



RLink

Getting Started

Document version
24 March 2017

Contents

1. INTRODUCTION.....	4
1.1 License.....	4
1.2 Purpose of this manual.....	4
1.3 Scope of this manual.....	4
1.4 Additional help or information.....	4
1.5 Raisonance brand microcontroller application development tools.....	4
2. RLINK VERSIONS.....	5
2.1 RLink1.xx.....	5
2.2 RLink2.xx.....	5
2.3 RLink3.xx.....	5
2.4 RLink4.xx.....	6
3. INSTALLING RLINK.....	7
3.1 Automatic installation of the RLink USB driver.....	7
3.2 Manual installation of the RLink USB driver.....	8
3.3 Uninstallation of the RLink USB driver.....	9
3.4 Alternative recovery procedure.....	10
3.5 Reading RLink serial number.....	11
3.5.1 Reading RLink serial number from Ride7.....	11
3.5.2 Reading RLink serial number from RFlasher.....	11
3.5.3 Reading RLink serial number from any Windows command prompt.....	11
3.6 Register and activate the software.....	11
4. USING RLINK.....	12
4.1 Before using RLink.....	12

RLink Getting Started

4.2 Using RLink with different target CPUs.....	12
4.3 Using RLink with the host software.....	12
4.4 RLink connectors.....	13
4.5 Adapter schematics.....	13
4.6 Connecting and powering.....	14
4.6.1 Avoiding electrical conflicts.....	14
4.6.2 Standard connecting sequence.....	14
4.6.3 Hot Plug complete sequence.....	15
4.6.4 Hot Plug simplified sequence.....	16
4.7 Operating conditions.....	17
4.8 Support and FAQ.....	17
4.9 Checklist of items to prepare before contacting support.....	18
5. CONFORMITY.....	20
6. GLOSSARY.....	21
7. INDEX.....	22
8. HISTORY.....	23

1. Introduction

RLink is a versatile high-performance microcontroller debugger and programmer which supports a range of target interfaces (JTAG, SWD, SWIM, ICC) and can connect to a wide range of 32-bit and 8-bit microcontrollers to program the target device and debug applications in real-time. The RLink is available in a number of formats: as a standalone product, embedded on a board, or as part of a package from another company.

1.1 License

Purchase of an RLink includes the appropriate Ride7 RKit software, and includes upgrades and support for a year. After one year, if you require support, or if you wish to upgrade the kit, you will need to purchase a new license for the RKit.

1.2 Purpose of this manual

This manual gives an overview of the RLink and links to the documentation that is relevant to RLinks. The RLinkTools_API software interface allows users to manage the RLink 24-pin connector I/Os to customize the use of the RLink, this is detailed in the RLinkTools_API user manual.

This manual does NOT explain how to configure and use the Raisonance software for communicating with the target CPU through RLink. This is explained in the Ride7 Getting Started document dedicated to the Ride7 support of this CPU family: GettingStartedARM, GettingStartedSTM8, etc. Please refer to these documents for information on how to actually use the RLink.

1.3 Scope of this manual

The manual provides an overview that is applicable to all RLinks.

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. RLink versions

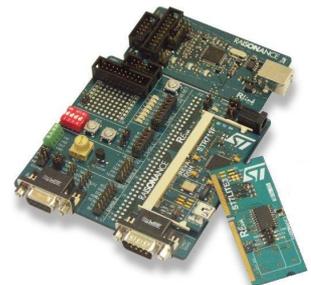
2.1 RLink1.xx

This independent device is not produced anymore. It works in the 3V-5.5V range. It can only be connected to a μ PSD and is in a white plastic box.



2.2 RLink2.xx

RLink version 2.xx is not produced anymore. It works in the 2.5V-5.5V range. It is attached to a REva2,xx board and can be snapped off if required. It can drive ARM, ST7, STM8, PPC and μ PSD devices.



2.3 RLink3.xx

Version 3.xx is the current version. This RLink can be:

- stand-alone, inside a grey plastic box,
- part of a REva3.xx board (cannot be snapped off),
- part of a Primer: ST7FOX-Primer, STM8A-Primer, STM3210B-Primer (circle), etc.

It is compatible with μ PSD, ST7, STM8 and ARM devices. However, when it is part of a Primer it can only be used with the CPU in the Primer.

It works in the 2.5V-5.5V range, and down to 1.5V with some protocols and adapters, like for example STM8-SWIM.

It existed in two formats, Standard and Pro, which both offer unlimited Flash programming of all supported 8-bit and 32-bit devices. The Pro is not produced any more, but the Standard can be upgraded by software to provide the same features as the Pro.



RLink-STD provides unlimited debugging of applications on supported 8-bit devices (STM8, ST7, ...) via the same connection while driven from the Ride7 integrated development environment. Debugging of supported 32-bit devices (ARM, PPC) may be limited depending on your target CPU and software license. (See your software tool documentation for details). Flash programming is not limited. This is the default type of RLink, present in REva boards and all Primers. They can be upgraded by software to have the same capabilities as the RLink-Pro.



