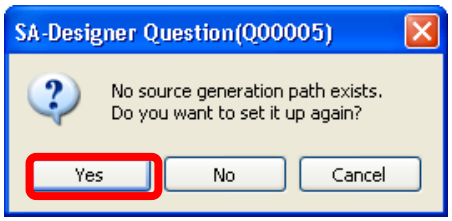
All Programs → Renesas Electronics Utilities → Smart Analog Tools → SA-Designer

On the SA-Designer **Start** screen, click **GO** under **Open Existing Configuration File**.

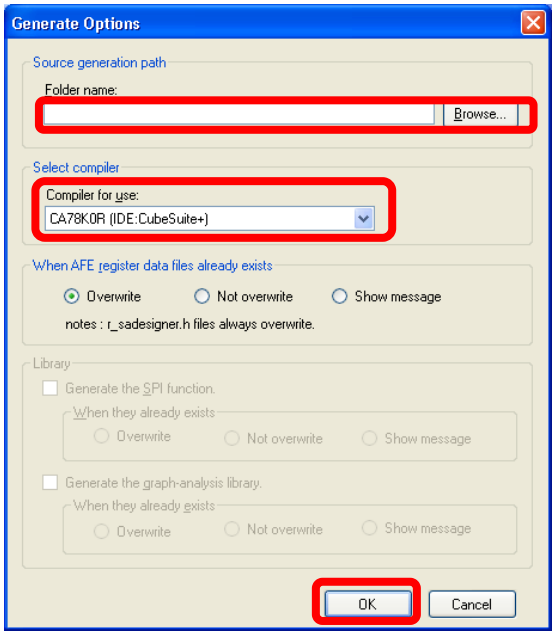


7.5 Reading the AFE register file (.ini file) generated by Renesas VA

Open the AFE design file (`Renesas.Register.RAA730500.Sensor-`*`XX_XX_XXXX`*`.ini`) obtained as a result of simulating the phototransistor NJL7502L by using Renesas VA.

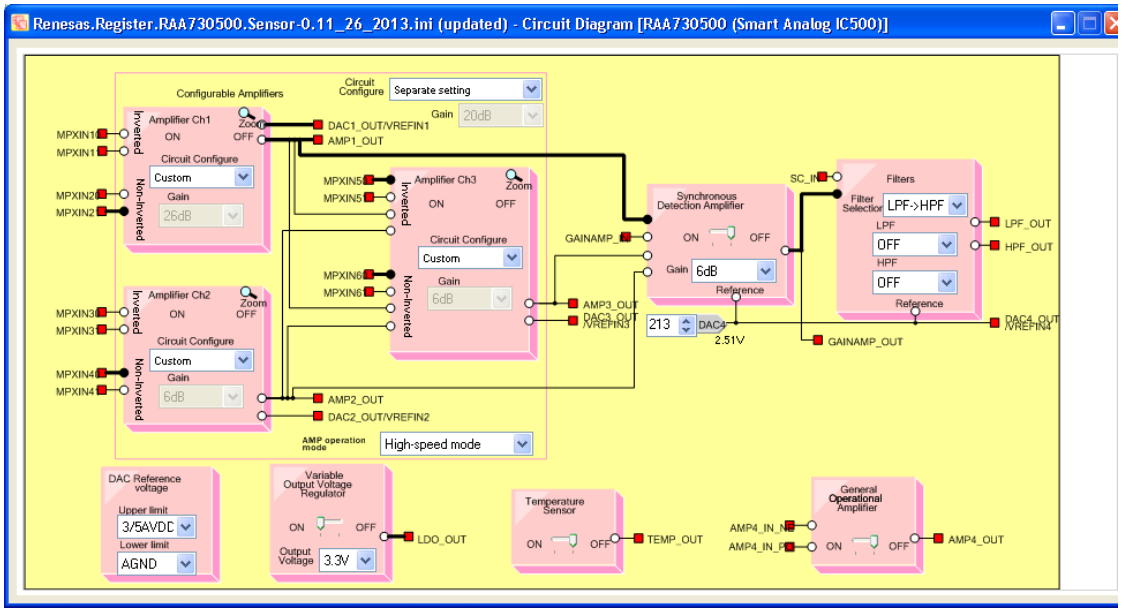


Click **Yes**.



In the dialog box that appears, specify the path for the directory in which the source file will be created. For **Compiler for use**, select "CA78K0R (IDE: CubeSuite+)", and then click **OK**.

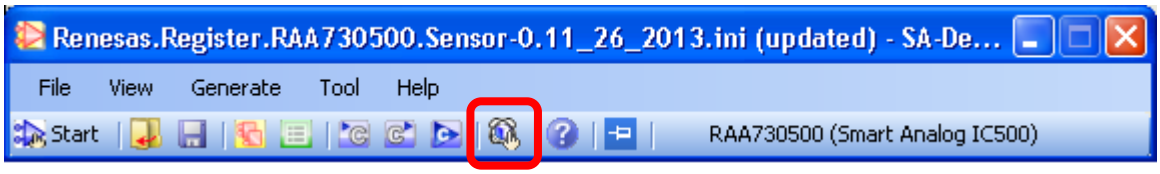
The SA-Designer circuit design screen appears.



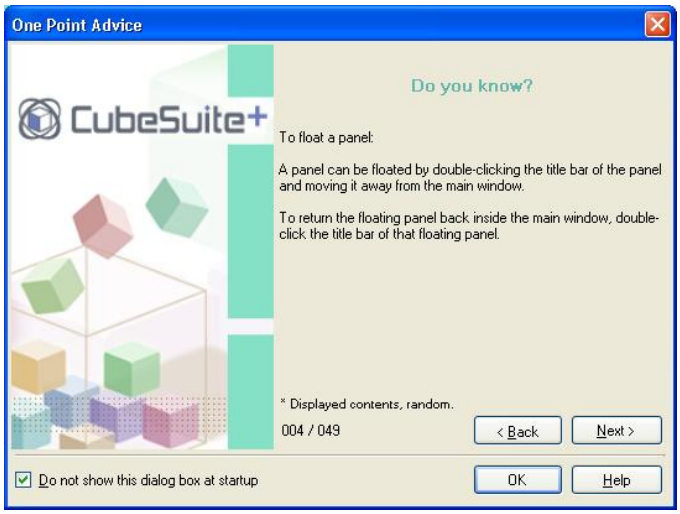
7.6 Starting CubeSuite+

Associate SA-Designer with CubeSuite+.

Click the button in the red frame in the figure below to start CubeSuite+.

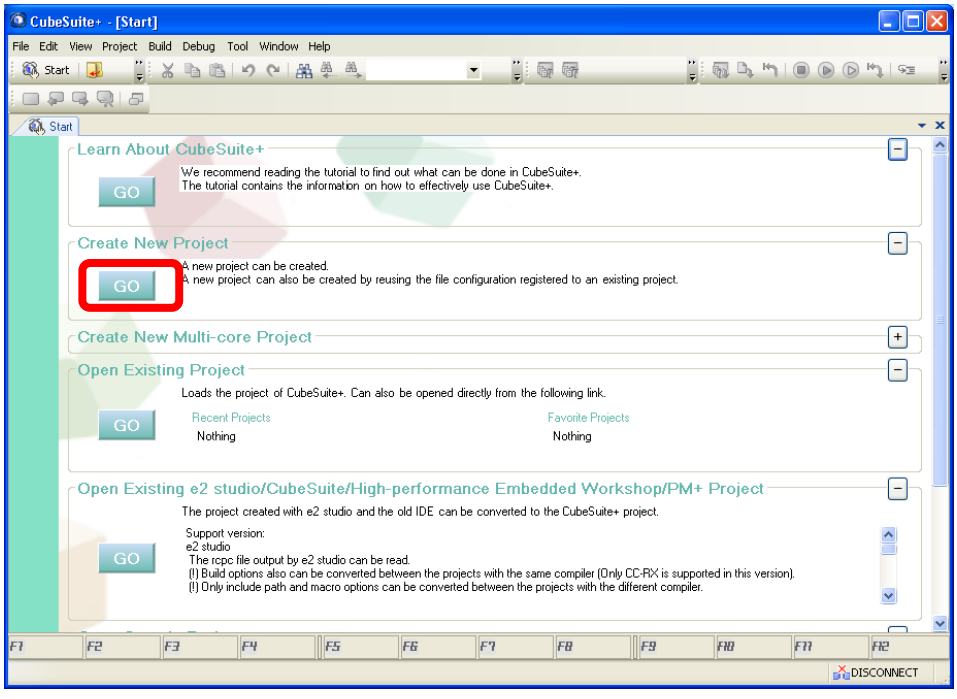


Create a project for the program described in this document.

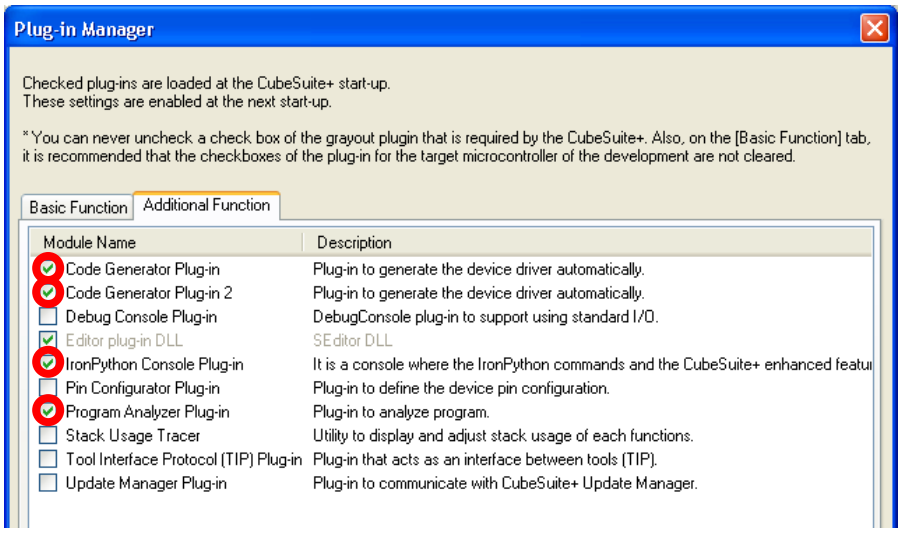


Click **OK**.

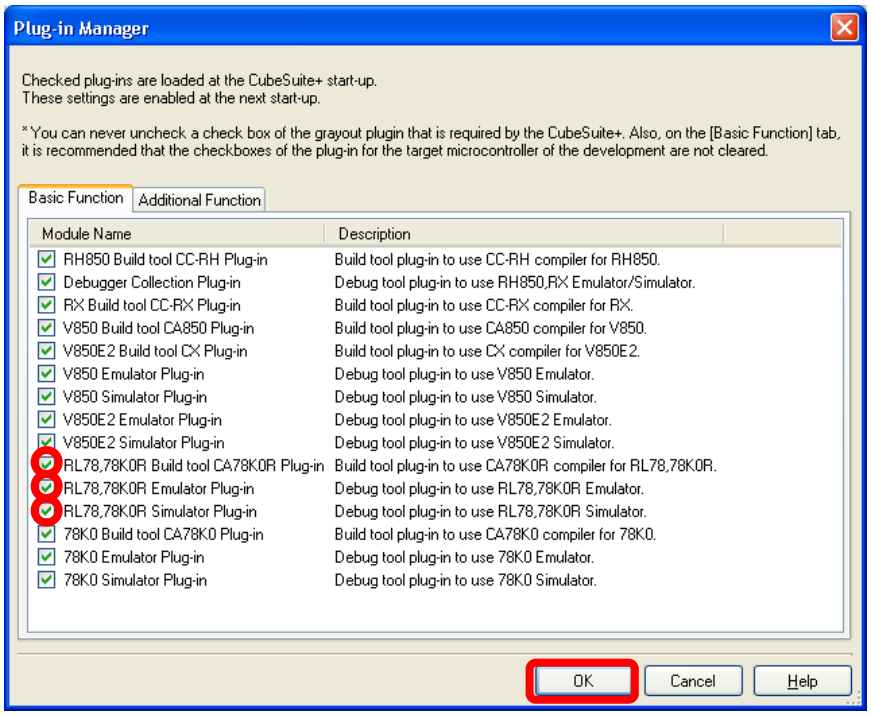
On the **Start** tab, click **GO** under **Create New Project**.



Note If the above start screen does not appear when you start CubeSuite+, the cause might be plugins. On the **Tool** menu, click **Plug-in Setting**, and then configure plugins as shown below.
If the start screen appears, go to the **Create Project** screen on the next page.



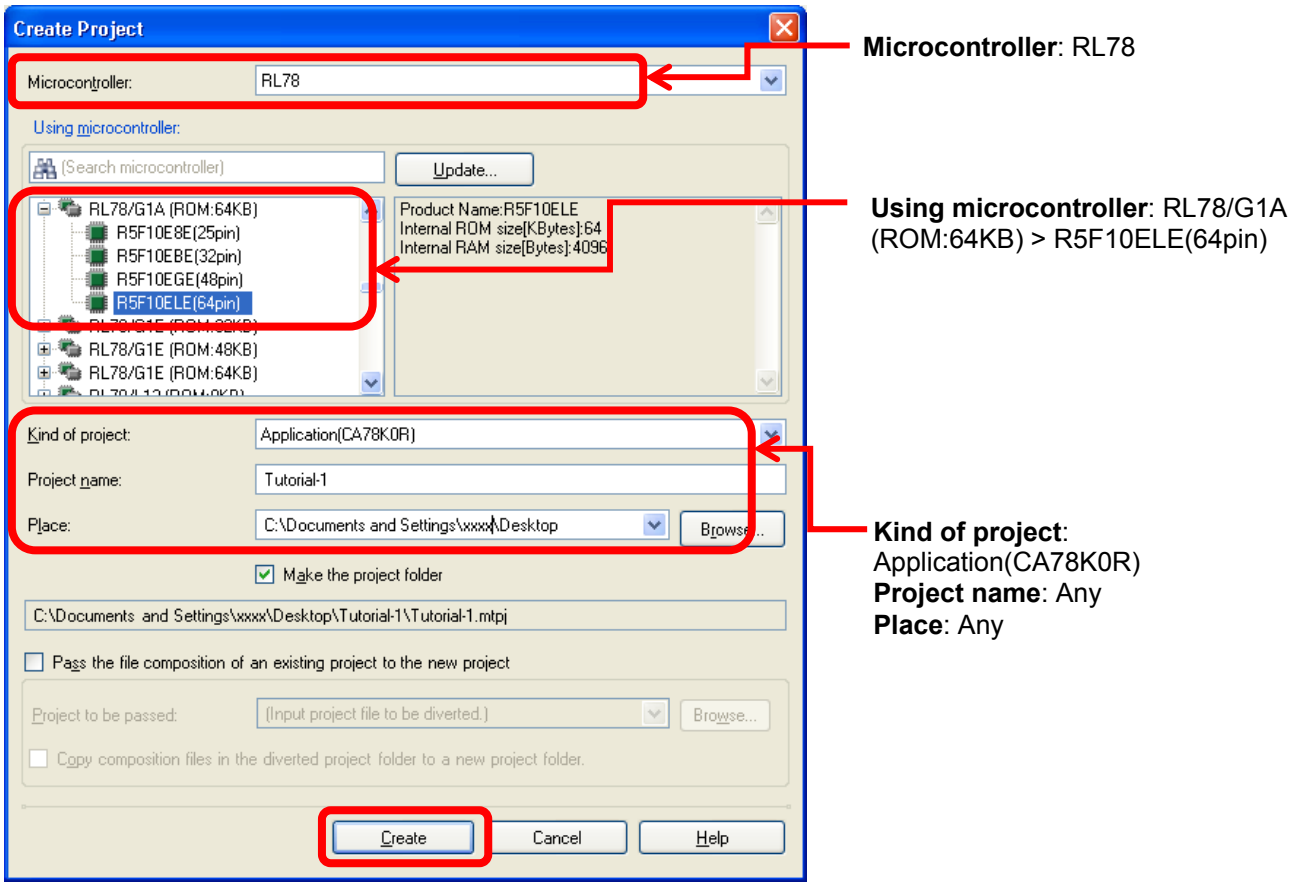
Select the plugins to be loaded when starting CubeSuite+. On the **Additional Function** tab, select "Code Generator Plug-in", "Code Generator Plug-in 2", "IronPython Console Plug-in", and "Program Analyzer Plug-in".



