

### ***Introduction to SANGO:***

Sample Program Generator and Organizer (SANGO) is the application deploying software, enabling system designers conceive, design and finish their projects with ease and confidence. SANGO generates C source code covering many project elements in the target applications at the press of a button. It also generates all the initialization code for most of the operating environments of the microcontroller. The generated code contains all the required comments for an easy understanding. Sango also contains an extensive documentation giving plenty of example circuits to test and use.

In short, starting your next application with Sango will definitely save a good amount of your precious development time.

### ***Features of SANGO.***

Sango generates the initialization routines required for the target micon like the clock setting, operating voltage and other required system facilities.

Sango generates various routines required for the internal peripherals. Different operating modes of the internal peripherals are taken into consideration for the code generation.

Since Sango generates the application code in C, a knowledge of assembly language is not required for the designers who migrate to R8C/Tiny devices from other controllers.

Size of the generated code and the RAM usage are automatically calculated for the designer's convenience.

The generated C code can be compiled with a single click without leaving the Sango environment.

Sango provides extensive documentation to motivate the designers to handle their code development without any trouble. This designer friendly documentation provides a wealth of design ideas with plenty of ready to use circuits. Also, the program flow is explained using flow charts for an easy implementation.

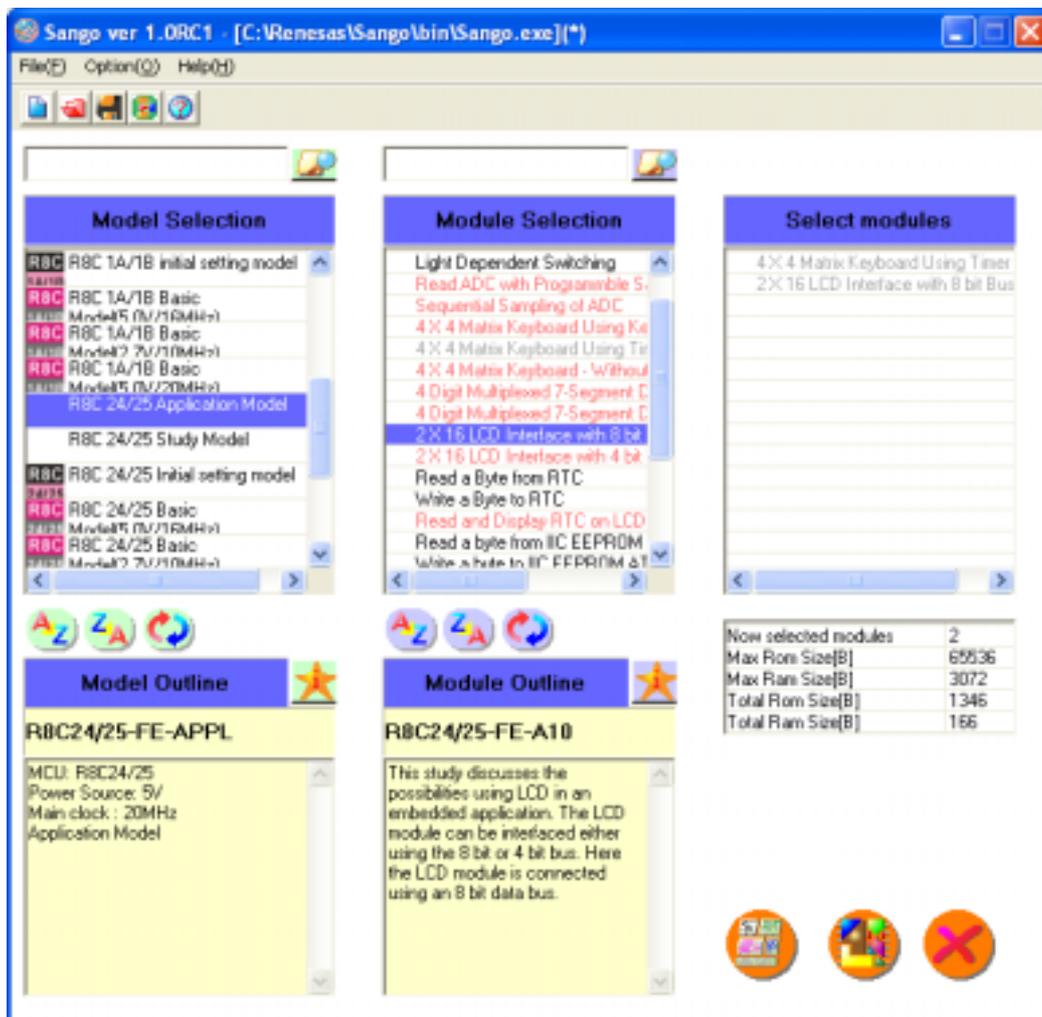
You need not refer the microcontroller data book for any initialization information since the Sango provides all the required details.