RLink-PRO provides unlimited debugging of applications on supported 8-bit and 32-bit devices. It only exists as a stand-alone version, but all standard RLinks, including the ones in REva boards (even REva2.xx) and in Primers can be upgraded to the PRO specification by software. There are also a number of specific versions of this RLink, for PPC, CoolRISC, BIRD, and some confidential target microcontroller architectures. Most of the information in this doc about USB drivers, voltage range, connections, support, etc. also applies to these special RLinks. However, you should consult the target-specific documentation for more details, hints, warnings, etc. RLink-PRO is not produced anymore. Customers who own it can still use it with the new software. New customers who want these features should purchase an RLink-STD and a software upgrade.

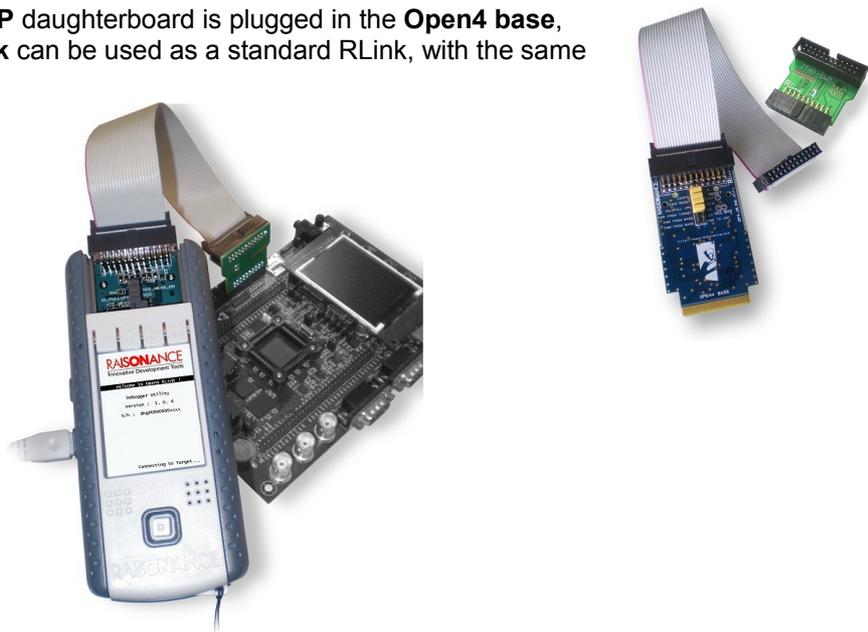
2.4 RLink4.xx

Version 4.xx is the next generation, and is already included in some products. It is sometimes called "RLinkP", which is different from "RLink-PRO".

The **Open4** base includes an **RLink4**. When a daughterboard is plugged in the base, as in a Primer the RLink can only be used with the CPU that is present on the daughterboard.



When the **OP4-RLink-ADP** daughterboard is plugged in the **Open4 base**, the resulting **Open4-RLink** can be used as a standard RLink, with the same adapters as a Rlink3.xx.



With current software versions and adapters, this RLink:

- Works only with Cortex-M (SWD) and STM8 (SWIM) devices.
- It does NOT support ARM7, ARM9, uPSD, PPC (JTAG), ST7 (ICC), etc.
- It supports SWO low-level trace with Cortex-M devices (which the older RLinks do not support).
- It can work in the 2.5V-3.3V range, down to 1.5V with the STM8-SWIM protocol.
- Like the RLink3, the RLink4 comes in STD and PRO versions, with the same differences (debug limitation for 32-bit devices, possibility to upgrade from STD to PRO by software, etc.).
- Future versions of software and hardware products (possibly just new adapters) will add more features and flavors to this RLink.

3. Installing RLink

RLink uses Raisonance software to connect to a Windows USB driver and to communicate with the RLink via USB. This USB driver and its configuration file are provided by Raisonance along with Ride7. There are two USB drivers:

- WinDriver, from Jungo, is used for Windows XP32 with SP3 (SP3 is required).
- WinUSB, from Microsoft, is used for Windows Vista32, Vista64, and all newer versions (7, 8).

Note: XP64 is not supported.