On the **Basic Function** tab, select "RL78,78K0R Build tool CA78K0R Plug-in", "RL78,78K0R Emulator Plug-in", and "RL78,78K0R Simulator Plug-in". Click **OK**.

On the **Start** tab, click **GO** under **Create New Project**, and then specify the settings shown below.

Select RL78/G1A to generate the code.



After you complete the above settings, click the **Create** button.

7.7 Configuring the code generation settings

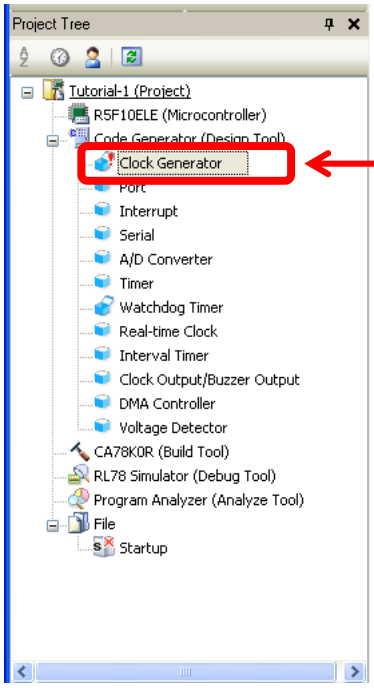
Create the program to actually run the microcontroller.

Here, you create the program by generating the code that configures the initial microcontroller settings, simply by specifying parameters on GUI screens.

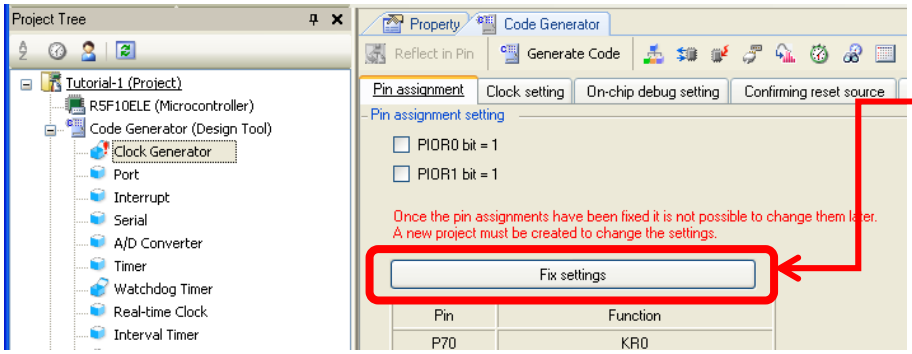
Basically, you can start setting the function you want to configure by double-clicking it in the project tree.

Configure each feature by following the procedures below.

7.7.1 Assigning clock generator pins

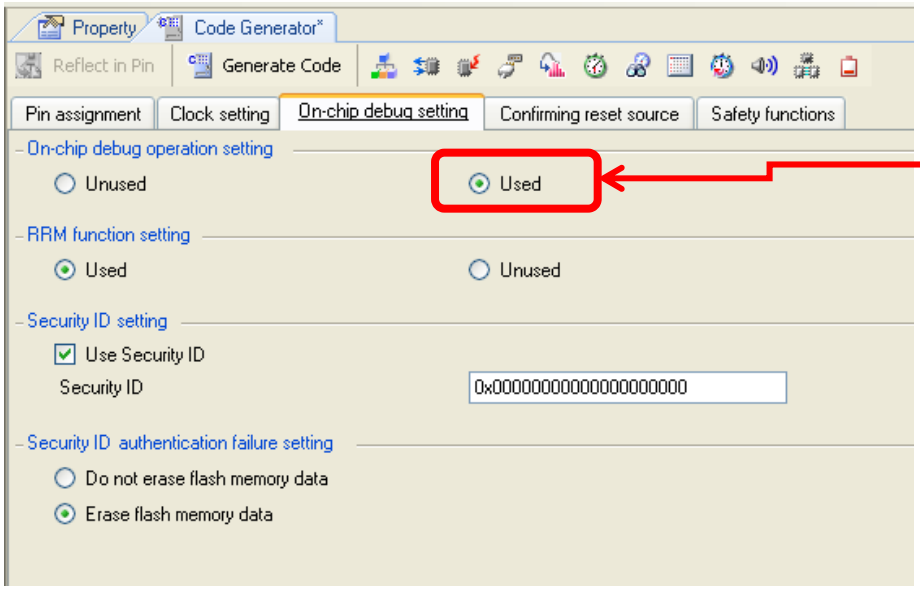


In the project tree, double-click **Clock Generator**.



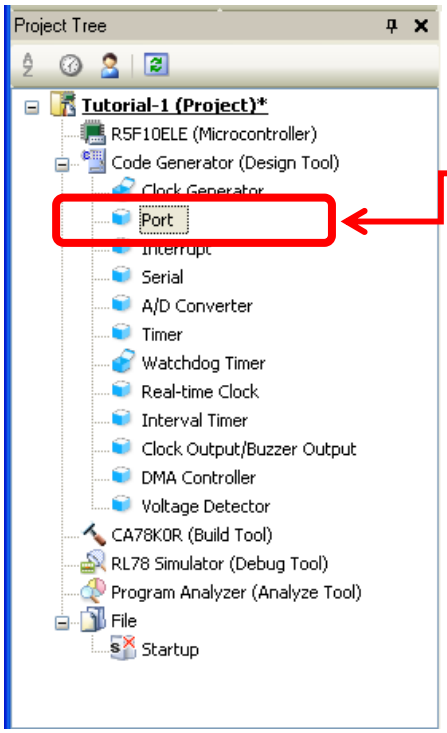
On the **Pin assignment** tab, click **Fix settings**.

On-chip debug setting for clock generator



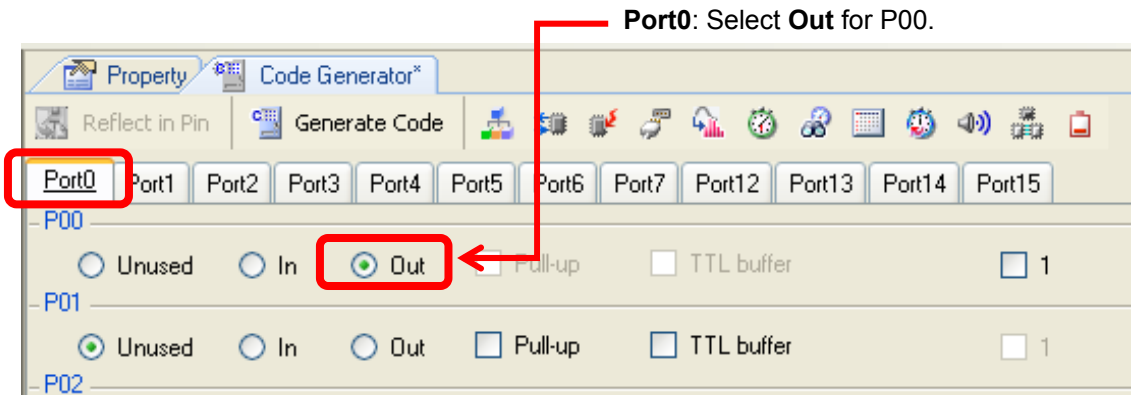
On-chip debug operation setting: Used

7.7.2 Assigning port pins



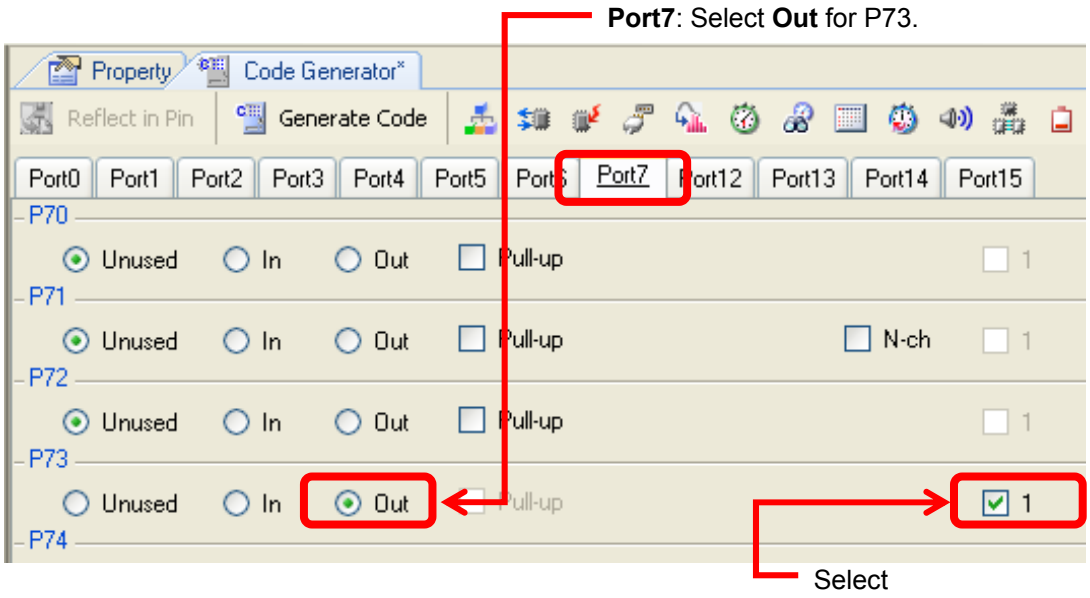
In the project tree, double-click **Port**.

Configure port 0. Port 0 is used to control the LED that blinks according to the behavior of the sensor.

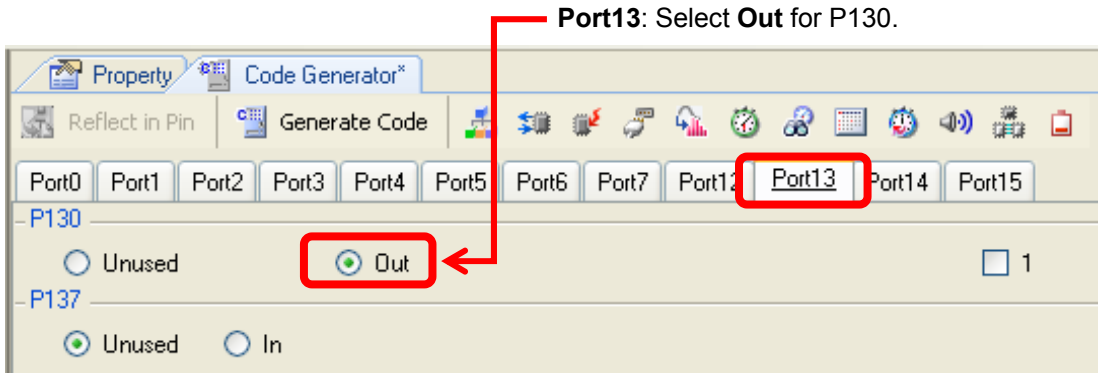


Configure port 7. Port 7 is used to control the CS pin on the Smart Analog IC.

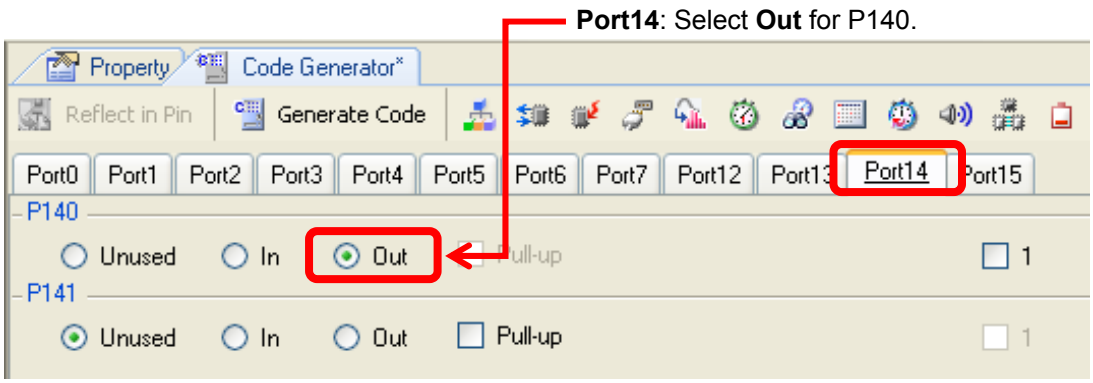
The active level of the output from this port is High, so select the check box "1" on the right end of the output port.



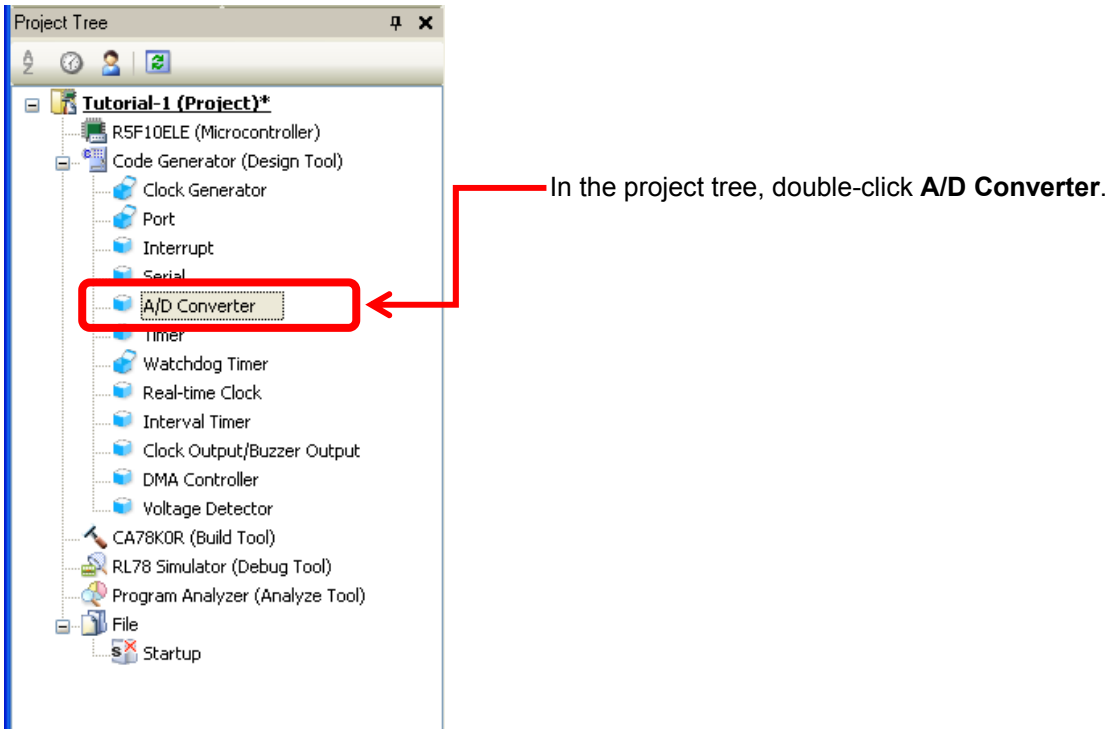
Configure port 13. Port 13 is used to control the RESET pin on the Smart Analog IC.

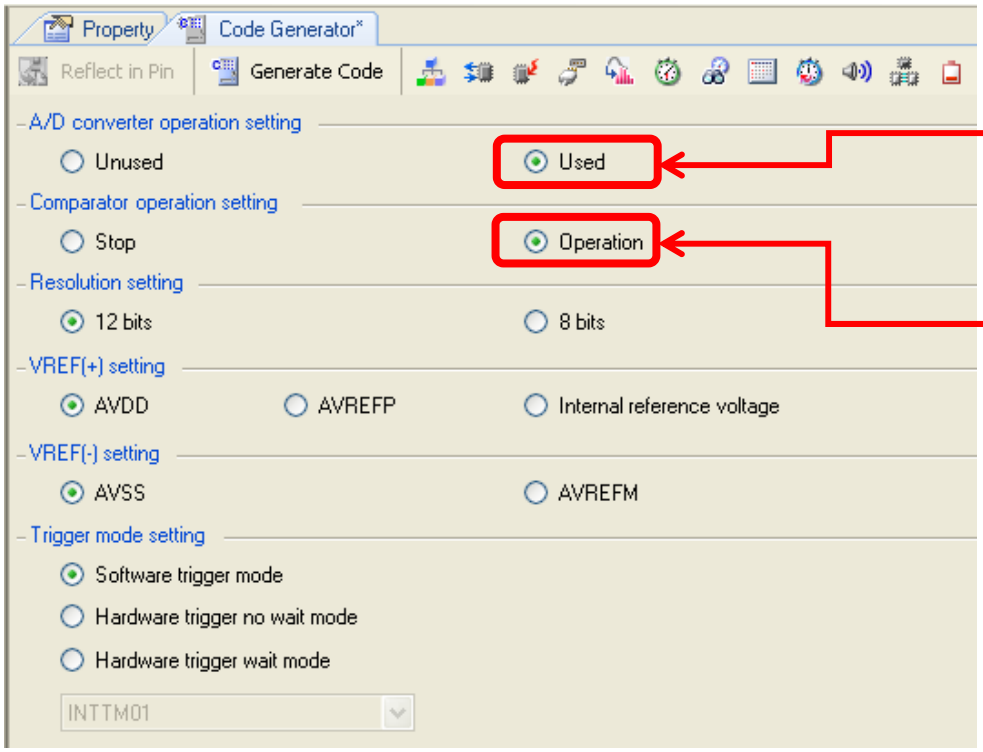


Configure port 14. Like port 0, port 14 is also used to control the LED that blinks according to the behavior of the sensor.



