Raisonance Tools for ARM

Raisonance Tools for ARM core-based microcontrollers

Getting Started

Document version
25 July 2019

Contents

1. Introduction

This guide should be used by anyone with an interest in Ride7 for ARM.

1.1 Purpose of this manual

This manual helps you get started using any of the Raisonance products that are offered by Raisonance's application development solution for ARM core-based MCUs. It describes how to compile and debug your application or the included sample applications. It assumes that you have the prerequisite knowledge of the C and ARM / thumb / thumb2 assembly languages and CPUs.

It also supports ST-Link/V2 (in STMicroelectronics' Discovery, Nucleo and other evaluation boards).

1.2 Scope of this manual

The tools specifically addressed in this manual include:

- GCC C compiler toolchain
- Ride7 development environment
- RFlasher7 programming interface
- RLink debugger programmer

1.3 Related documents

Each tool in this document has its own user manual. These documents are provided with the Raisonance software installation and in ***http://support.raisonance.com/***

- RLink user manual
- Open4 (EvoPrimer) user manual
- REva v3 user manual
- REva v2 and earlier user manual
- REva STM32 daughter boards user manual
- REva STRx daughter boards user manual
- Ride7 Online help (provided with installation)
- RFlasher7 user manual
- GCC toolchain user documentation (provided with installation)

1.4 Additional help or information

If you want additional help or information, if you find any errors or omissions, or if you have suggestions for improving this manual, go to the IoTize site for Raisonance microcontroller development tools www.raisonance.com, or contact the microcontroller support team.

Microcontroller website:	www.raisonance.com
Support extranet site:	support.raisonance.com (software updates, registration, bugs database, etc.)
Support Forum:	forum.raisonance.com
Support Email:	support@raisonance.com

1.5 Raisonance brand microcontroller application development tools

February 1, 2017, Raisonance became the brand under which the company IoTize sells its microcontroller hardware and software application development tools.

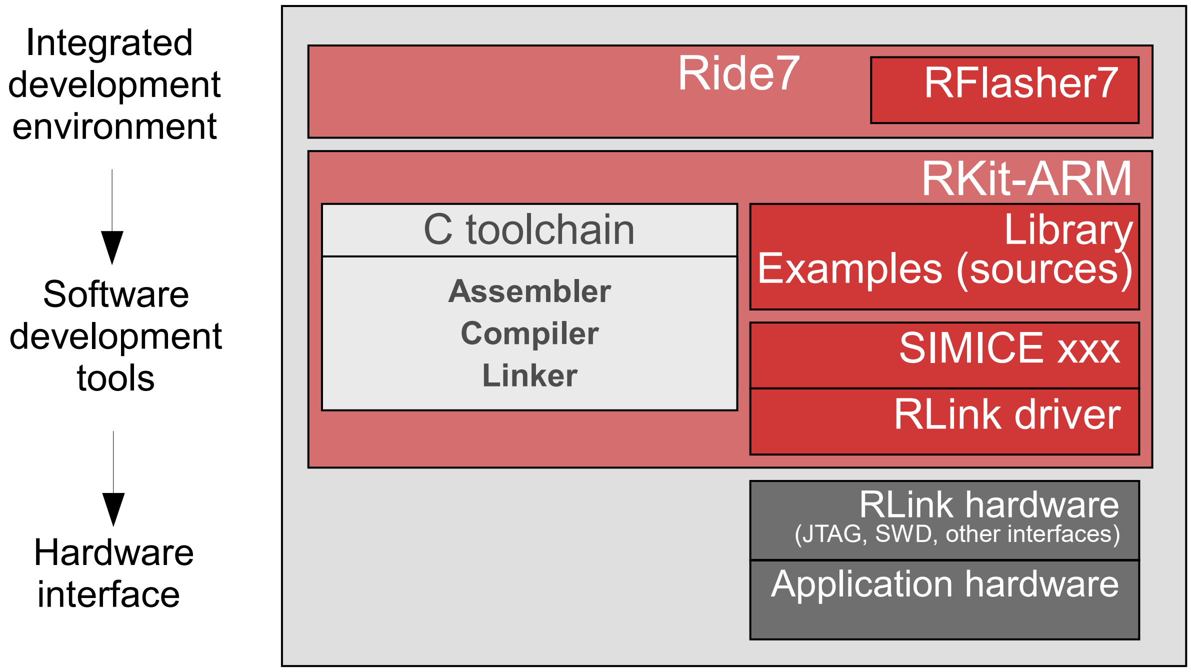
All Raisonance branded products regardless of their date of purchase or distribution are licensed to users, supported and maintained by IoTize in accordance with the companies' standard licensing maintenance and support agreements for its microcontroller application development tools. For information about these standard agreements, go to:

Support and Maintenance Agreement:	http://www.raisonance.com/warranty.html
End User License Agreement:	http://www.raisonance.com/software-license.html

2. Raisonance tools for ARM overview

Raisonance's integrated development environment, Ride7 for ARM®, interfaces with a range of development tools including:

- The tools that make up Ride7 IDE...
  - Editor
  - Debugger
  - Project manager
  - Plugins manager
  - RFlasher7 Programming Interface (this tool has its own doc)
  - RLink USB driver installer
- RKit-ARM, which integrates...
  - GNU GCC toolchain for ARM
  - Software ARM simulator
  - Hardware debugger graphical interface
  - RLink debugger interface driver
- RLink USB to JTAG/SWD programming and debugging dongle from Raisonance.
- REva boards, STM32-Primers, EvoPrimers, Open4 devices (these all include an RLink).



2.1 Ride7

Ride7 is the user interface for all the tools. It gives you start-to-finish control of application development including code editing, compilation, optimization and debugging.

2.2 RFlasher7

The RFlasher7 interface can program the Flash memory of the target MCU. It is installed with Ride7.

See the RFlasher7 documentation for instructions about this interface.

The ARM-specific and RLink-specific configuration of RFlasher is the same as in Ride7. See the debug section of this document (this one) for information concerning that.

2.3 RKit-ARM

The RKit-ARM is an add-on to Ride7. It includes a C toolchain (GCC) controlled directly from Ride7, and defines the devices, script tools, and debugging and programming interface features (see Section 7 for details).

- **Lite** license includes default Raisonance hardware tools, GCC toolchain, and limited software features.
- **Enterprise** license includes support for 3rd party compilers and C++ programming.

2.4 SIMICE ARM simulator

Raisonance's SIMICE simulates the ARM core (including all memory space) and some ARM peripherals, or serves as a graphical interface when you program and debug your ARM CPU using an RLink JTAG/SWD standard emulator (or products that include it).

SIMICE ARM is included in the RKit-ARM-Lite. Complex peripherals (USB, CAN) and some uncommon peripherals are not simulated.

The same user interface is used for the simulator and the hardware debugging tools (RLink).

2.5 RLink

RLink is a JTAG/SWD standard emulator with a USB interface. It allows you to program ARM devices on an application board and debug the application while it runs on the target. Raisonance's REva evaluation boards feature an embedded (detachable) RLink.

RLink-STD (and Primer, REva, EvoPrimer, Open4, etc.) used with **RKit-ARM Lite** are **limited** to a code size of half the target device's Flash or 64 K bytes, whichever is smallest.

RLink-PRO for ARM (native PRO or upgraded STD) with any version of RKit-ARM allows debugging with **unlimited** code size.

RKit-ARM Enterprise with any version of RLink allows debugging with **unlimited** code size.

Flash programming is never limited.

Note: RLinks have various capabilities for programming and debugging ARM and other Ride7-supported target microcontrollers, depending on the type of RLink and the installed and registered software.
Your RLink's capability is shown when Ride7 reads your RLink's serial number during the Connect to Debugger test.
For a description of the different debugging capabilities, refer to *Debugging with Hardware Tools*.