3.1 Automatic installation of the RLink USB driver

Here is the normal procedure for installing the RLink driver on all supported versions of Windows:

1. Make sure the RLink is NOT connected to the PC.
2. Make sure you are logged in as Administrator, or a user that has administrator rights.
3. Install Ride7 or run RLinkUSBInstall.exe. In Vista, you must run this in 'run as admin' mode. This program detects which version of Windows you are running and installs the correct driver (WinDriver or WinUSB) accordingly. On Vista and recent versions of XP, it pre-installs the driver in Windows DriverStore (or DRVSTORE), on older versions, it will install it directly in the Windows drivers directory.
4. Remove any CDs from the drives. This is important. You really must do it!
5. Plug in the RLink. Check that the PWR and BUSY LEDs turn ON. The PWR LED means that the RLink is powered (!) and the BUSY LED means that the RLink firmware is running correctly. This validates most of the RLink hardware and firmware.
6. Tell the 'Found new hardware' wizard from Windows not to search on the internet.
7. Tell the 'Found new hardware' wizard from Windows to search automatically in Windows but do not point it to other places (this should be the default answer).
8. It should then find the driver, install it, and associate it with this RLink.
9. Then you should see the BUSY LED turning OFF, which means that Windows and the RLink are able to communicate through the USB driver and cable.
10. Check with RLinkCapab that you are able to read the RLink serial number as explained later in this documentation.

Note: You **must** install the driver **before** you plug the RLink on the PC. Plugging the RLink before installing the driver can lead to the installation of incorrect files in Windows, preventing you from performing the installation afterwards. If this happens, you will have to uninstall all RLink drivers, as explained later in this document.

3.2 Manual installation of the RLink USB driver

This procedure should only be used when the automatic procedure fails, which can happen on some versions of Windows with special configurations.

For XP (and earlier) users:

1. Make sure the RLink is NOT connected to the PC.
2. Make sure you are logged in as Administrator, or a user that has administrator rights.
3. Install Ride7.
4. Remove any CDs from the drives, including the Ride7 CD (this is important!).
5. Open a command prompt.
6. Change Directory to `<Ride>\Driver\RLinkDrv\Jungo_WinDriver_2000_NT_XP\`
7. Execute `wdreg -inf windrvr6.inf install` (watch out for error messages).
8. Execute `wdreg -inf RLinkDrv_WDP.inf install` (watch out for error messages).
9. If you have a recent version of XP, or if you have automatic updates activated, check that there are two directories in `C:\WINDOWS\SYSTEM32\DRVSTORE\...`, one with the name starting with `windrvr6`, and the second with `RLinkDRV`.
10. Plug in the RLink. Check that the PWR and BUSY LEDs turn ON. The PWR LED means that the RLink is powered (!) and the BUSY LED means that the RLink firmware is running correctly. This validates most of the RLink hardware and firmware.
11. Tell the Windows 'Found new hardware' wizard not to search on the internet.
12. Tell the Windows 'Found new hardware' wizard not to search automatically, but point it to `<Ride>\Driver\RLinkDrv\Jungo_WinDriver_2000_NT_XP\`
13. It should then find the driver, finish to install it, and associate it with this RLink.
14. Then you should see the BUSY LED turning OFF, which means that Windows and the RLink are able to communicate through the USB driver and cable.
15. Check with RLinkCapab that you are able to read the RLink serial number as explained later in this documentation.

For Vista, Win7 (and later) users:

1. Make sure the RLink is NOT connected to the PC.
2. Make sure you are logged in as Administrator, or a user that has administrator rights.
3. Install Ride7 in 'Run as Admin' mode.
4. Remove any CDs from the drives, including the Ride7 CD (this is important!).
5. Open a command prompt in 'Run as Admin' mode.
6. Change Directory to `<Ride>\Driver\RLinkDrv\MS_WinUSB_XP_Vista\`
7. Execute `RLink_WinUSB_Install_verbose.bat` (watch out for error messages).
8. Check that there is a directory in `C:\Windows\System32\DriverStore\FileRepository\...` with the name starting with `mlink_winusb`.
9. Plug in the RLink. Check that the PWR and BUSY LEDs turn ON. The PWR LED means that the RLink is powered (!) and the BUSY LED means that the RLink firmware is running correctly. This validates most of the RLink hardware and firmware.
10. Tell the Windows 'Found new hardware' wizard not to search on the internet.
11. Tell the Windows 'Found new hardware' wizard not to search automatically, but point it to `<Ride>\Driver\RLinkDrv\MS_WinUSB_XP_Vista\`
12. It should then find the driver, finish to install it, and associate it with this RLink.
13. Then you should see the BUSY LED turning OFF, which means that Windows and the RLink are able to communicate through the USB driver and cable.
14. Check with RLinkCapab that you are able to read the RLink serial number as explained later in this documentation.

3.3 Uninstallation of the RLink USB driver

Sometimes you need to completely uninstall the RLink driver. Normally it is because a wrong driver (for the wrong version of Windows) has been installed. If that happens, you must completely uninstall the driver, which is no easy task.

You must get to the point where when you plug in RLink and let Windows search automatically for a driver, it doesn't find any. Only then can you install the correct driver using one of the procedures described previously. If you cannot reach this point, see the next subsection which gives a method to force Windows to use the driver you choose.