7.7.3 Configuring the A/D converter

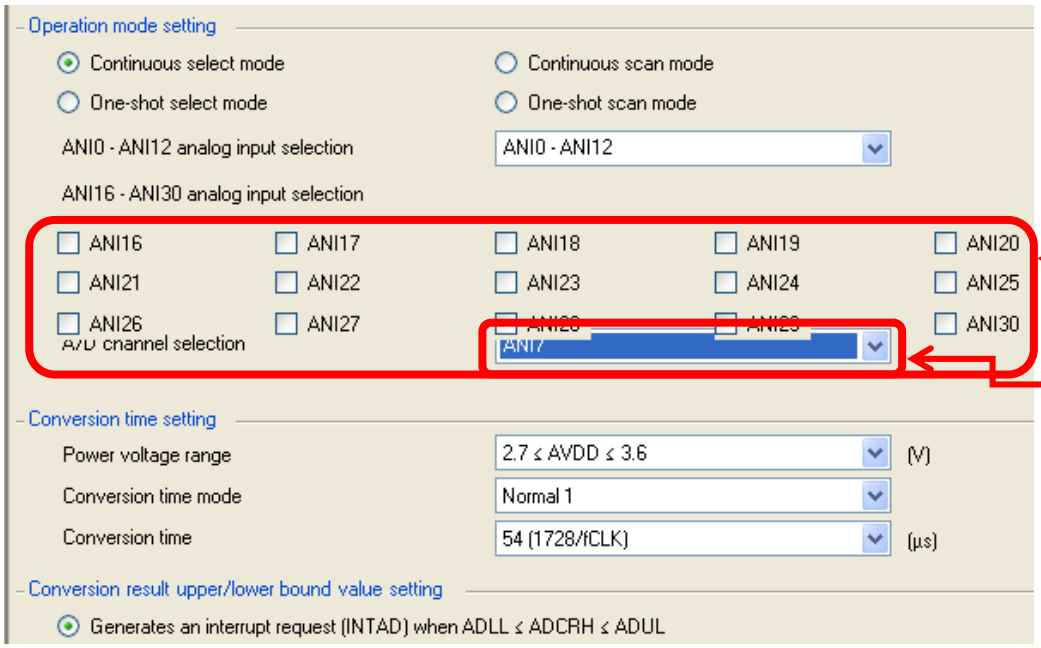




A/D converter operation setting: Used

Comparator operation setting: Operation

Scroll the screen to the bottom

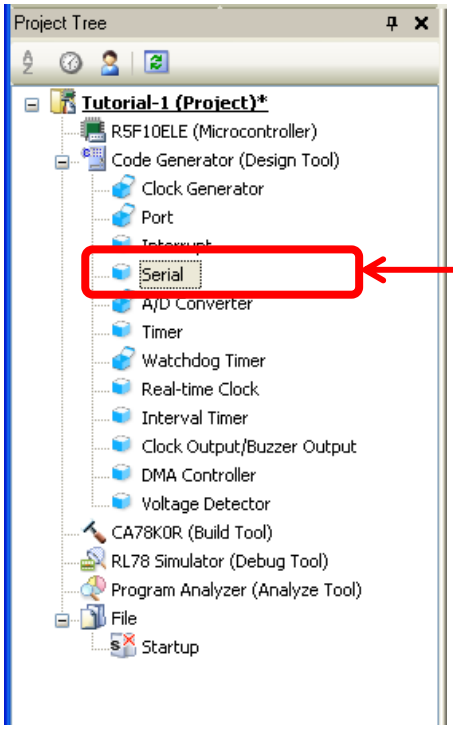


Operation mode setting:
Clear all the check boxes.

A/D channel selection:
Select ANI7.

This program only uses ANI7, so clear the check boxes for all the other A/D converter pins AIN16 to AIN30.

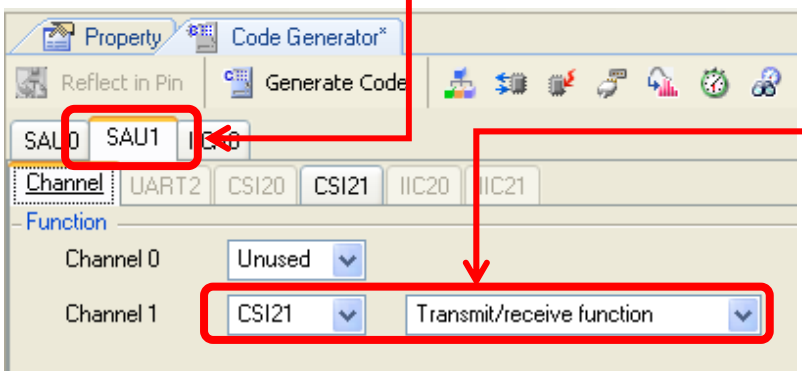
7.7.4 Configuring the serial interface



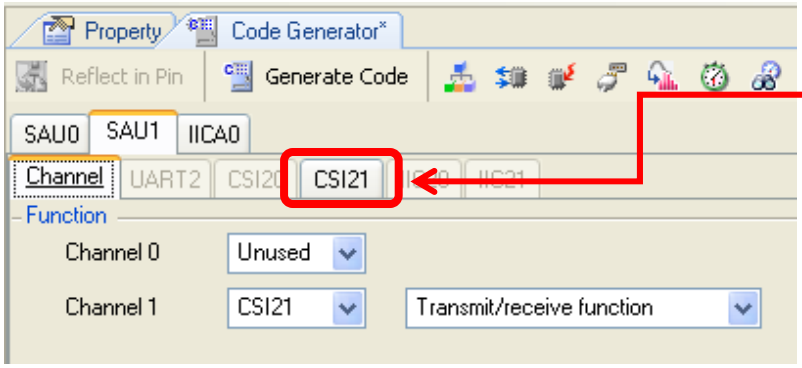
In the project tree, double-click **Serial**.

Serial array unit 1 uses channel CSI21, so select "CSI21" and specify its function.

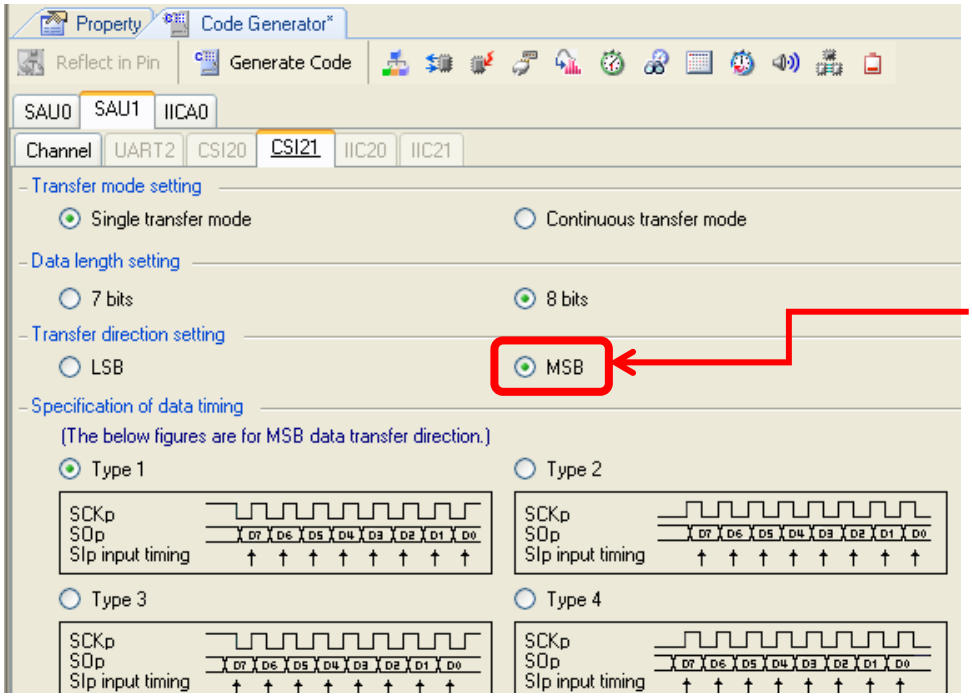
Click the **SAU1** tab.



Channel 1:
CSI21
Transmit/receive function

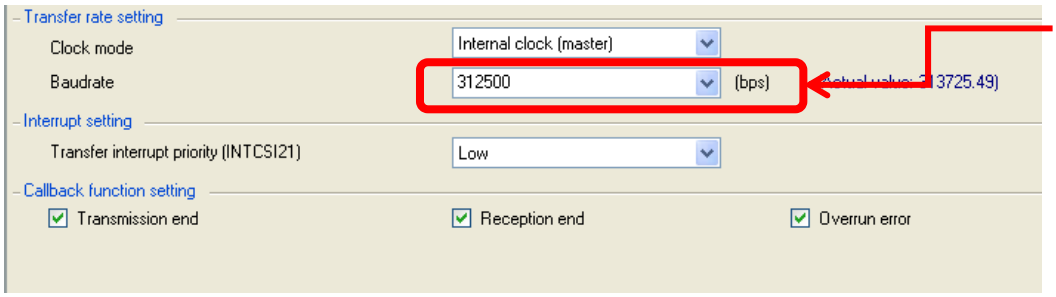


Click the **CSI21** tab.



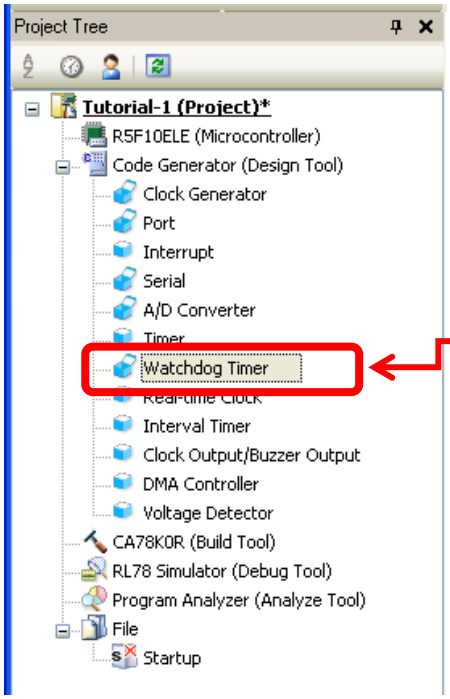
Transfer direction setting: MSB

Scroll the screen to the bottom

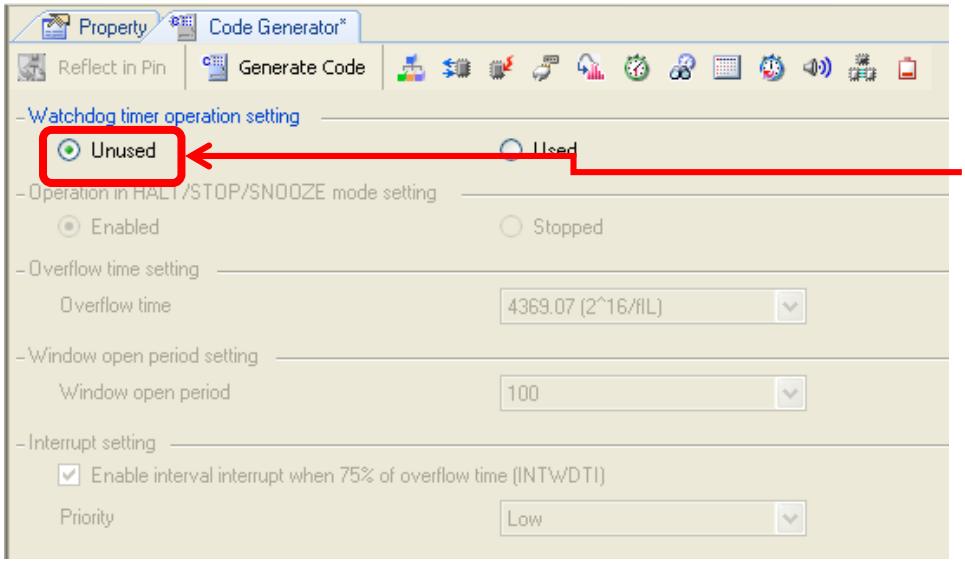


Transfer rate setting:
Baudrate: 312500

7.7.5 Configuring the watchdog timer



In the project tree, double-click **Watchdog Timer**.

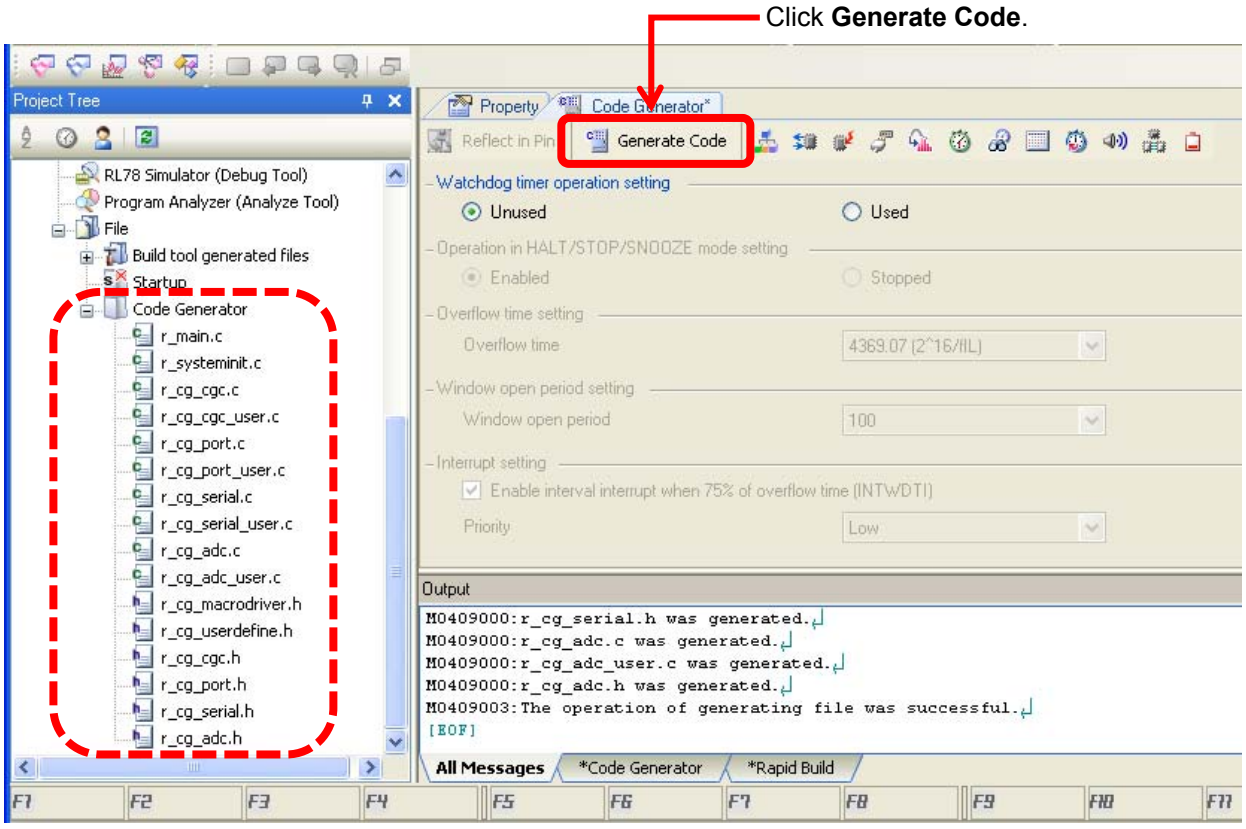


Watchdog timer operation setting: Unused

7.8 Generating code

Click **Generate Code** to generate code.