## How to generate code using SANGO?

The Sango has two different parts to configure the required application:

1. Models
2. Modules.

**Models** configure the R8C/Tiny microcontrollers as a whole operating at various possible options like voltage, frequency, etc. These models cover R8C/Tiny family devices: 11/13, 1A/1B and 24/25. Each model has different modules. When a model is selected, the modules associated with the selected model will be displayed in the modules window.



**Model outline window** gives a brief description of the selected model. The star icon in the window gives a detailed information on the selected model in the PDF format.

**Modules** part takes care of all peripheral features of the micon and many external embedded facilities.

**Module outline window** gives a brief description of the selected module. The star icon in the window gives a detailed information on the selected module in the PDF format.

**Selected Module window gives the list of the selected modules.**

Three buttons Generate, Compile and Close are present in the window. Use **Generate button** to generate C source codes after proper selection of models and modules.

### Step 1:

Select the suitable model (for example - R8C 24/25 Application model) from the **Model Selection Window**. Then the modules associated with the selected model will be displayed in the **Module Selection Window**.

### Note:

Select models based on R8C 24/25 microcontrollers, since R8C/25 family device is provided in the contest board.

### Step 2:

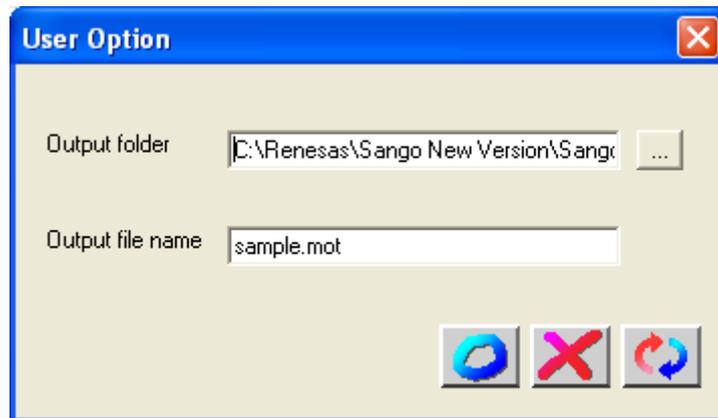
From the modules list, select the required modules one by one. Double click the required module. The module will be selected and will be displayed in the selected module window.

During this phase, the sango keeps track of the on-chip resources. When a peripheral facility is already used by an selected module, remaining modules which use the same peripheral facility will be disabled (Displayed in red color) for the selection.

The selected module will be displayed in gray color.

### Step 3.

Select the output directory using the **User Option** from the **Option(O)** menu. The C source codes for the selected modules will be generated in this directory. The output file name does not have much significance.



### Step 4.

Press the **Generate** button to automatically generate all the required C source code along with suitable header files ready for study and compilation.

During this code generating phase, Sango indicates the required size of program memory and the RAM usage for the application.

### Step 5.

Copy the generated files from the output directory and paste them in the new HEW project for further compilation. Refer to the "Quick Start guide for HEW " for detailed information.