Standard procedure

Here is how to uninstall the RLink driver:

1. Plug in the RLink.
2. If a driver is loaded for it, go to the Device Manager and ask Windows to uninstall it (the driver, not just the RLink).
3. If you see a "Jungo" section in the Device Manager, you should also try to uninstall it.
4. Unplug the RLink.
5. Remove all the files, directories and registry keys (using *regedit.exe* started as admin) in the list below. In *DriverStore* (Vista) or *DRVSTORE* (XP), try to remove the whole directories. If you cannot, remove as many of the files in them as you can. Also try to remove the registry entries, you will need to change the ownership to yourself, recursively, then give "everyone" complete access to the keys, directories and files, then you should be able to remove them.
6. Go back to the first point of this procedure, plug in the RLink, and repeat the process until Windows does not find any driver for the RLink when you ask it to search automatically. If after several tries it still finds one, go to the Device Manager, display the details of the driver that is associated with RLink, and try to locate the *inf* and/or *sys* files that are used. Then, repeat the procedure and make sure that you remove those files.

Other items to remove (if they exist)

Files and folders to remove:

- *C:\Windows\inf\Windrvr6.**
- *C:\Windows\inf\RLink*.**
- *C:\Windows\System32\Drivers\Windrvr6.**
- *C:\Windows\System32\Drivers\RLink*.**
- *C:\Windows\System32\DriverStore\FileRepository\Windrvr6*.**
- *C:\Windows\System32\DriverStore\FileRepository\RLink*.**
- *C:\Windows\System32\DRVSTORE\Windrvr6*.**
- *C:\Windows\System32\DRVSTORE\FileRepository\RLink*.**
- search for all *C:\Windows\inf\oem*.inf* files that contain the string "RLink" or "Jungo". If you find any, you must remove them, and also the associated *oem*.pnf* files.

Registry key to remove:

HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Enum/USB/VID_138E&PID_9000/...

3.4 Alternative recovery procedure

In some situations, uninstalling the USB driver as described in the previous section can be difficult, because Windows keeps hidden copies of the drivers. If after several attempts you still don't manage to get to the point where Windows doesn't find any driver for the RLink, then you can follow this procedure.

This procedure should only be used if the normal procedure described in the previous section has failed, because this second procedure must be done for all your RLinks. The normal procedure need only be performed once even if you have several RLinks.

1. Manually install drivers with RLinkUSBInstall.exe from <Ride>\Drivers\....
2. Plug in the RLink, let Windows associate it with the (wrong) driver it already has.
3. In the Device Manager, Right-click on *RLink* -> choose **Update Driver**.
4. Select **Do not connect to Windows Update** -> Click **Next**.
5. Select **Software from list** -> Click **Next**.
6. Select **Don't search, I will choose...** -> Click **Next**.
7. Select **List compatible hardware** ->
Choose **RLink USB** if in XP, **RLinkWinUSBDevice** if in Vista/Win7 -> Click **Next**.
8. **Finish**.

Note: The exact sequence and quotes above might differ a little depending on the version and configuration of Windows, and of course the language. But all versions of Windows should propose this procedure or an equivalent.

3.5 Reading RLink serial number

Reading the serial number of the RLink is the simplest way to check that the USB communication between RLink and the host PC is correctly configured and working. If this works, it validates the USB driver and its installation, the USB cable, and the USB and firmware parts of the RLink.

You will be asked for your RLink's serial number whenever you contact our support team about RLink.

Note: This serial number is different from that used for **registration**. The latter corresponds to a **license**, not to an RLink product.

3.5.1 Reading RLink serial number from Ride7

1. Open an ST7, STM8 or ARM project.
2. Choose **RLink** as **Debug Environment** in the **Advanced Options**.
3. Select **Options > Project Properties > RLink Configuration > Advanced Options**.
4. Click on the box **Connect to RLink and read Serial Number**.

3.5.2 Reading RLink serial number from RFlasher

1. Open RFlasher.
2. Select an ST7, STM8 or ARM **Target CPU**.
3. Click the button on the panel named **Connect to RLink [and read serial number]**.

3.5.3 Reading RLink serial number from any Windows command prompt

1. Execute `RLinkCapab.exe`

Note: This application is installed along with Ride7. You can also download it from the link below:
<ftp://www.raisonance.com/pub/Support/RLink/RLinkCapab.zip>

3.6 Register and activate the software

You can use the software for a year. After this period you can continue to use your RLink with its existing software, but you cannot upgrade or have support until you purchase, register and activate the newest RKit software.

For details, refer to the Ride7 Getting Started document dedicated to the Ride7 support of your CPU family: GettingStartedARM, GettingStartedSTM8, etc.

4. Using RLink

4.1 Before using RLink

Before plugging in and using RLink, you **must**:

- Install Ride7 and the kit corresponding to your target CPU (ARM, STM8, etc.).
- Remove the CD from the drive (see the driver section for more information).