In the project tree, under the File folder, C source files and header files are generated.



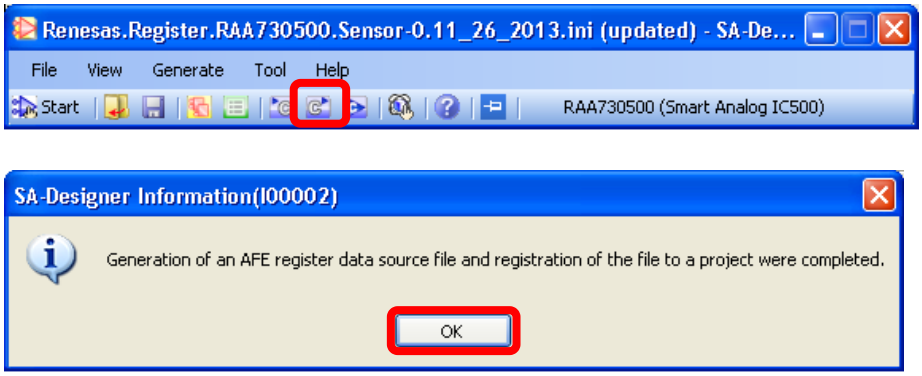
Files for code that initializes the microcontroller are generated.
APIs that control specific features are also generated.

7.8.1 Generating and registering C source files of circuit parameters

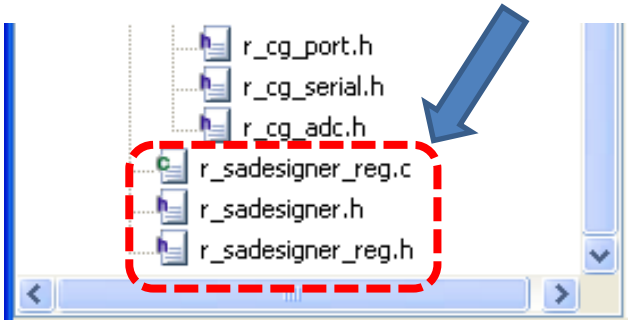
Return to the operation using SA-Designer.

Use SA-Designer to generate C source files of the Smart Analog IC parameters and register the generated C source files to Cube Suite+ automatically.

Click the icon enclosed by the red frame in the figure below.



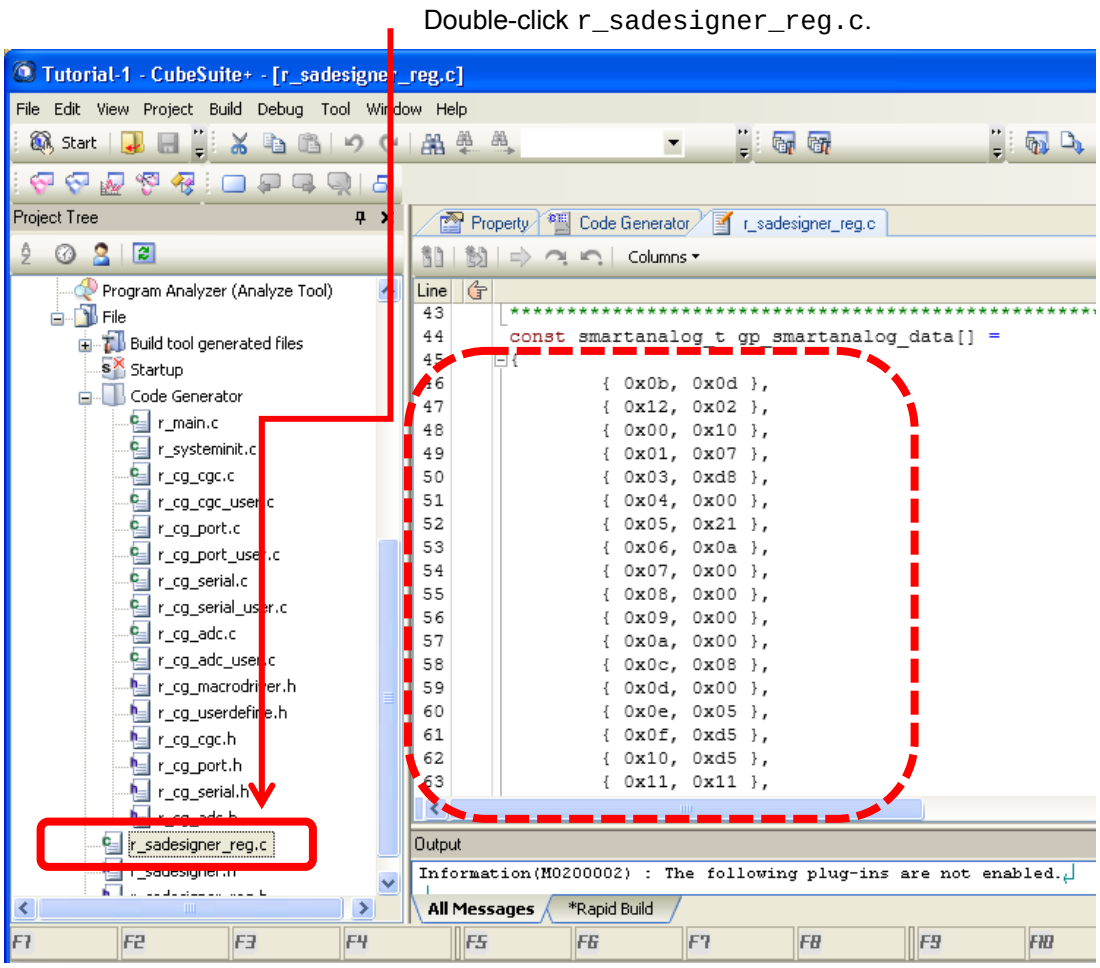
Click **OK**.



The C source files of circuit parameters are registered in the CubeSuite+ project tree.

7.9 Supplement for the files generated by SA-Designer

The Smart Analog IC parameters are defined as structure variables (`const`) in CubeSuite+.



The circuit parameters are generated as const variables defined in the form of {address, data}.

`{0xff, 0x00}` in the last line is a delimiter value.

7.10 Editing the program

7.10.1 r_main.c

Open `r_main.c` in the project tree.

Add the lines **in blue**.

```
/***********************************************************************************
***************************************
Includes
***********************************************************************************
***************************************/
#include "r_cg_macrodriver.h"
#include "r_cg_cgc.h"
#include "r_cg_port.h"
#include "r_cg_serial.h"
#include "r_cg_adc.h"
/* Start user code for include. Do not edit comment generated here */
#include "r_sadesigner.h"
/* End user code. Do not edit comment generated here */
#include "r_cg_userdefine.h"
/***********************************************************************************
***************************************
Global variables and functions
***********************************************************************************
***************************************/
/* Start user code for global. Do not edit comment generated here */
void R_SAIC_Write(smartanalog_t * const p_saic_data);
void R_SAIC_Create(void);
extern const smartanalog_t gp_smartanalog_data[];
/* End user code. Do not edit comment generated here */
void R_MAIN_UserInit(void);
/***********************************************************************************
***************************************
* Function Name:  main
* Description : This function implements main function.
* Arguments : None
* Return Value : None
***********************************************************************************
***************************************/
void main(void)
{
R_MAIN_UserInit();
/* Start user code. Do not edit comment generated here */
R_CSI21_Start();
R_SAIC_Create();
R_ADC_Start();
while (1U)
{
;
}
/* End user code. Do not edit comment generated here */
}
/***********************************************************************************
***************************************
* Function Name:  R_MAIN_UserInit
* Description : This function adds user code before implementing main function.
* Arguments : None
* Return Value : None
***********************************************************************************
***************************************/
void R_MAIN_UserInit(void)
{
/* Start user code. Do not edit comment generated here */
EI();
/* End user code. Do not edit comment generated here */
```

Include the source files generated by SA-Designer.

Declare the SPI function and global variables.

Configure initial setting of SPI and Smart Analog IC and start A/D converter.

```
}
/* Start user code for adding. Do not edit comment generated here */
void R_SAIC_Create(void)
{
volatile uint16_t w_count;
/* Analog IC Reset */
P13.0 = 0U;
/* Change the waiting time according to the system */
for (w_count = 0U; w_count < 0x1A; w_count++)
{
NOP();
}
/* Analog IC Reset release */
P13.0 = 1U;
R_SAIC_Write(gp_smartanalog_data);
}
void R_SAIC_Write(smartanalog_t * const p_saic_data)
{
uint8_t adrs;
uint8_t dat;
uint8_t wait;
smartanalog_t *p_saic_write;
PMC3 = 0xfc;
CSIMK21 = 1U; /* mask INTCSI21 */
p_saic_write = p_saic_data;
while (p_saic_write->address != 0xff)
{
P7.3 = 0U; /* SAIC CS=L */
/* SA stable waiting time (tSKA) */
for (wait = 0U; wait < 1U; wait++)
{
NOP();
}
adrs = (p_saic_write->address & 0x7f) | 0x80; /* 0x80 is data write mode */
SIO21 = adrs; /* send SAIC Address data */
while (CSIIF21 == 0U); /* wait for CSI send */
CSIIF21 = 0U;
dat = p_saic_write->data;
SIO21 = dat; /* send SAIC Regiseter Data */
while (CSIIF21 == 0U); /* wait for CSI send */
CSIIF21 = 0;
/* SA Stable waiting time (tKSA) */
for (wait = 0U; wait < 1U; wait++)
{
NOP();
}
P7.3 = 1U; /* SAIC CS=H */
/* SA Stable waiting time (tSHA) */
for (wait = 0U; wait < 1U; wait++)
{
NOP();
}
p_saic_write++;
}
}
/* End user code. Do not edit comment generated here */
```

Reset the Smart Analog IC.

Cancel the Smart Analog IC reset period and call configuration function.

Detect a delimiter.

Select a chip.

Address transmission

Data transmission

Cancel chip selection.

When you complete coding up to here, save the changes.

7.10.2 r_cg_adc_user.c

Open `r_cg_adc_user.c` in the project tree.

Add the lines **in blue**.

```
/***********************************************************************************
***************************************
Global variables and functions
***********************************************************************************
***************************************/
/* Start user code for global. Do not edit comment generated here */
uint16_t g_tmp = 0;
/* End user code. Do not edit comment generated here */
/***********************************************************************************
***************************************
* Function Name:  r_adc_interrupt
* Description : This function is INTAD interrupt service routine.
* Arguments : None
* Return Value : None
***********************************************************************************
***************************************/
__interrupt static void r_adc_interrupt(void)
{
/* Start user code. Do not edit comment generated here */
uint16_t adcr_2bit = 0;
R_ADC_Get_Result(&g_tmp);
/* Example of updating LED by dividing A/D-converted value by 4 */
/* Use higher 2 bits out of A/D-converted lower 12 bits of g_tmp */
adcr_2bit = g_tmp >> 10;
/* Change the LED control port pin output based on 2-bit value. 1: LED = OFF, 0:
LED = ON */
switch (adcr_2bit)
{
case 0U:  P0.0 = 1;
P14.0 = 1;
break;
case 1U:  P0.0 = 0;
P14.0 = 1;
break;
case 2U:  P0.0 = 1;
P14.0 = 0;
break;
case 3U:  P0.0 = 0;
P14.0 = 0;
break;
default:  /* Through */
break;
}
/* End user code. Do not edit comment generated here */
}
/* Start user code for adding. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */
```

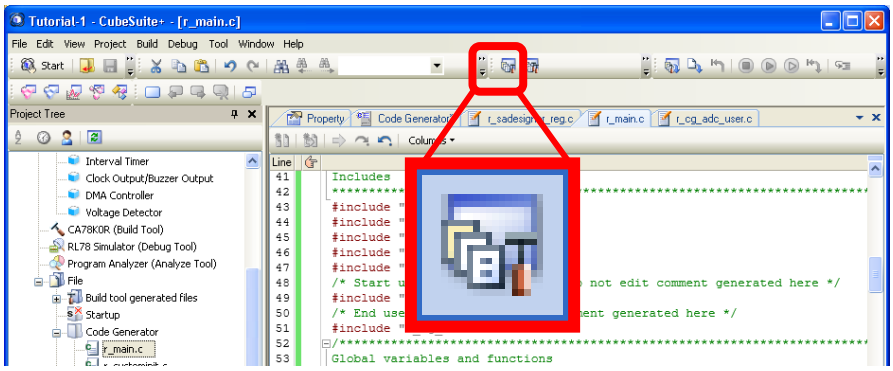
Define the variable for acquiring A/D conversion results.

Divide the A/D conversion result by 4 to use each for LED control.

When you complete coding up to here, save the changes.

7.11 Building the program

When you complete editing the source files, generate a program (load module) by using the **Builds the project** button. Click the button enclosed by the red frame in the figure below.

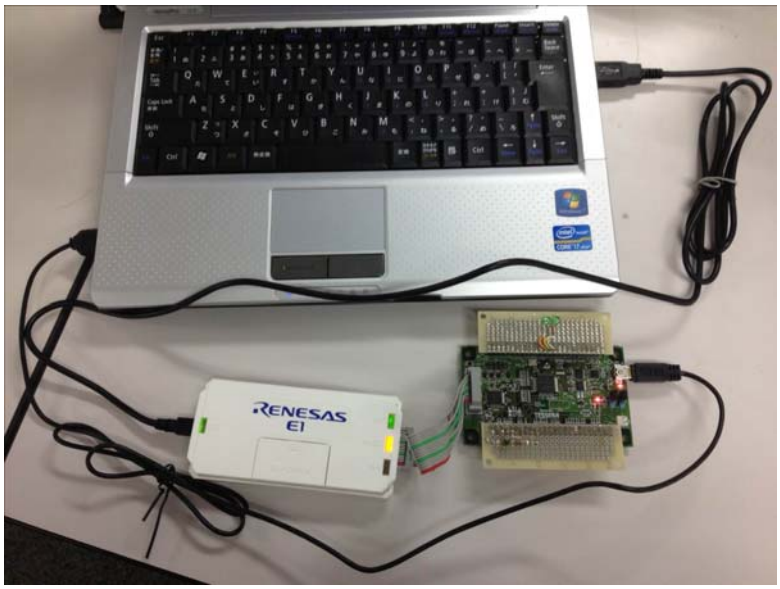


If no error is output here, the program has been generated successfully.

If an error is output, check the error message and review the settings configured in section 7.7 and 7.10.

7.12 Creating a program execution environment (connecting the emulator)

Connect the evaluation board, E1 emulator, and computer.



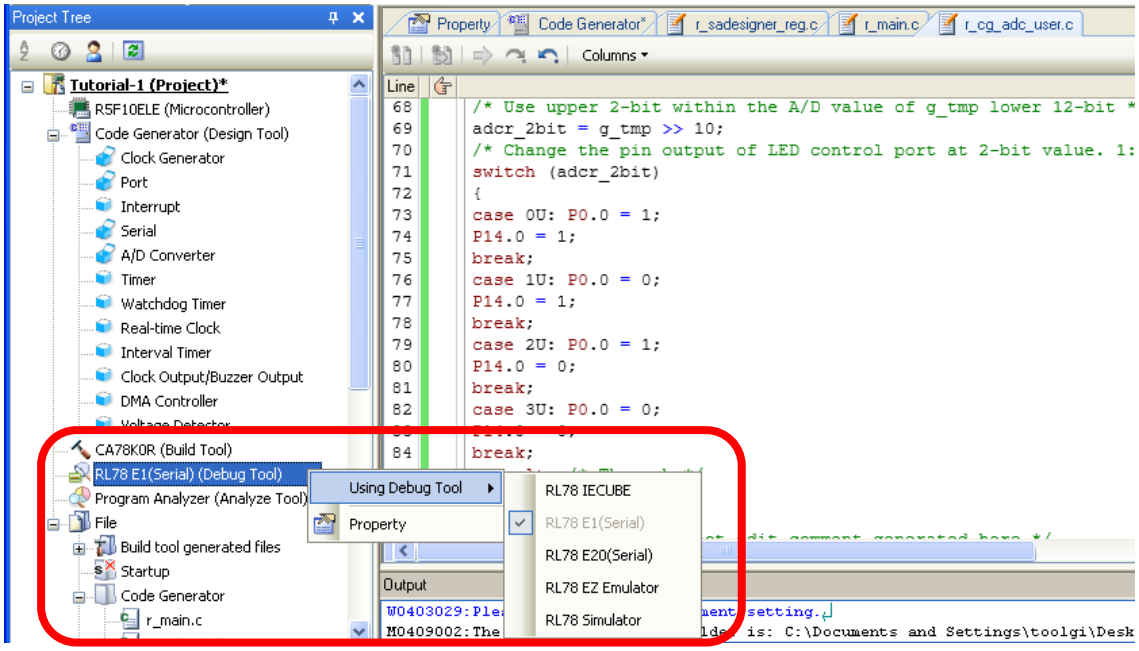
Prepare two USB cables.

