

Study Flow.

To get an overall idea about the R8C/ tiny microcontroller, a study flow is organized as shown. You can follow this flow and if necessary you can skip the detailed study of the peripherals which are not required to start your design.

General features of R8C/Tiny.

Gives an overall picture of R8C/tiny family microcontrollers.

R8C/Tiny Architecture study.

Detailed information of the architecture, memory of the device. Refer to chapter 1 of R8C Software programming manual.

R8C/Tiny Instruction set study.

Information of the instructions, addressing modes and assembly language programming.

Clock Generation Circuit study.

Details of different internal and external clock generating circuits including clocks for on-chip peripherals. Refer to chapter 10 of hardware manual of R8C 24/25 micon.

For more details, you can refer following Sango module documents.

On-chip High Speed Oscillator Study.

On-chip Low Speed Oscillator Study.

External Oscillator Study.

Programmable I/O Port study.

You can get details of the port lines available in R8C 24/25 micon and different functions of each port line. Refer to chapter 7 of hardware manual of R8C 24/25 micon.

For more details, you can refer following the Sango module document.

I/O Port Study.

Timers Study.

Gives more details about different operating modes of timers available in R8C 24/25 micon. Refer to

chapter 14 of hardware manual of R8C 24/25 micon.

For more details, you can refer following the Sango module documents.

Timer - Waveform Generation Study

Timer - External Event Counter Mode Study

Timer - Pulse Width Measurement Study

Timer - One Shot Waveform Generation Study

Timer - Output Compare Mode Study

Timer - Watch Dog Timer Study

Timer - PWM Mode Study

Timer - Real Time Clock Mode Study

Analog to Digital Converter Study.

Gives more details about on-chip analog to digital converter operations in both 8 bit or 10 modes. Refer to chapter 18 of hardware manual of R8C 24/25 micon.

For more details, you can refer following the Sango module documents.

ADC-Study with 5V Reference

ADC-Study with Variable Reference

Interrupts Study.

Gives more details about interrupt system of R8C/Tiny. Refer to chapter 12 of hardware manual of R8C 24/25 micon.

For more details, you can refer following the Sango module documents.

Interrupts Study - 8 Bit Up/Down Counter

Interrupts Study - With Different Priority Levels

Keyboard Interrupt Study

Serial Port Study.

Gives more details about serial port of R8C/Tiny. Refer to chapter 15 of hardware manual of R8C 24/25 micon.

For more details, you can refer following the Sango module documents.

Serial Port - Multiprocessor Communication: Master

Serial Port - Multiprocessor Communication: Slave1

Serial Port - Multiprocessor Communication: Slave2

Clock Synchronous Serial Mode Study.

Gives more details about built-in peripherals like SSU and IIC of R8C/Tiny. The Refer to chapter 16 of hardware manual of R8C 24/25 micon.

For more details, you can refer following the Sango module documents.

Serial Port - SPI Bus Master

Serial Port - SPI Bus Slave

Steps to start a project:

1. Install the HEW software .
2. Install SANGO Part I and Part II software.
3. Copy the Flash Starter Application (FlashStarter.exe) from the CD to your directory on hard disk.
4. HEW (High Performance Embedded Workshop) is an Integrated Development Environment with built-in C compiler, Assembler and a Simulator.
5. Sango is the Sample Application Program Generator and Organizer for R8C/Tiny devices. Sango can be used to generate processor initialization and application code in C language for the internal peripherals and externals modules.
6. Open a new project in HEW to develop your program. Refer to the Quick Start Guide for HEW to create a new project.

7. Generate the required routines using SANGO and copy them to the HEW project. Refer to the Quick Start Guide for SANGO to generate codes. Also refer to SANGO to HEW.pdf file for more details.
8. Compile the project in the HEW and the final output file will be in .mot file.
9. Use R8C/Tiny System Simulator for verification of your code. Using this Simulator you can simulate your software before your hardware gets ready.
10. Program the code into the R8C/25 device using Flash Starter application. For programming the application code to the flash area of the microcontroller refer Quick Start Guide for Flash Starter.