2.6 TapNLink

TapNLink is a IoT device that, among other features, can connect with a PC using BLE, and with a Cortex device using SWD. It allows you to program Cortex-M devices on an application board and debug the application while it runs on the target, without any wire connection with the debugging PC.

2.7 Licenses

Any files not mentioned here are under license by IoTize or other companies. They should not be copied or distributed without written agreement from their owners.

- The GNU toolchain is under the GPL license, which makes it free to use without royalties, as are some of the files written by Raisonance and ST. You can assume that everything under the *arm-gcc* and *GNU* sub-directories of the Ride7 installation folder can be freely used, copied and distributed, but is without any warranty and only limited support from Raisonance.
- Raisonance's solution for ARM core based MCUs has two levels of license:
  - Hardware license: controls the hardware-based debug limitation of 64 K bytes of code in RAM or Flash memory.
  - Software license: controls software features, details are provided at http://www.mcu-raisonance.com/software_packages_arm.html

2.8 Supported devices and tools

2.8.1 ARM MCUs

The ARM-core based microcontrollers addressed by this solution include:

- ARM Cortex™ (M3, M4, M0)
  - STMicroelectronics (STM32F, STM32L, STM32W)
  - NXP (LPC17, LPC11)
  - Texas Instruments / Stellaris (LM3S)
  - Silicon Labs / Energy Micro (EFM32)

Support is provided for some sub-families that are based on legacy ARM9 and ARM7 cores, including:

- ARM966E (supported legacy devices) STMicroelectronics (STR9)
- ARM7TDMI (supported legacy devices)
  - STMicroelectronics (STR7)
  - NXP (LPC21, LPC23, LPC24)

For a complete listing of specific supported MCUs and derivatives refer to the list of selectable MCUs in the Ride7 project settings. For this purpose, users can install Ride7 with an evaluation version of RKit-ARM (Download at: http://www.mcu-raisonance.com/arm-download.html). Newly supported derivatives are reported in the release notes for each version.

2.8.2 Third party tools used in conjunction with Ride7 for ARM

Ride7 for ARM can be used together with a number of third-party tools including:

- **GNU GCC toolchain** for ARM® (ARM-none-eabi-gcc, ARM-none-eabi-as, ARM-none-eabi-ld): they allow you to compile applications in assembler and/or C language. Ride7 automatically installs and calls the free open-source GNU toolchain. See http://www.gnu.org/ for more information about GNU programs. The GCC toolchain is maintained by ARM engineers. You can get the sources and binaries (new or old versions), report bugs and ask compile-chain-specific questions here: https://launchpad.net/gcc-arm-embedded . If you encounter problems with Ride's integration of GCC (including using old Ride projects with a newer version of Ride/RKit-ARM/GCC), please check our forum and FAQ, which will be updated more frequently than this document: http://forum.raisonance.com/index.php
- **ST-Link/V2**: ST-Link is a JTAG/SWD standard emulator with a USB interface designed and produced by STMicroelectronics. It allows you to program STM32 devices on an application board and to debug the application while it runs on the target. Most of ST's evaluation boards (including Discovery and Nucleo) feature an embedded (sometimes detachable) ST-Link.

Note: RKit-ARM 1.52 and later versions do not provide specific support for the Phyton CodeMaster-ARM compiler toolchain in Ride7. Customers using CodeMaster-ARM compiler may continue to use the specific support feature under previous versions that offered this. Users of RKit-ARM 1.52 and later versions can use the CodeMaster-ARM compiler with a makefile using the Makefile toolchain of Ride.

Note: RKit-ARM 1.46 and later versions do not support the JTAGJet emulator from Signum in Ride7. Customers using JTAGJet can either stick with an older version of RKit-ARM or use the Signum software, or use an RLink.