- Use one USB cable to connect the E1 emulator to the computer.
- Use the other USB cable to connect the evaluation board to the computer.

Connect the E1 emulator and evaluation board TSA-IC 500 by using the cable that comes with the E1 emulator.

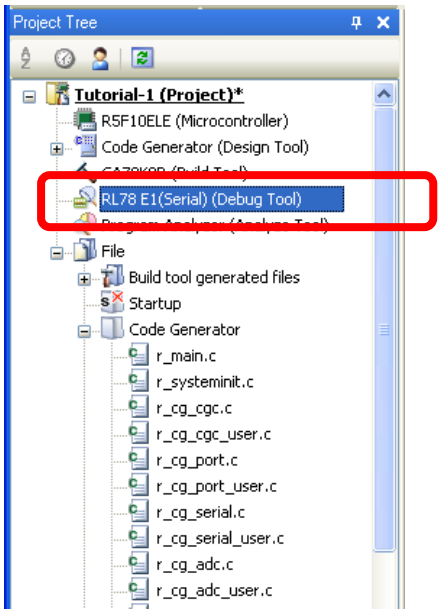
7.12.1 Configuring the debug tool

Configure the debug tool.



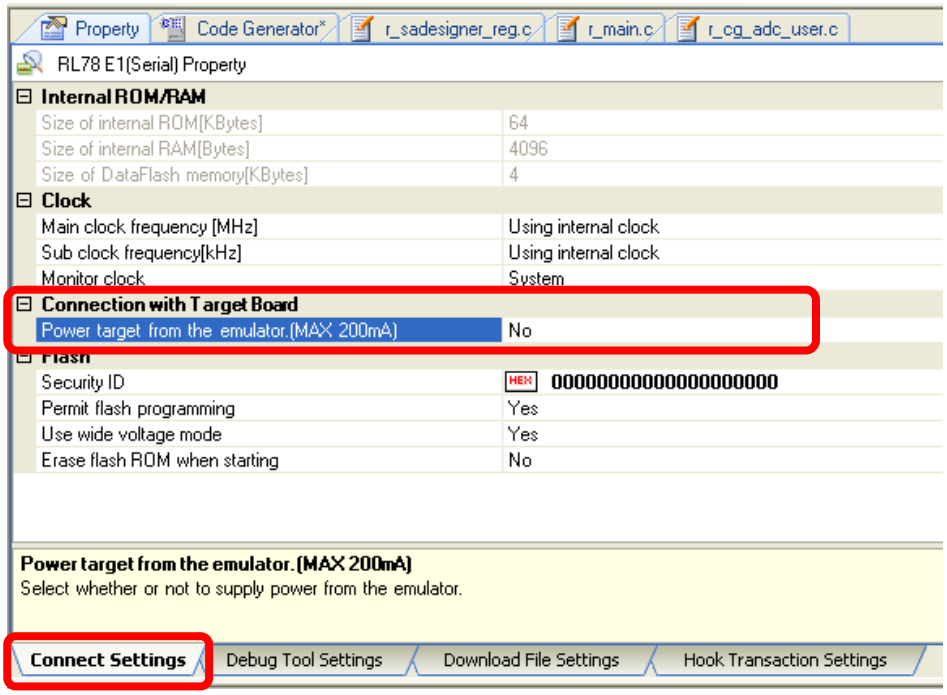
In the project tree, right-click **RL78 E1(Serial) (Debug Tool)**, point to **Using Debug Tool**, and then click **RL78 E1(Serial)**.

Configure the power settings for the E1 emulator connection and the setting to enable memory space access during program execution.



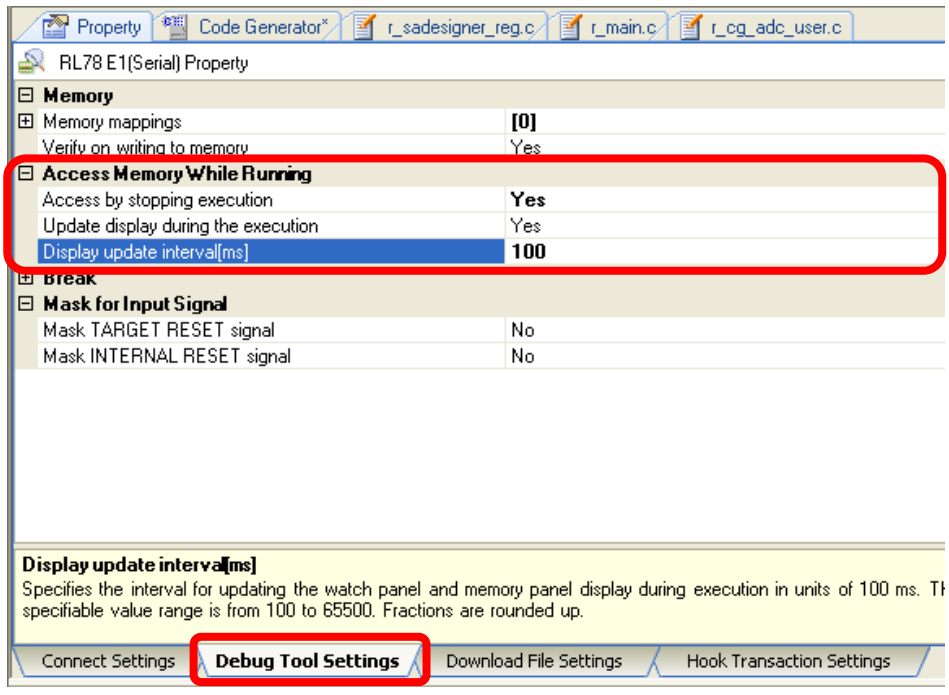
In the project tree, double-click **RL78 E1(Serial) (Debug Tool)**.

Configure the settings as shown below.



Connect Settings tab > **Connection with Target Board** > **Power target from the emulator. (MAX 200mA)**:

No



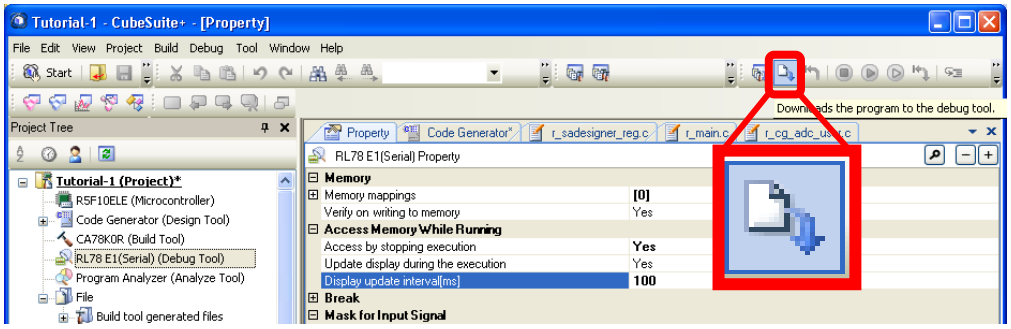
Debug Tool Settings tab > **Access Memory While Running**

Access by stopping execution: Yes

Display update interval[ms]: 100

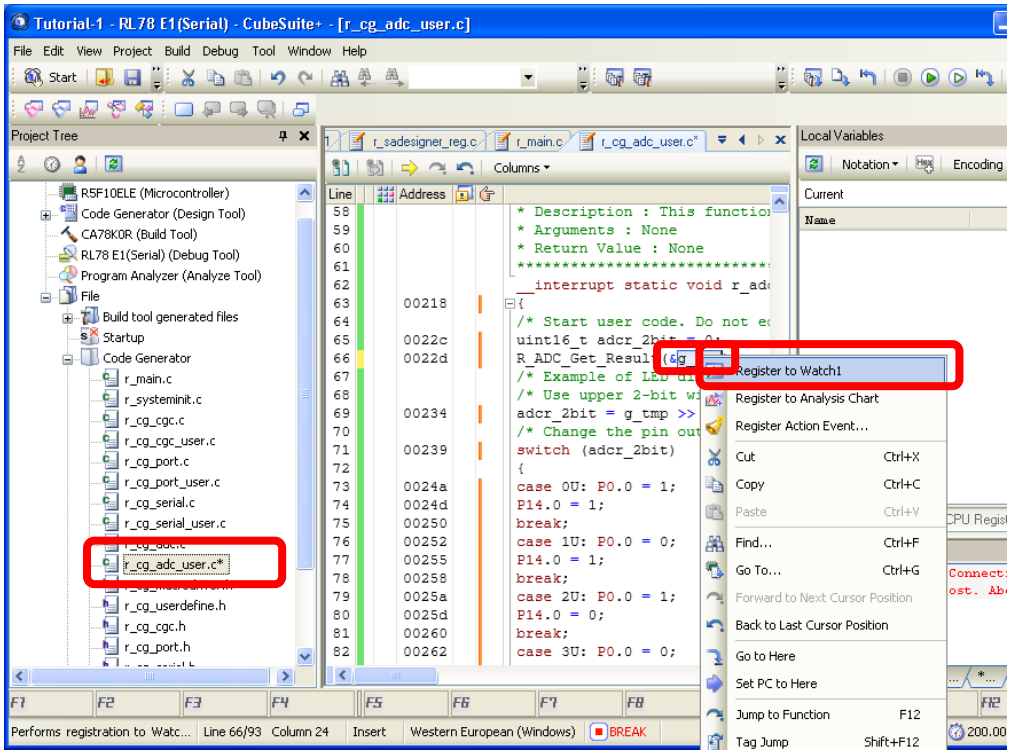
7.13 Downloading the program to the microcontroller

Download the created program (load module) to the microcontroller. Click the button enclosed by the red frame in the figure below.



7.14 Monitoring the A/D-converted values

Values A/D-converted in the program are stored in the variable `g_tmp`. To monitor the operation, add `g_tmp` to a watch list.



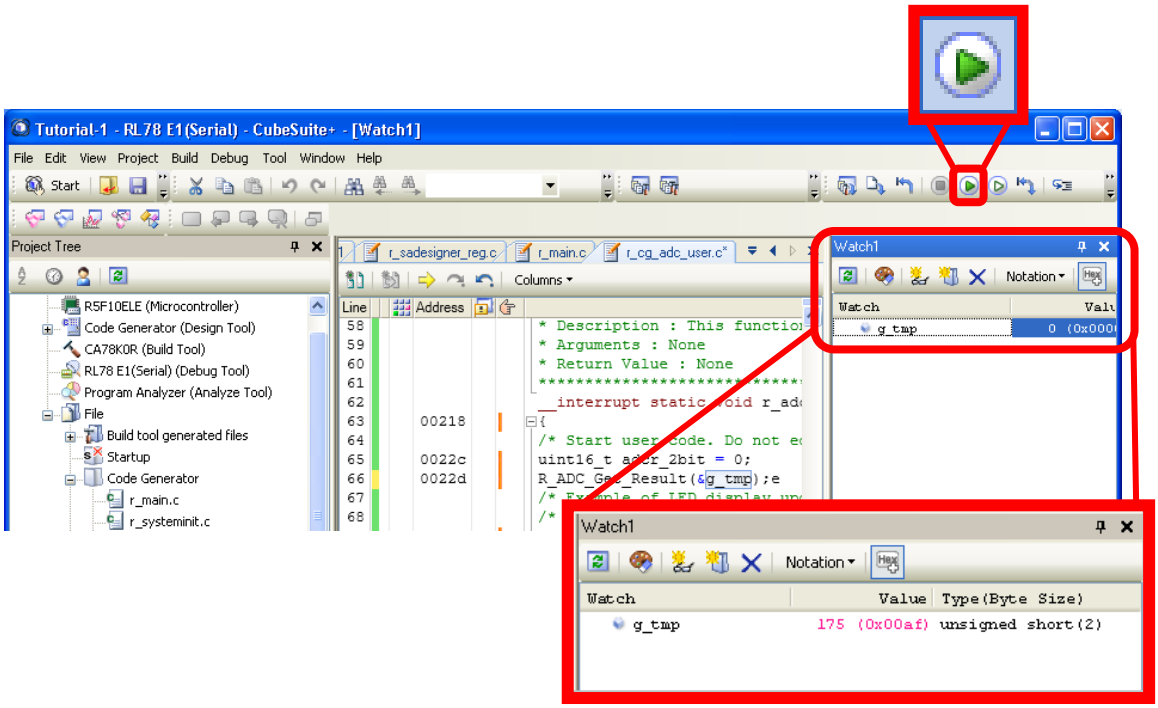
In the project tree, double-click "**r_cg_adc_user.c**". In the right pane, double-click `g_tmp` in code, right-click the same `g_tmp`, and then select **Register to Watch1**.

If **Register to Watch1** does not appear, on the menu bar, click **Debug** and then **Download** to add the watch registration function.

7.15 Running the program to check the sensor operation

Run the program downloaded to the microcontroller to check the sensor operation.

Click the button enclosed by the red frame in the figure below.



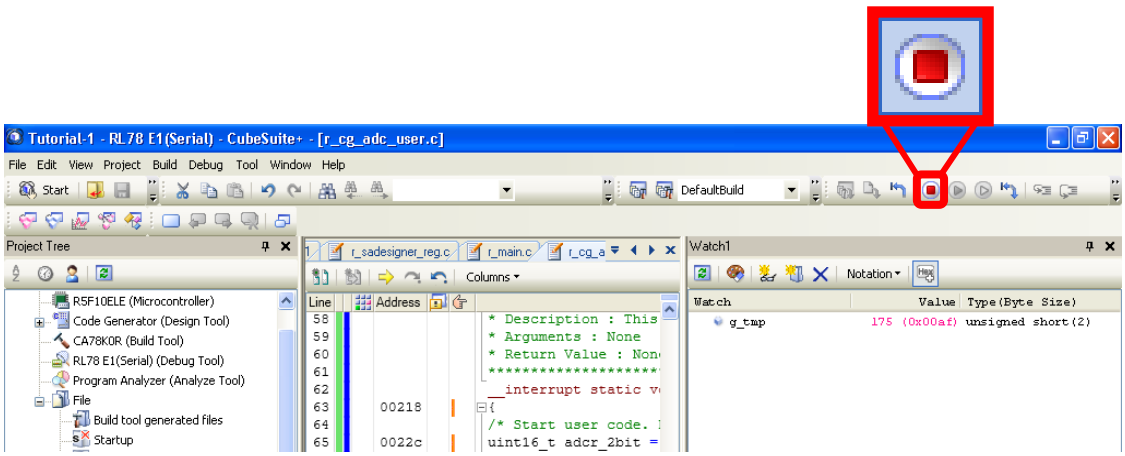
The value of g_tmp that has been added is displayed.

Take an action such as holding your hand above the sensor or move the sensor to a light source such as a fluorescent light, and confirm that the g_tmp value changes according to the sensor operation.

Also check that the LEDs light.