4.2 Using RLink with different target CPUs

Refer to the Getting Started document for your Ride7 kit, which explains how to use RLink (and the other Raisonance tools) with the associated CPU family. Here is a non-exhaustive list:

- *GettingStartedSTM8.pdf*, installed with the RKit-STM8 kit, contains information on using the Raisonance tools, including the RLink, with STM8 and ST7 devices.
- *GettingStartedARM_Ride7.pdf*, installed with the RKit-ARM kit, contains information on using the Raisonance tools, including the RLink, with ARM7, ARM9 and Cortex-M3 devices.

4.3 Using RLink with the host software

Once you have installed Ride7, RLink can be driven by Raisonance's integrated development environment (Ride7) or by other tools such as RFlasher7 and some command-line programming executables.

- **Ride7** drives the RLink and offers seamless control of software development tools (project manager, editor, compiler, assembler, linker, debugger, etc.) from an intuitive graphical interface. It offers full integration of the relevant C/C++ toolsets, project management, code editor and SIMICE instruction set simulator.
Refer to the Ride7 Getting Started document related to your CPU for more information on how to use RLink with Ride7 for this CPU. (GettingStartedARM, GettingStartedSTM8 and so on).
- The **RFlasher7** easy-to-use device programming interface drives RLink and allows users to erase, program, view and verify microcontroller memory. RFlasher7 also includes an automated mode for automatic execution of programming sequences for mass programming and a project mode that allows users to save these programming configurations.
Refer to the Ride7 Getting Started document related to your CPU for more information on how to use RLink with RFlasher for this CPU. (GettingStartedARM, GettingStartedSTM8, ...).
- The **command-line executables**, such as `STM8_pgm.exe`, `Cortex_pgm.exe`, etc., allow you to execute all the standard programming operations, in a way that can easily be integrated into a mass production environment.
They can be called from batch files and any kind of script that is able to start a process. They are installed along with Ride7 and contain their own documentation: just execute them without any argument from a command prompt to see the help.

4.4 RLink connectors

The RLink must be connected to a PC via a USB link which also provides power to the RLink. Before this it must also be connected to a device on an application board via one of three adapters (or directly in case of RLinks 1 and 2). These adapters provide connections which follow the standards defined by the CPU manufacturers (ARM, ST, etc.), as shown here:

Connector	Protocol	Device	Relevant document
20-pin	JTAG	ARM® core-base	ARM Getting started
20-pin (or 10)	JTAG/SWD	ARM Cortex-M3	ARM Getting started
14-pin	JTAG	µPSD	Application Note 48
10-pin	ICC	ST7	STM8/ST7 Getting started
4-pin	SWIM	STM8	STM8/ST7 Getting started

Note: Using a different connector / adapter is possible but not **guaranteed**. If hardware is damaged when doing this, the **warranty** does **not** apply.

4.5 Adapter schematics

The RLink adapters used with the standalone version of RLink3.xx, feature connectors to plug onto the target board. These connectors follow the standards of their respective designers (ARM for JTAG-ARM, ST for ICC-ST7 and SWIM-STM8, etc.). Here are the internet links where you can find the schematics of these adapters:

RLink adapter schematic sPDFs links	
ADP JTAG-µPSD	ftp://www.raisonance.com/pub/forum/RLink_ADPs/ADP_uPSD14.pdf
ADP ICC-ST7 and SWIM-STM8	ftp://www.raisonance.com/pub/forum/RLink_ADPs/ADP_ST7_STM8.pdf
ADP JTAG-SWD (replaces old ADP JTAG-ARM)	ftp://www.raisonance.com/pub/forum/RLink_ADPs/ADP_ARM20.pdf
Older adapter schematics links	
ADP ICC-ST7 v1.0	ftp://www.raisonance.com/pub/forum/RLink_ADPs/ADP_ST7_v1.0.pdf
ADP ICC-ST7 v1.2	ftp://www.raisonance.com/pub/forum/RLink_ADPs/ADP_ST7_v1.2.pdf

This forum FAQ contains more information about these adapters, along with the up-to-date list:
<http://forum.raisonance.com/viewtopic.php?id=2449>

4.6 Connecting and powering

4.6.1 Avoiding electrical conflicts

To avoid electrical problems, potentially leading to hardware damage or personal health injury, it is advised to follow these simple recommendations.

- Ensure the **target** board is **powered either** from RLink (if the ADP allows it) **or** from its own power supply, but **not from both**.
- If the target board is powered from its own supply, make sure that it is properly **isolated** from the mains. Keep in mind that the RLink is connected to ground via USB and the host PC.
- Ensure the voltage is in the **voltage range** supported by your target board, RLink, ADP and protocol.

4.6.2 Standard connecting sequence

This is the simplest and safest procedure to connect and power the PC/RLink/ADP/target system.

Use it if your ADP is not compliant with hardware hot plug, or if you do not want to perform hardware hot plug, which is the case in most situations for programming/debugging/monitoring.

However, this procedure does not allow hardware hot plug (connecting the RLink to the target board while it is running and without resetting it) even using ADPs that are hardware hot plug compliant.

