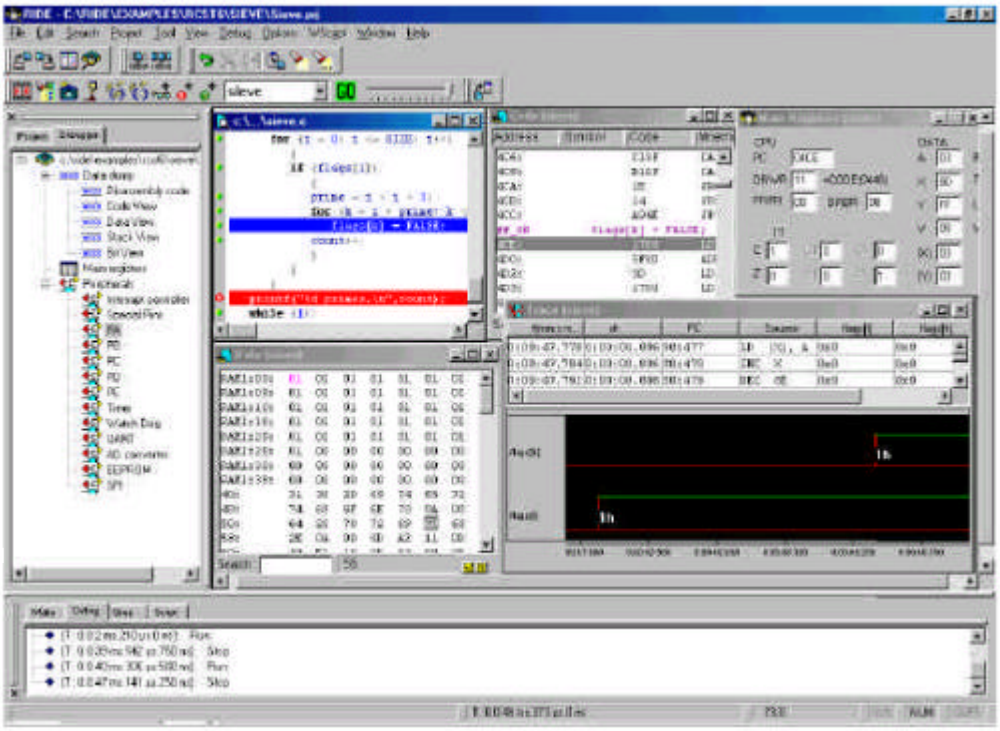
RKit-ST6

RKit-ST6 is a complete toolchain for the ST6 family of microcontrollers. It features an optimizing ANSI C Compiler, a Macro-Assembler, a Linker and a Simulator/Debugger. All these tools are smoothly and conveniently integrated into Raisonance Integrated Development Environment, which also provides drivers for the ST6 emulators.



Integrated into RIDE

RKit-ST6 is delivered with Raisonance Integrated Development Environment (RIDE), a fully integrated IDE featuring color syntax highlighting editor, project manager, on-line help and full control over all the tools of the toolchain. From editing to compiling, linking and debugging (with the simulator, starter kit or real time emulator), RIDE increases productivity by putting all the tools at your fingertips into a coherent and easy to use user interface.
When it comes to debugging, RIDE provides a rich variety of views into your application (Main registers, Hardware,

Project Manager

The project manager creates links between the various files that make up a project and the tools necessary to create that project. Its tree structure ease the management of the most complex applications.

Flexible Tools

Each of the RKit-ST6 tools are tightly integrated into RIDE, offering the professional developer a complete and cleanly integrated tool kit. In addition, each tool: assembler, compiler, linker, simulator can be run "stand-alone", as an individual, self-supported tool.

Optimizing ANSI-C Compiler

A super-set of ANSI-C

RC-ST6 implements the ANSI standards for the C language, extended with ST6 specific keywords:

Asm	At	code	data
Generic	Interrupt	intrinsic	sfr

Memory Models

The memory model specifies default location for variable declarations and default type for generic pointers. Two memory models are available: SMALL, for devices up to 4K Bytes and LARGE for bigger devices that require a bank switching Mechanism.

Base Types

Integer Types:
- 8 bits : "signed char" and "unsigned char",
- 16 bits : "signed int" and "unsigned int",

Pointers Types

Two types of pointers are always available, generic pointers and space qualified pointers.

Code Optimizations

RC-ST6 optimizes the code to be as compact and fast as possible. Nevertheless, when a choice is to be made RC-ST6 has been designed to put emphasis on the Code Size rather than the Speed.

Libraries

RC-ST6 is supplied with ANSI C standard libraries as described in: stdio.h, string.h, ctype.h, and stdlib.h standard header files.
Specific ST6 libraries are provided to allow a fast configuration of all the internal peripherals present in the ST6 derivatives.

Implementation

RC-ST6 Compiler and the libraries are fully autonomous and do not require the use of any other coding tool. A C function can be called from an assembly program, and can call routines written in other languages.

ST6 Specific Features

Local variables are located into overlaid data segments to mimic a stack behavior.
RL-ST6 provides an accurate control onto the hardware stack.

MA-ST6 Macro-Assembler

MA-ST6 is a comprehensive assembler, associated to a powerful macro-preprocessor. In connection with the RL-ST6 Linker/locator, MA-ST6 provides an easy control on the bank switching process mechanism. MA-ST6 features a preprocessor that accepts and translates the former AST6 syntax.

RL-ST6 Linker/Relocator

RL-ST6 is an optimizing linker/locator which acts not only on the code generated by the RC-ST6 Compiler, but also on the code written directly in assembler. The RL-ST6 linker supports a Bank Switching mechanism for both code segments and data segments. The overlay analyzer allows automatic optimization of the data space usage. RL-ST6 produces a symbolic output format, a HEX output file, and a complete listing file reporting all the optimizations performed.

SIMICE-ST6 Simulator/Debugger

The simulator is tightly integrated into RIDE and uses the information generated by RC-ST6 or MA-ST6 tools to provide full symbolic high level debugging.

SIMICE-ST6 takes into account the characteristics of the selected device and simulates (in considerable detail) all internal peripherals.
Associated with RIDE, SIMICE-ST6 allow simulation of multi-processor applications, and offers various solutions that can simulate external inputs and outputs.

SIMICE-ST6 provides various emulator-like functions such as trace management or complex breakpoint control. SIMICE-ST6 can also be run as a stand alone simulator.
SIMICE-ST6 features various analysis tools as Code Coverage or a Performance Analyzer.

Hardware Drivers

RIDE provides drivers for official STMicroelectronics ST6 emulators and for some third party emulators.
All emulators are driven through exactly the same interface as the simulator.