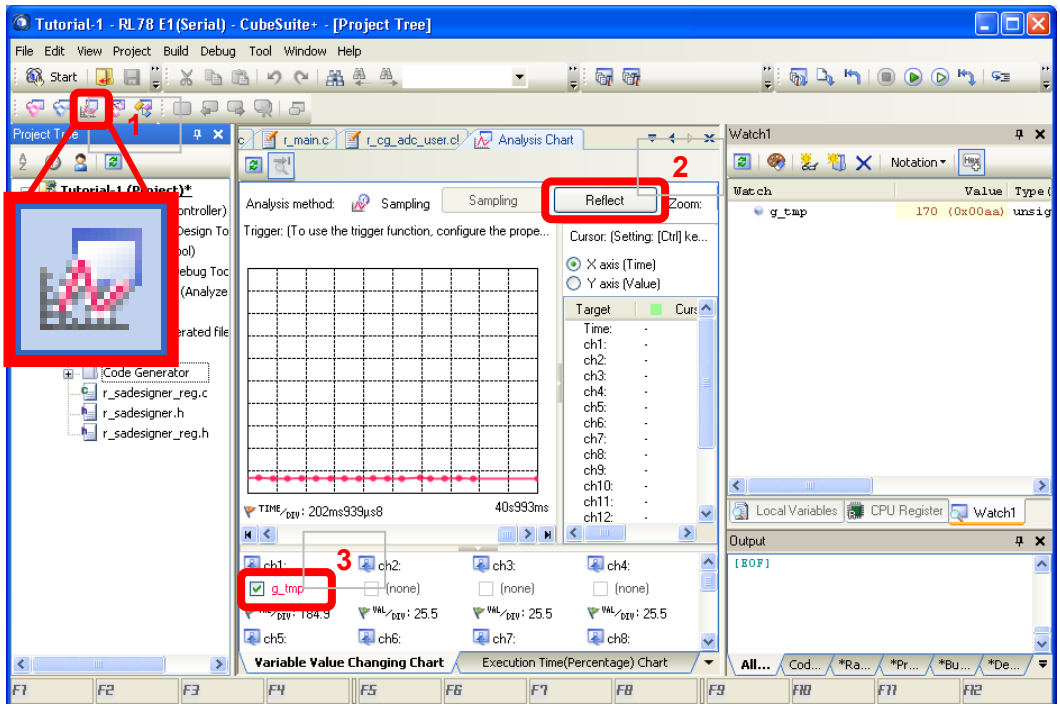
Here, display the sensor operation in graph form to make it easy to grasp.

Stop the program. Click the button enclosed by the red frame in the figure below.



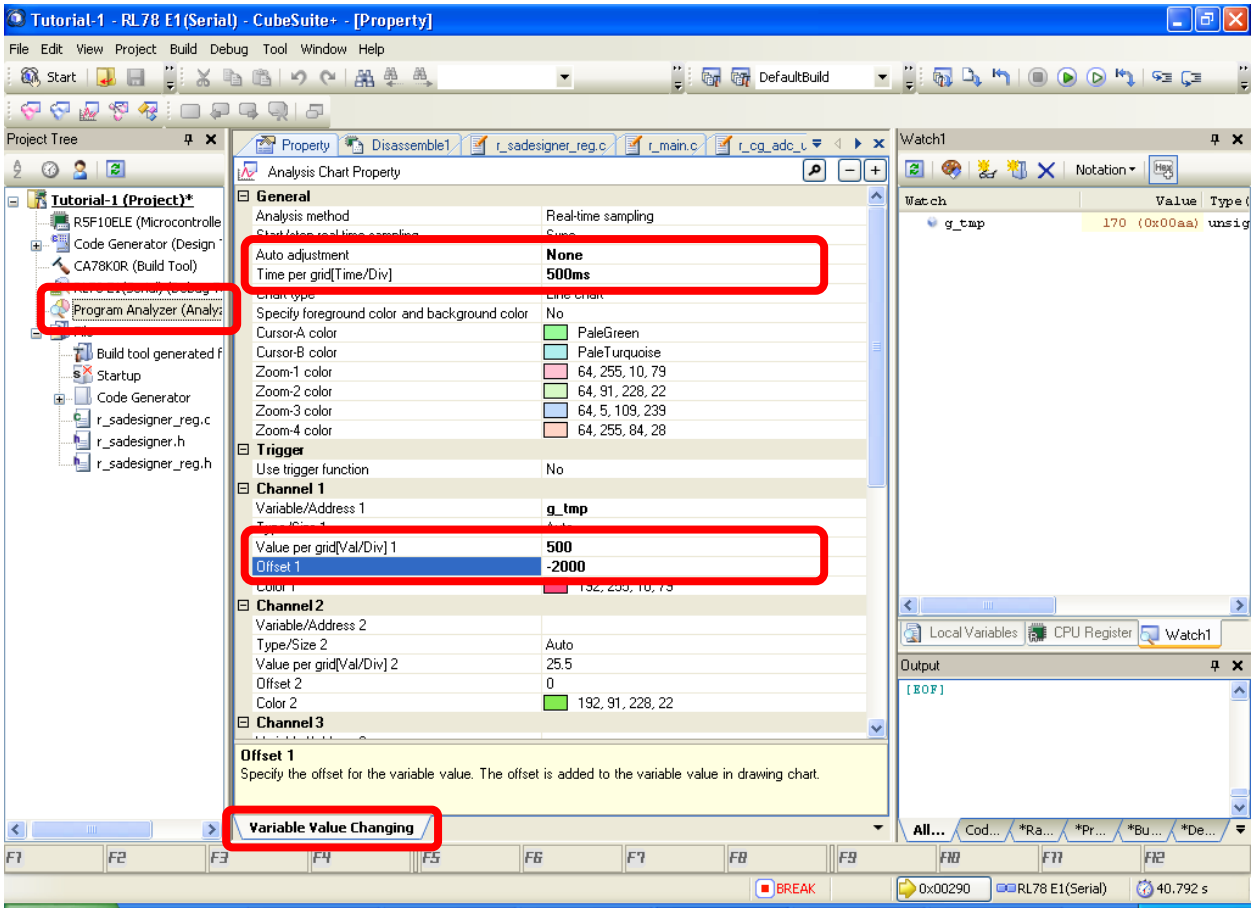
7.16 Displaying the sensor operation in graph form on the Analysis Chart panel

To display the sensor operation in graph form, perform the following steps 1 to 3.



1. Click the **Displays Analysis Chart** panel icon on the top left of the screen.
2. A blank graph appears. Click the **Reflect** button above the graph.
3. Confirm that `g_tmp` has been added below the graph.

In the project tree, double-click **Program Analyzer (Analyze Tool)**.



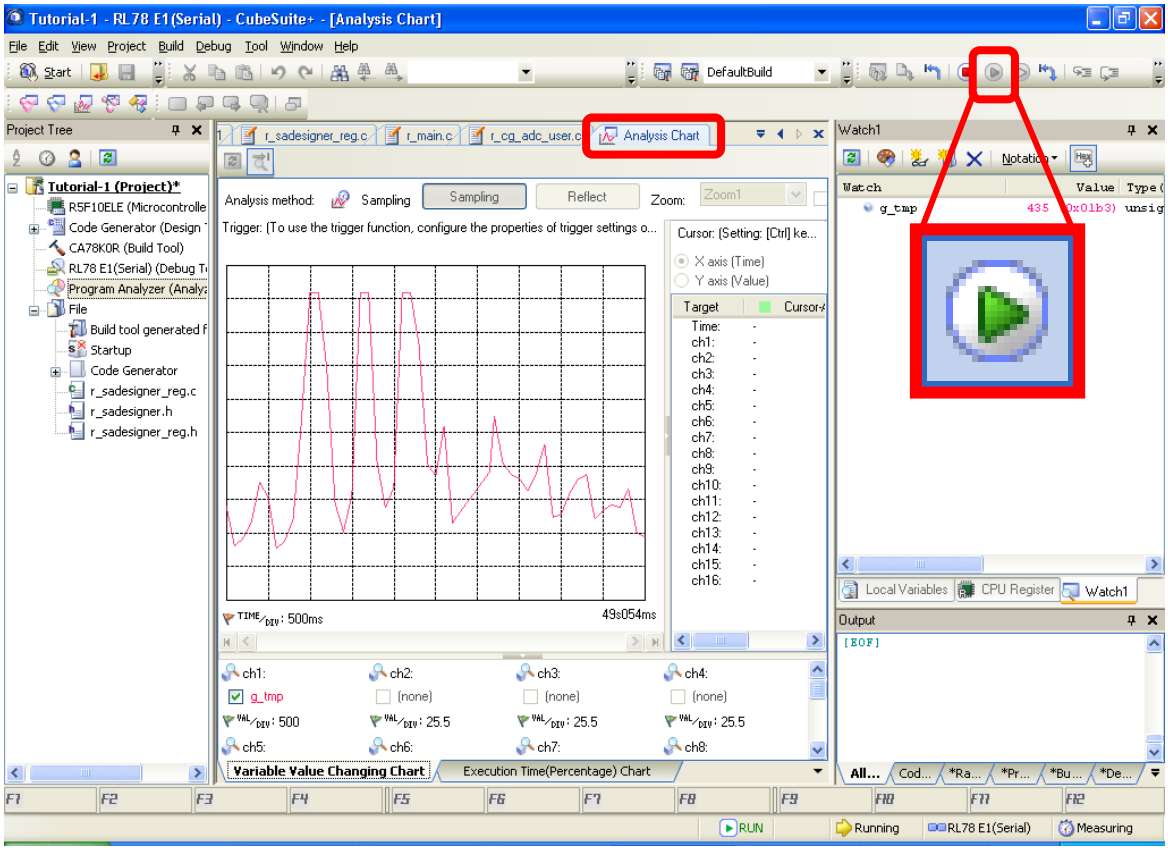
Variable Value Changing tab > **General** > **Auto adjustment**: None

Time per grid[Time/Div]: 500ms

Channel 1 > **Value per grid[Val/Div] 1**: 500

Offset 1: -2000

When you have finished the above settings, click the icon enclosed by the red frame in the figure below to run the program again.

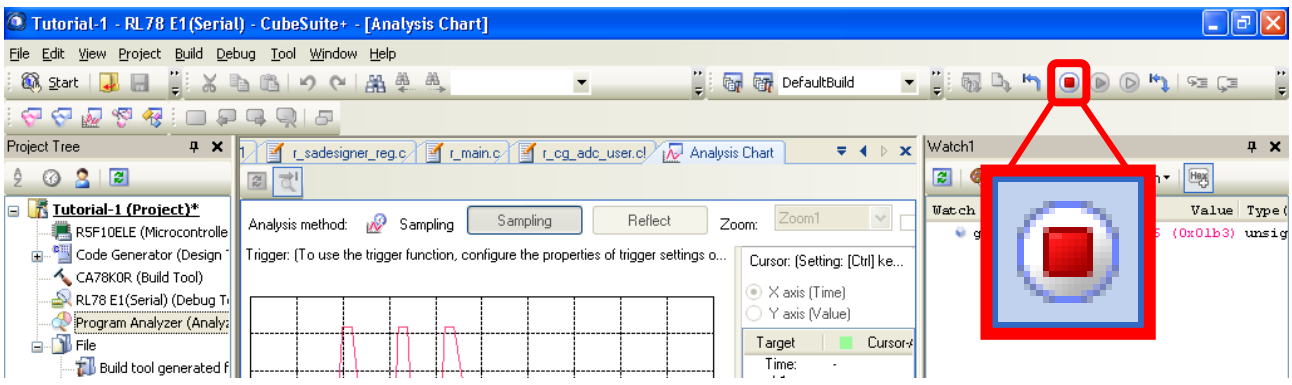


On the **Analysis Chart** tab, take an action such as holding your hand above the sensor or moving the sensor to a light source such as a fluorescent light, and confirm that the sensor operation is displayed in graph form.

The program is now created.

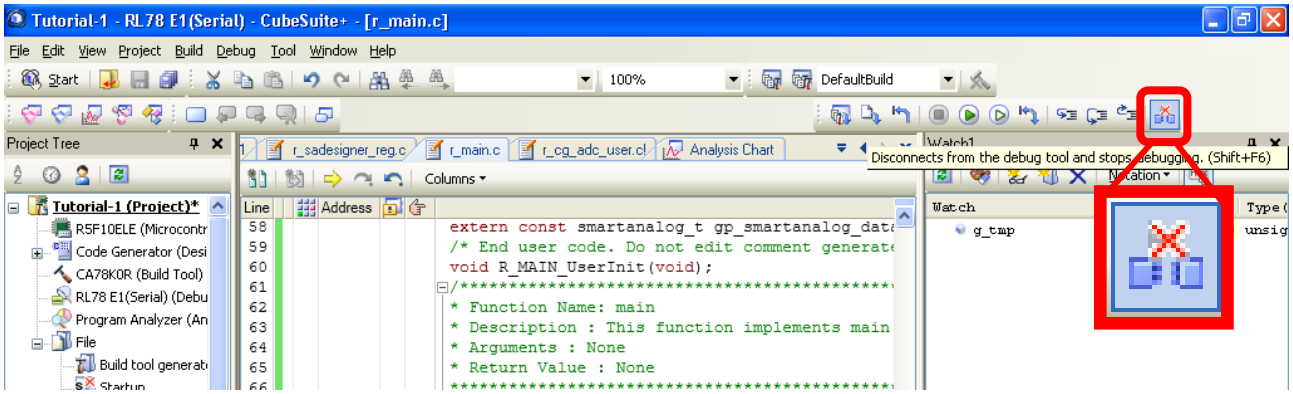
Stop the program.

Click the button enclosed by the red frame in the figure below.

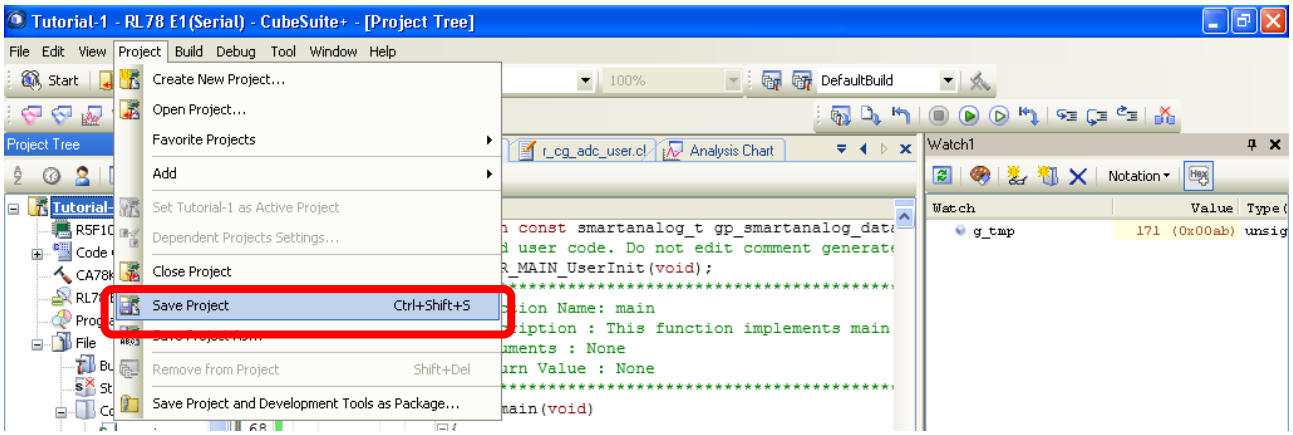


Terminate the debug tool to finish debugging.

Click the icon enclosed by the red frame in the figure below.

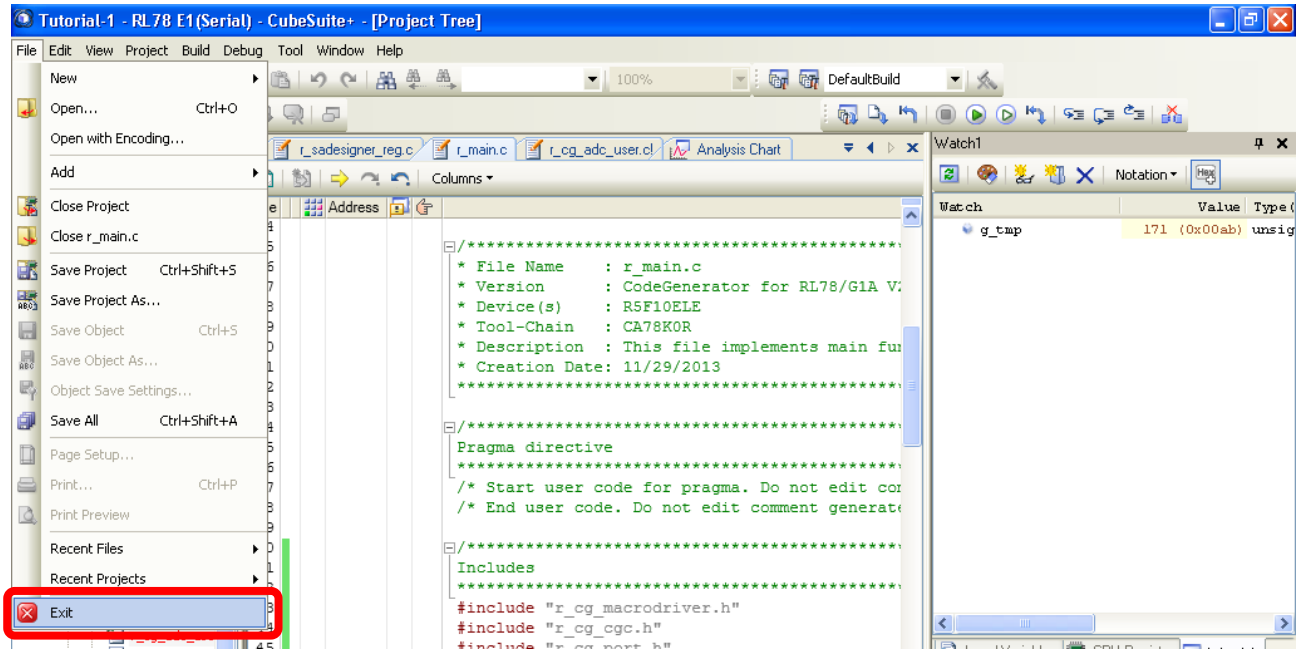


Save the project and terminate CubeSuite+.