Follow this **order** as much as possible:

1. Plug/unplug the **jumpers** on the **RLink ADP** required for your target device and protocol (or REva board) as/if required. If there are jumpers on your ADP, refer to Ride documentation for your target architecture for details (e.g. GettingStartedSTM8.pdf).
2. Connect the **RLink** to the **RLink ADP** required for your target device and protocol (this step is not required for Primers and REva boards).
3. Connect the **RLink ADP** to the **target board** (be careful how you plug it if the connector does not provide a key) (this step is not required for Primers and REva boards).
4. Power the target board (if it is not powered by the RLink). Be careful to use an **isolated power supply**, as the RLink will be connected to the Ground through the USB and PC.
5. Plug the RLink's **USB** to the PC (after powering-up the PC itself of course).

Notes:

* The order of connecting non-powered connections (steps 1, 2 and 3) is not very important, but it is **critical** to make all **non-power** connections (1, 2, 3) **before** all the **power** connections (4, 5).

* **Unpowering** and **disconnecting** should be performed in the reverse order.

* Using a different (dis)connection order is possible, but is not recommended and **not guaranteed**. If some hardware is **damaged** when doing this, the **warranty** does **not apply**.

* Using a different (dis)connection order can appear to work, whilst it in fact slowly **deteriorates your hardware** and eventually burns it.

* A **reset of the target CPU** might happen at step 5. If this is a problem, to try and avoid it you need to use an ADP that is compliant with hardware hot plug, and use one of the other, more complex, hot plug sequences that are described next.

4.6.3 Hot Plug complete sequence

Use this sequence for debugging/monitoring with hardware hot plug (connecting the RLink to the target board while it is running and without resetting it) when either target board or debugger (or both) is **not grounded**, (e.g. laptop computer, OP4-LAB operating on battery, target board powered by isolated power supply, ...) or when **target CPU reset** is a **critical** problem.

This procedure requires the use of an **ADP** that is compliant with **hardware hot plug**, such as the one (for ARM) shown on the picture at right.

Usually these ADPs contain switches, but it could be jumpers or some other means of connecting the signals in a specific order. The explanations below mention “hot plug switches”, but on some ADPs this could refer to some other similar component. In the same way, the term “configuration jumpers” is used for configuration components on the ADP (protocol selection, power configuration, etc.), but it could be switches or other similar items.



1. Power target board (and turn it ON if applicable).
Target CPU reset happens at this time.
2. Open all hot plug switches on the ADP, or plug/unplug configuration jumpers on the ADP as required, if applicable.
3. Plug ADP to target board. If possible avoid touching any metal part of the system during this phase.
4. Power RLink by PC and USB, or power-up OP4-LAB (for Standalone Monitoring).
5. Wait for USB enumeration or OP4-LAB startup to finish.
6. Wait for at least 15 seconds.
7. Connect the RLink to the ADP. If possible avoid touching any metal part of the system during this phase.
8. Wait for at least 15 seconds.
9. Close ADP Switch 1. If possible avoid touching any metal part of the system during this phase.
10. Wait for at least 30 seconds.
11. Close ADP Switch 2. If possible avoid touching any metal part of the system during this phase.
12. Wait for at least 15 seconds.
13. Close ADP Switch 3. If possible avoid touching any metal part of the system during this phase.
14. Close ADP Switch 4.
15. Close all other ADP Switches (order does not really matter).
16. Start debugging/monitoring (remember to activate software hot plug).

Note:

There is **no guarantee** that a reset will not occur during this procedure.

This is the procedure with the smallest risk that a reset happens, but this risk is **not zero**, especially during steps 3, 7, 9, 11, 13.

If it is really critical to minimize the risk of a reset, **increasing the waits duration** (the longer the better in steps 6, 8, 10, 12), and **grounding** the different system components, using any method you have at your disposal (for example a wire and a resistor) **before** starting the procedure can reduce this risk further.

4.6.4 Hot Plug simplified sequence

Use this sequence for debugging/monitoring with hardware hot plug, when using **previously grounded** target board AND debugger, and when avoiding a **reset** of the CPU is **not critical**.

This procedure requires the use of an **ADP** that is compliant with **hardware hot plug**.

1. Power target board (and turn it ON if applicable). Target CPU reset happens at this time.
2. Open all hot plug switches on the ADP.
Plug/unplug configuration jumpers on the ADP as required, if applicable.
3. Connect ADP to RLink.
4. Connect PC to RLink via USB (assuming PC is powered, turned ON and Windows started).
5. Connect the ADP to the target board.
6. Close ADP hot plug switches concerning grounds, powers and reset (usually switches 1, 2, 3, 4) in the correct order. **Wait at least 1 second** after every switch closes.
7. Close all remaining ADP switches.
8. Start debugging/monitoring (activate software hot plug).