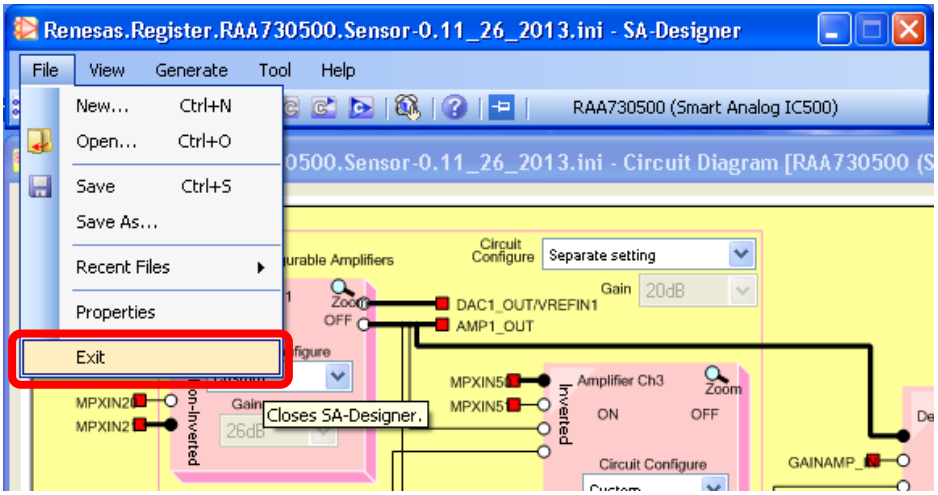


Remark: When CubeSuite+ is started the next time and this project is loaded, the above settings are automatically restored.

On the **File** menu, click **Exit** to terminate CubeSuite+.



Terminate SA-Designer.



On the **File** menu, click **Exit** to terminate SA-Designer.

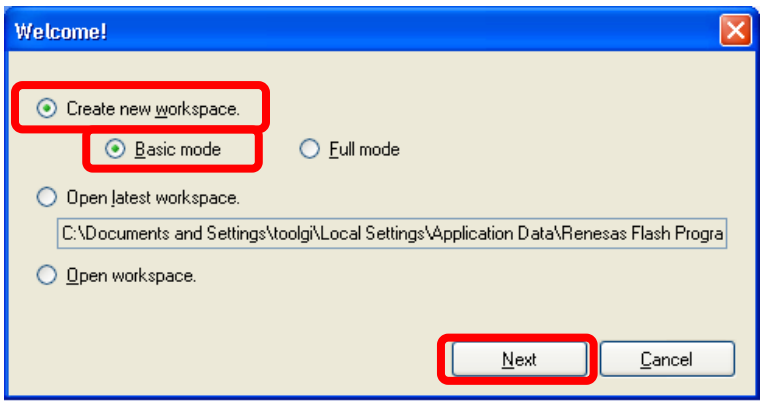
You are asked whether to save the AFE register file used here. Save the file, as necessary.

Keep the evaluation board, the E1 emulator, and the computer connected because they will be used to write the program to the microcontroller in the next step.

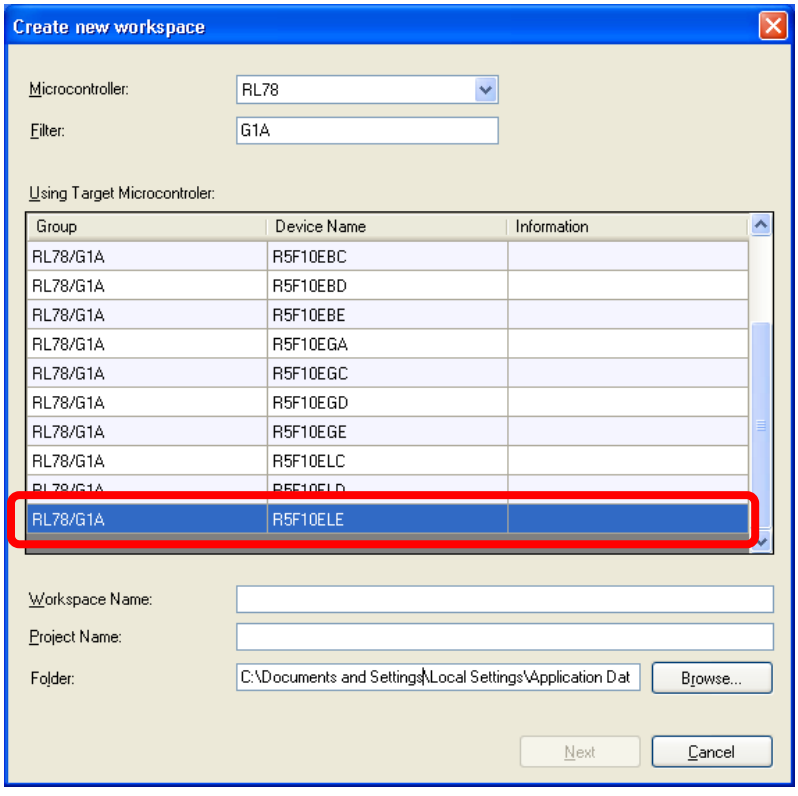
7.17 Writing the program to the microcontroller

Write the program to the RL78/G1A. Start the programming tool from the Start menu.

All Programs > **Renesas Electronics Utilities** > **Programming Tools** > **Renesas Flash Programmer V*x.xx***



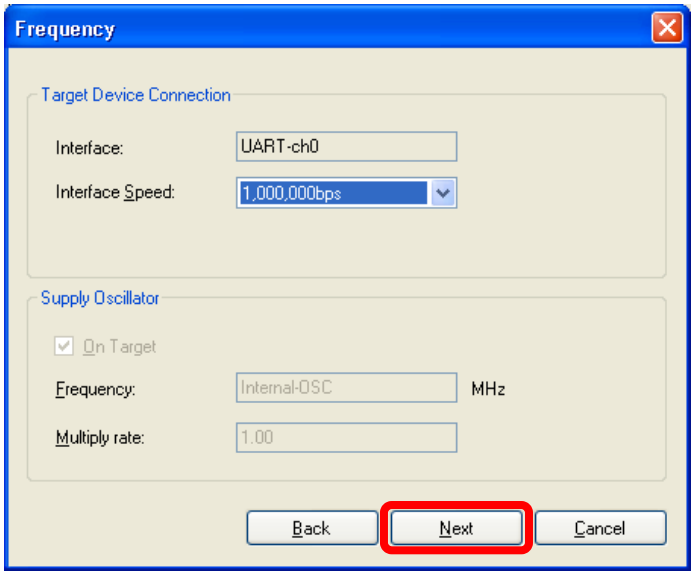
In the **Welcome** dialog box, select **Create new workspace**, **Basic mode**, and then click **Next**.



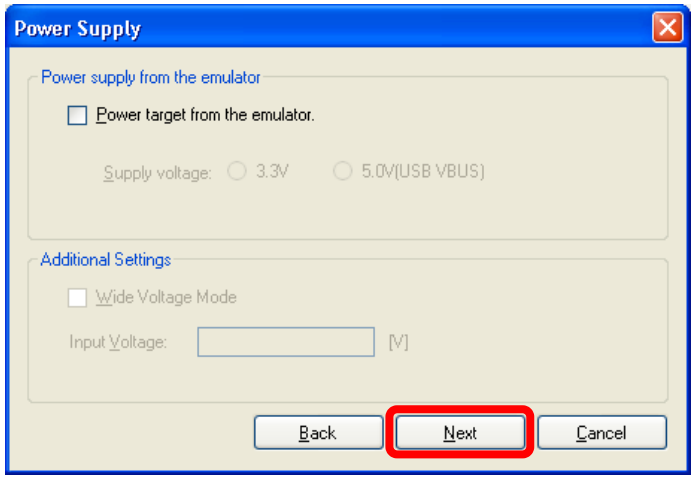
On the **Create new workspace** screen, select RL78/G1A - R5F10ELE under **Group** and **Device Name**, type any workspace name, and then click **Next**.

(The workspace name is automatically inserted to the **Project Name** box.)

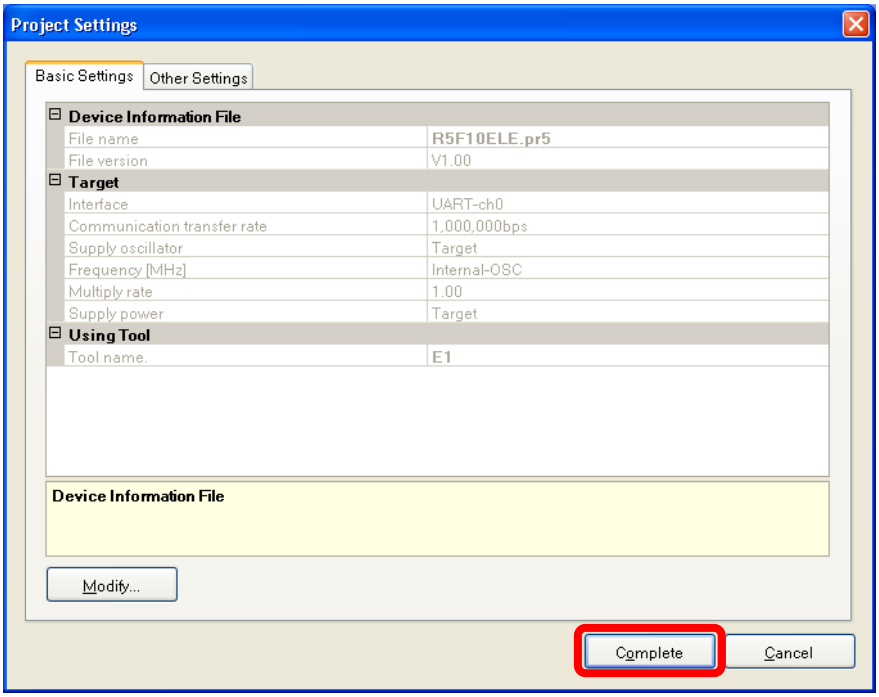
Confirm that **Tool** is set to "E1", and then click **Next**.



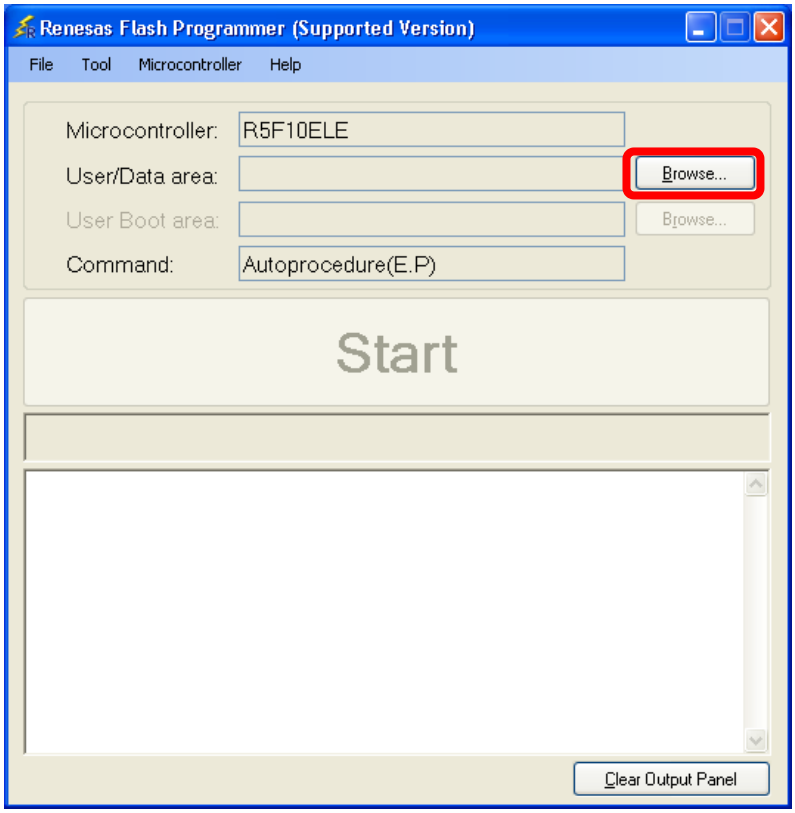
Use the interface speed selected by default as is. Click **Next**.



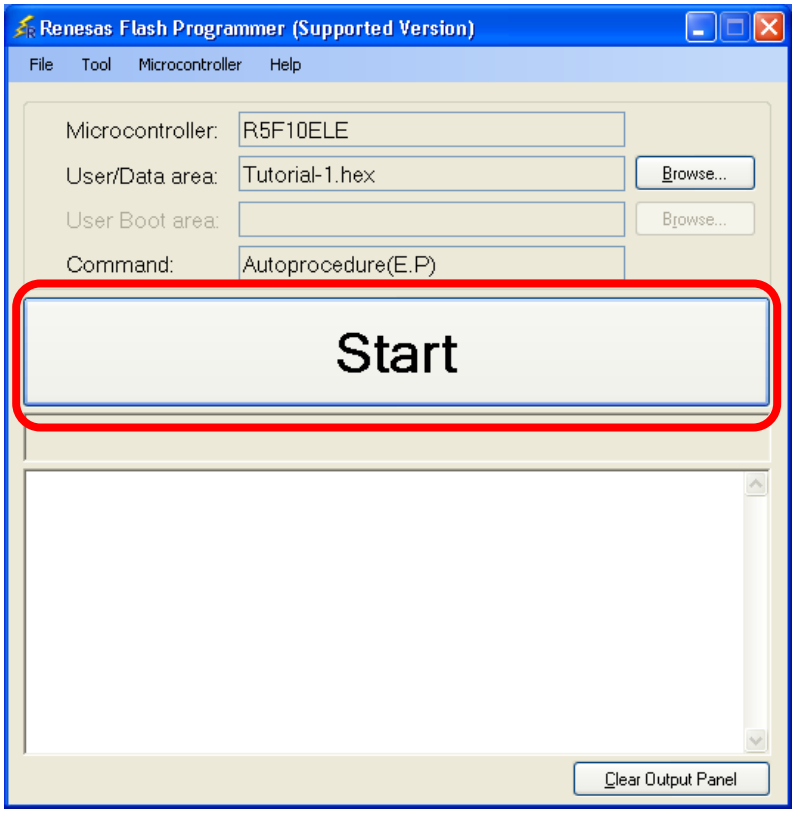
Click **Next**.



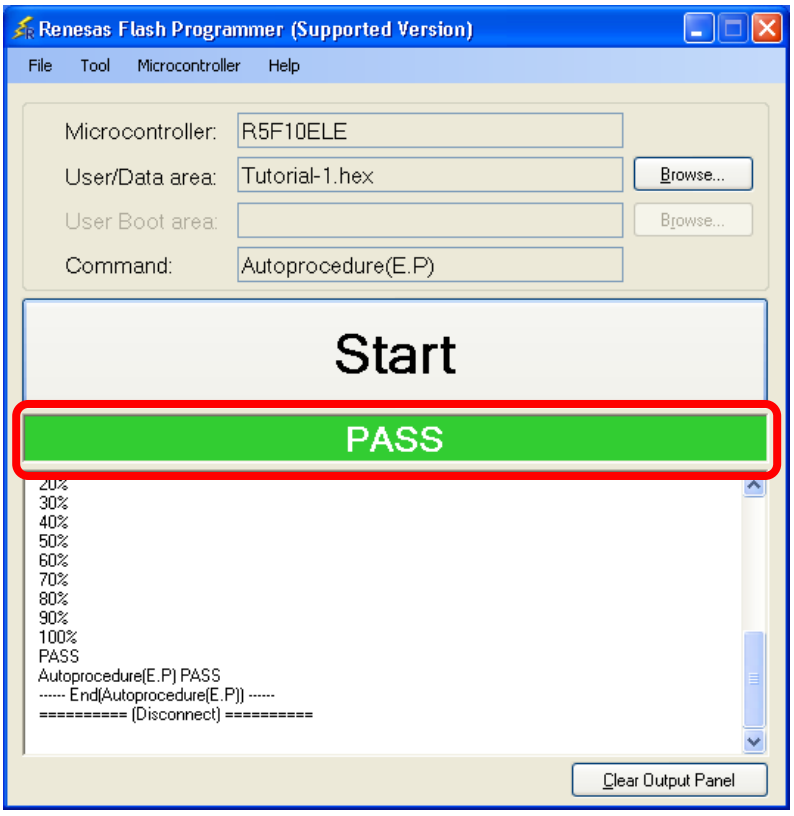
Click **Complete**.



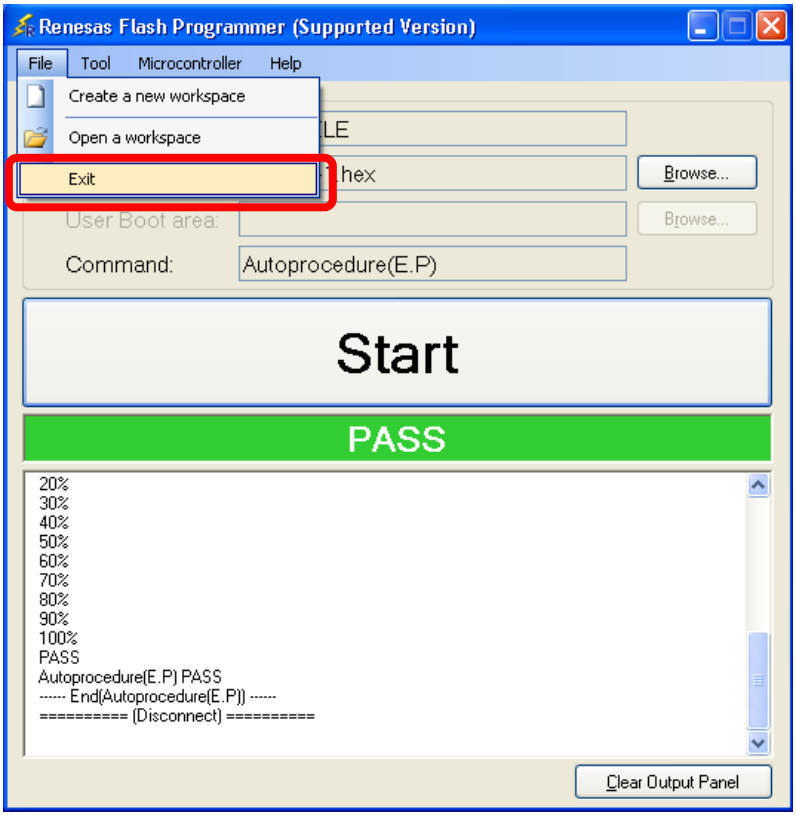
Click the **Browse** button and specify *filename*.hex from the project files created in the above procedure.



Click **Start**. Programming then starts.



When **PASS** is displayed, the program has been written successfully.



On the **File** menu, click **Exit** to terminate the Renesas Flash Programmer.

The program is now written.

Disconnect the cables between the computer and E1 emulator, and between the E1 emulator and evaluation board TSA-IC 500.

Do not disconnect the cable between the computer and evaluation board TSA-IC 500 because it is used to supply power to the evaluation board.

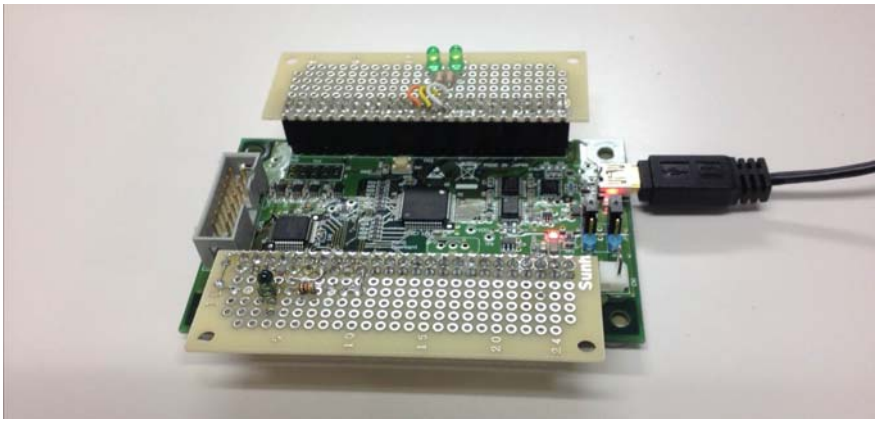
The program creation by using SA-Designer and CubeSuite+ is now complete.

Finally, check the operation.

8. Checking the operation on a real machine

Check the operation on a real machine.

Press the RESET (SW2) switch on the evaluation board to reset the IC on the evaluation board.



Actual system

Hold your hand above the sensor (seen on the near side in the above figure) to confirm that indication of the LEDs changes according to the luminance around the sensor.

The Smart Analog system has now been developed.

If you still have time to go further, try the following:

- Tune the sensitivity of the sensor by changing the gain setting in Renesas VA.
- Increase the number of LEDs.
- Light LEDs in different ways.

If your system do not work properly

If your system does not work properly, review the following:

- Read section 7 and confirm that the settings and program code are correct.
- Check that the evaluation board, sensor board, and LED board are connected correctly.

9. Supplement

This application note describes how to simulate the sensor operation, how to develop a program for microcontroller operation, and how to verify the operation on a real machine, for those who use Smart Analog for the first time.

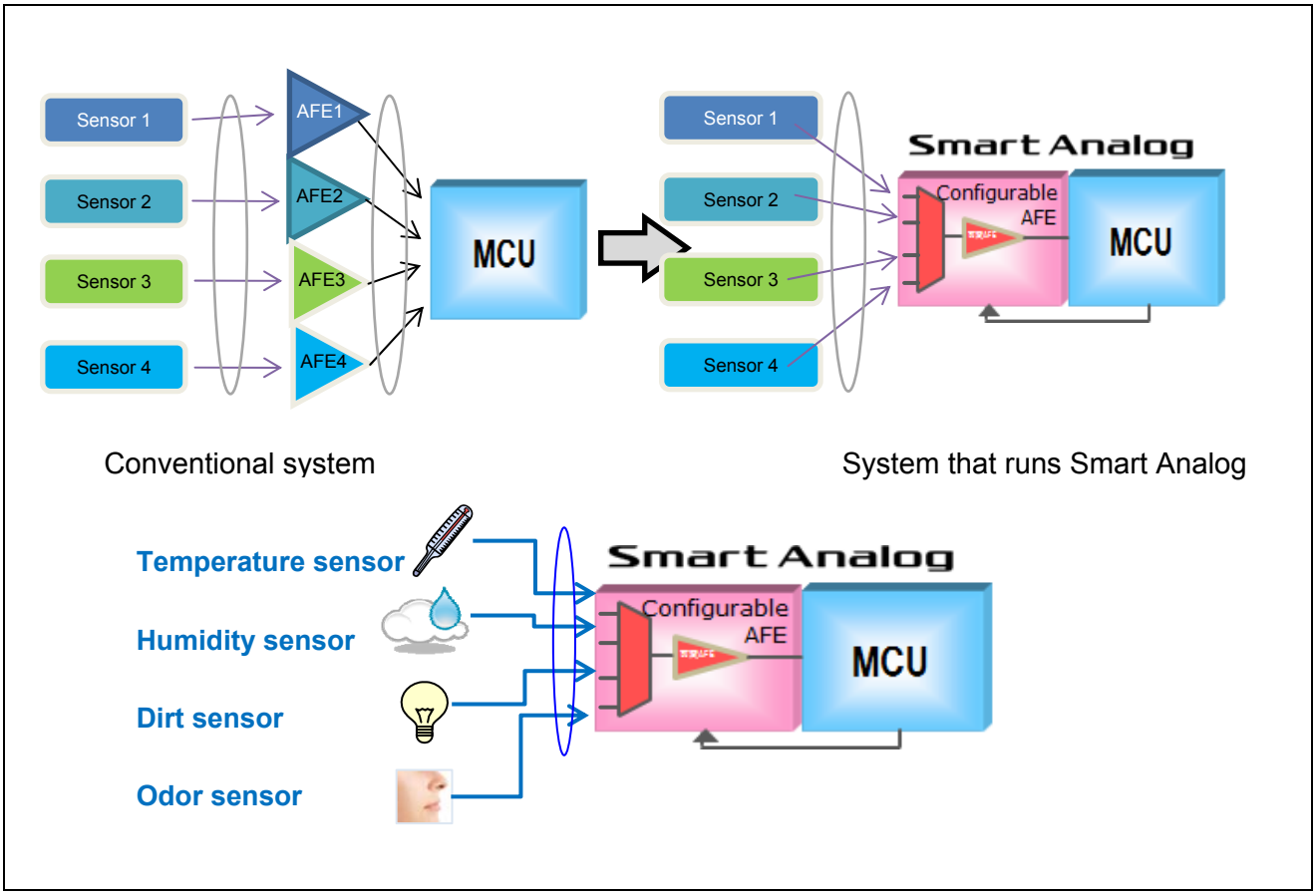
Based on the knowledge presented in this application note, we recommend you to try controlling other types of sensor and using Smart Analog internal features (such as filtering and synchronous detector).

Smart Analog products offer additional useful features. Three representative examples are provided below.

9.1 Multi-sensor control

A Smart Analog IC incorporates multiplexers with six channels, each of which has a differential pair on its input stage.

By using a multiplexer and the configurability (that enables changing the amplifier usage among inverting, non-inverting, differential, and transimpedance dynamically) of the built-in amplifier, you can use multi-sensor control that allows you to control multiple sensors with different control method at the same time.

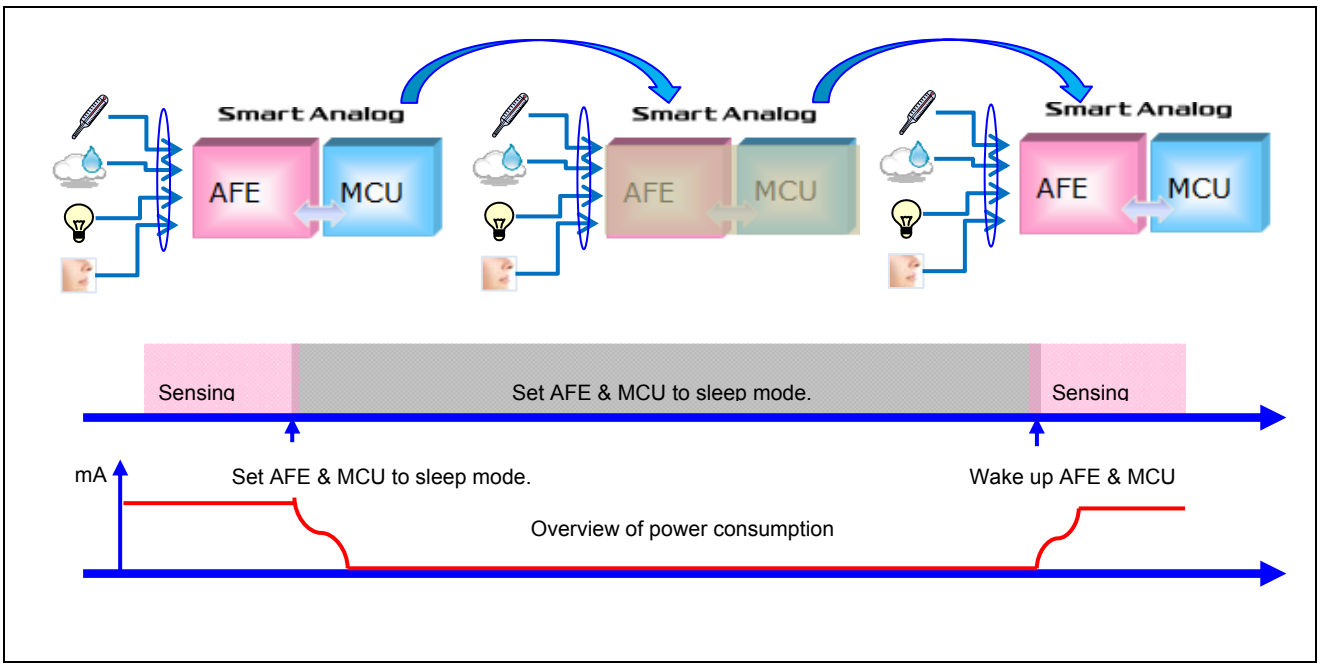


Example of multi-sensor control using Smart Analog

9.2 Low power operation achieved by intermittent operation

In a Smart Analog IC, the incorporated analog circuit blocks can be turned on and off individually.

By using this feature, you can control an analog circuit block so that it is turned on when it is sensing, and off when it is not sensing, which reduces the power consumption.

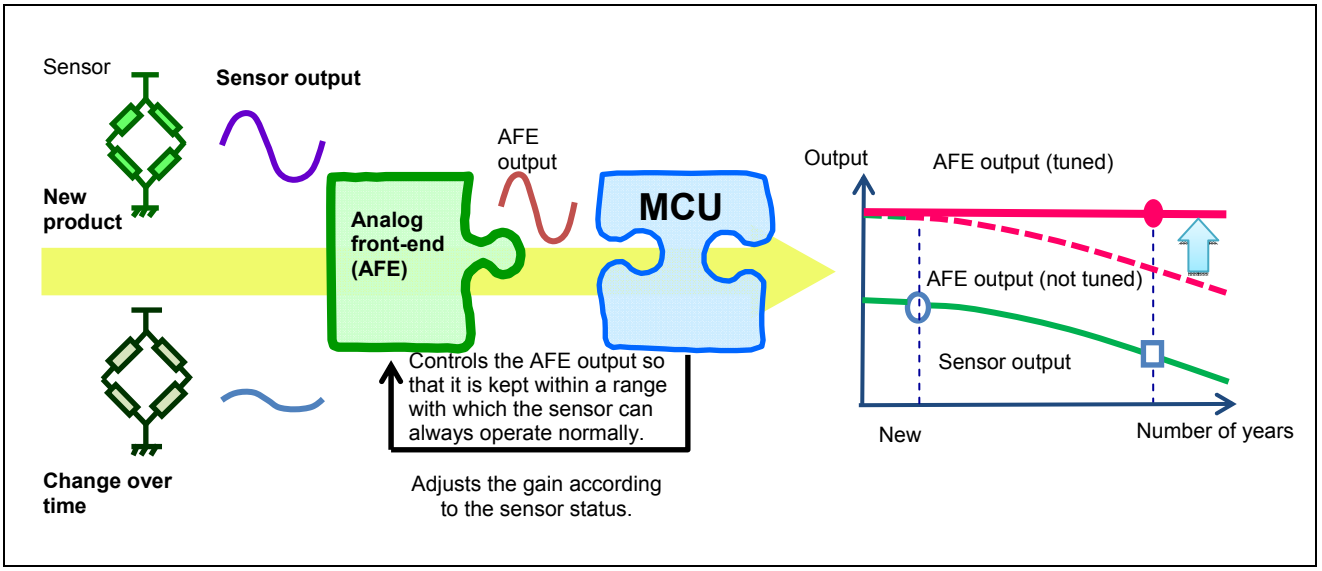


Example of low-power operation using Smart Analog

9.3 Auto calibration control

In Smart Analog ICs, you can change the parameters such as the gain and offset voltage for internal analog circuits dynamically.

By using this feature, a sensitivity drop caused by dirt or long-term sensor drift can automatically be corrected.



Example of auto calibration control using Smart Analog

In addition to the above features, you can use many other features.

Try your new idea with Smart Analog.

Website and Support

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

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

Revision Record

Rev.	Date	Description	
		Page	Summary
Rev.1.00	Mar 31, 2014	---	First edition issued

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this 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 an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

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

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

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

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, 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 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022/9044

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

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 4.0