Note:

If the target board OR host PC is not grounded, then a target CPU reset and other problems including hardware damage will probably happen on the target board during step 5 due to electrostatic discharges (ESD). Even with a grounded board and host, there is no guarantee that a target CPU reset will not occur during this procedure, although this risk is quite low.

If it is critical to avoid a reset, use the complete procedure instead, which provides better, safer grounding and reduces further the risk of reset. With grounded parts the risk of hardware damage is very low but still not null.

4.7 Operating conditions

The RLink is designed to operate in "normal laboratory conditions". This means, amongst other things:

- Respect the **voltage** ranges, **power** scheme (ensure isolation from mains, avoid conflict between several power supplies, etc.), **order** of connecting and powering, etc. mentioned earlier in this document (which depends on the version of RLink, target protocol, etc.)
- Use it only with official, licensed (or demo) **software** from IoTize or its partners.
- Use it in **reasonable conditions** of temperature, pressure, electro-magnetic environment, etc.
- Avoid exposing it to strong or persistent **mechanical or electromechanical stress** (shocks, vibrations, heat, pressure, microwaves, magnets, X-Ray, etc.) even when not using it.
- Do not **open** it (unless our support team tells you to).
- Do not try to reverse-engineer it in any way, including software.
- This list is **not** exhaustive. Use your common sense to avoid problems.



Failure to meet these requirements will make you lose the **warranty** on the product. Even if you meet these requirements, IoTize and its partners **cannot** be held **responsible** of any damage or loss occurring from the use of RLink and its associated products (ADPs, Primers, REva/OP4 motherboards and daughterboards, etc.).

4.8 Support and FAQ

The IoTize and Raisonance web sites contain a lot of useful information, documents and FAQs:

- <http://support.raisonance.com>
- <http://forum.raisonance.com>
- <http://support.raisonance.com>

4.9 Checklist of items to prepare before contacting support

Here is a list of questions that we will ask whenever you request any support from IoTize for RLink. If you plan to contact us by email, telephone or on the forum, please include all this information in your message (or have it ready if calling on the phone).

For an easier, faster and more efficient support, the best is to copy the text from this form into an email <http://forum.raisonance.com/viewtopic.php?pid=7573#p7573>, answer each question in the line below the question, and then send it to support@raisonance.com.

The rest of this section lists an obsolete list of questions to give you an idea of the information needed.

1 Questions on the Raisonance tools

- 1.1 What is your customer serial number, if you have one? (open Ride7, click **Help>About** and look for Serial Number, or send us a screen shot).
- 1.2 What is the version of your software? (Ride7)
 - 1.2.1 If in Ride6, what is the build number of your Ride6? (open Ride6, click **Help>About** and look for BN, or send us a screen shot)
 - 1.2.2 If in Ride7, what are the versions of your Ride7 kits? (open Ride7, click **Help>About** and look for Installed Products, or send us a screen shot)
 - 1.2.3 If you are using other software (STVD, CAPS, ...), we will need the name(s) and version number(s).
- 1.3 What kind(s) of RLink(s) do you have? (REva? RLink-Std? RLink-STX?)

2 Questions on the RLink

- 2.1 Does the PWR LED turn ON when you connect RLink to the PC?
- 2.2 Is the USB driver correctly installed?
 - 2.2.1 Does the BUSY LED turn ON and then OFF when you connect RLink to the PC?
 - 2.2.2 Does RLink appear under the Jungo section in Windows Device Manager when it is plugged in?
 - 2.2.3 What is your host system (CPU, version of Windows, service pack(s), etc.)?
 - 2.2.4 Can you read the RLink serial number? (open RFlasher, select any ST7 or ARM device as target and RLink as programmer, click **Connect to RLink**)
Note: if you think that the USB driver is not installed, reinstall Ride7 or run *RLinkUSBInstall.exe* which is available on the Ride7 CDs and in the Ride7 installs.
- 2.3 What is your RLink(s) serial number? (see 2.2.4. above. the SN is not written on the RLink itself).

3 Questions on the target

- 3.1 What is your target CPU (we need the complete name of the derivative)?
- 3.2 What is your target board(s)?
 - 3.2.1 Please send us the schematic of the board.
 - 3.2.2 If it is a commercial board, who is its manufacturer and reference? (+ did you modify it?)
 - 3.2.3 What is the configuration of the jumpers and switches on the board, if any?
 - 3.2.4 How do you power the board?
 - 3.2.5 If we need to get the real board for testing, under what conditions would it be possible (sign a NDA, return it after the tests, ...)?

4 Questions on the problem itself.

- 4.1 Please describe the problem you are experiencing as precisely as you can.
 - 4.1.1 What software are you using (Ride7, RFlasher, CAPS, STVP7, ...)?
 - 4.1.2 What is the simplest sequence of operation that we can perform in order to trigger the problem?
 - 4.1.3 How does the problem manifest itself (please write down all error messages, if any, or make screen shots)?
 - 4.1.4 Did the problem appear in a configuration that was working before?
 - 4.1.5 If the answer to the previous question was "yes", can you think of anything that changed between the last time it succeeded and the first time it failed?
- 4.2 Do you have any comment or personal interpretation of the problem?
- 4.3 Please do a few tests.
 - 4.3.1 If you have other software, (from ST, Signum, ...) does it all show the same problem? (please tell us what you could test and how it went).
 - 4.3.2 If you have other programmers/debuggers, even other RLinks, do they all show the same problem? (please tell us what you could test and how it went).
 - 4.3.3 If you have other boards/CPUs, (demo boards, ...) even other boards identical to the one showing the problem, do they all show the same problem? (please tell us what you could test and how it went).
 - 4.3.4 If the problem appears with a particular application, can you send us your project?
Note: If we need it, we will need the complete project folder, including source, include, object, hex and listing files. In some specific cases, the hex file might be enough, but we cannot guarantee that. We can sign NDAs if this information is confidential. We can work with a reduced version of the application, if it shows the same problem, but we need a complete project that we can recompile at IoTize.
 - 4.3.5 If you only have one application, please try to reproduce the problem with one of the examples provided with Ride7, or try to reduce your application until there is only the minimum remaining to show the problem.

5 Can we share the information in the answers to this form with our partners?

Note: This is NOT for advertising purposes. Depending on the problem you have, our partners (mostly, but not only, the chip manufacturers like ST) might have more technical information than us on the cause of your problem and be able to answer faster and more accurately, or just to help us a little.

Thank you for taking the time to answer. You will see that it has not been wasted, because it allows us to provide you with fast and efficient support.

5. Conformity



ROHS Compliance (Restriction of Hazardous Substances)

IoTize products are certified to comply with the European Union RoHS Directive (2002/95/EC) which restricts the use of six hazardous chemicals in its products for the protection of human health and the environment.

The restricted substances are as follows: lead, mercury, cadmium, hexavalent chromium, polybrominated biphenyls (PBB), and polybrominated diphenyl ethers (PBDE).



CE Compliance (Conformité Européenne)

IoTize products are certified to comply with the European Union CE Directive.

In a domestic environment, the user is responsible for taking protective measures from possible radio interference the products may cause.



FCC Compliance (Federal Communications Commission)

IoTize products are certified as Class A products in compliance with the American FCC requirements. In a domestic environment, the user is responsible for taking protective measures from possible radio interference the products may cause.



WEEE Compliance (The Waste Electrical & Electronic Equipment Directive)

IoTize disposes of its electrical equipment according to the WEEE Directive (2002/96/EC).

Upon request, IoTize can recycle customer's redundant products.

For more information on conformity and recycling, please visit the IoTize website www.iotize.com

6. Glossary

Term	Description
CScript	Microsoft Windows' script tool
RScript	Script language

7. Index

Additional help or information.....	4	rlink_winusb.....	8
CE.....	20	RLinkCapab.....	7, 8
Compliance.....	20	RLinkDRV.....	8
Conformity.....	20	RLinkUSBInstall.exe.....	7
Directive.....	20	ROHS.....	20
DriverStore.....	7	Scope of this manual.....	4
DRVSTORE.....	7	Vista32.....	7
FCC.....	20	Vista64.....	7
Found new hardware.....	7	WEEE.....	20
incorrect files.....	7	WinDriver.....	7
Jungo.....	7	windr6.....	8
Jungo_WinDriver_2000_NT_XP.....	8	WinUSB.....	7
Lead.....	20	XP.....	8
MS_WinUSB_XP_Vista.....	8	XP32.....	7
Purpose of this manual.....	4	XP64.....	7

8. History

Date	Description
24 Feb 09	Initial version. V0.2
16 Jun 09	Add alternative recovery procedure for USB driver installation. V1.0
23 Apr 12	Add information about RLink4 and operating conditions.
10 Jul 12	Modified cover page, final page and section 1.3 Additional help or information for IoTize
28 May 13	Modified template
21 June 13	Added information about Hot Plug ADPs and connection sequences.



Disclaimer

Information in this document is subject to change without notice and does not represent a commitment on the part of the manufacturer. The software described in this document is provided under license and may only be used or copied in accordance with the terms of the agreement. It is illegal to copy the software onto any medium, except as specifically allowed in the license or nondisclosure agreement.

No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the purchaser's personal use, without prior written permission.

Every effort has been made to ensure the accuracy of this manual and to give appropriate credit to persons, companies and trademarks referenced herein.

This manual exists both in paper and electronic form (pdf).

Please check the printed version against the .pdf installed on the computer in the installation directory of the most recent version of the software, for the most up-to-date version.

The examples of code used in this document are for illustration purposes only and accuracy is not guaranteed. Please check the code before use.

Copyright © IoTize 2017 All rights reserved