RAISONANCE

a KEOLABS brand

RLink & REva for SX8810

Raisonance Tools for C816 family

Getting Started

Document version
7 May 2013

Contents

1. Introduction

This document describes how to debug an SX8810 CPU using an RLink-SX or an SX8810 REva board together with Raisonance's Ride7 and RKit-C816 (hereafter referred to as **Ride7-C816**).

It assumes that you have already read and understood the C816 Getting Started for Ride7, that you know how to create and use projects for making applications, and how to use the Ride7 debugger. It does not repeat information from these other documents. You can find the C816 Getting Started document by clicking in Ride7: **> Help > View documentation** under **C816\Ride7 for C816.**

1.1 Purpose of this manual

This guide can be used by anyone interested in programming and debugging SX8810 targets using Ride7-C816.

1.2 Scope of this manual

This document describes how to get started using Ride7-C816 to compile and debug your application or one of the included sample applications. It assumes that you have the prerequisite knowledge of C and C816 assembly languages.

1.1 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 KEOLABS' 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/index.php
Support Email:	support@raisonance.com

For information and support about SX8810 chips, contact Semtech: http://www.semtech.com/

1.2 Raisonance brand microcontroller application development tools

January 1, 2012, Raisonance became the brand under which the company KEOLABS 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 KEOLABS 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. Presentation of the tools

Ride7-C816 works with the following tools from Raisonance or Semtech:

- **Ride7**: PC software, designed by Raisonance, that you can use to make applications for the SX8810, and also to program and debug them with RLink.
- **SX8810 simulator**: Part of Ride7 that simulates the CPU core, memories and some peripherals.
- **SX8810 debugger**: Graphical interface for debugging SX8810 applications using the simulator or an RLink connected to an SX8810 target board.
- **RLink-SX (Semtech part number: SX8810EVKB)**: USB-to-JTAG dongle for programming and debugging SX8810 CPUs in-situ.
- **SX8810 REva board (Semtech part number: SX8810EVKA)**: Development board including RLink functionality, demonstration hardware and a daughterboard featuring an SX8810 CPU.
- **SX8810 programming board (Semtech part number: SX8810EVKBDB)**: Programming board for production or prototyping of small applications.

See the GettingStartedC816_Ride7.pdf documentation for information about Ride7 and the debugging interface (software and hardware).

2.1 Ride7

Ride7 is the PC software, designed by Raisonance, that you can use to make applications for the SX8810, and also to program and debug them using RLink. Please read the Getting Started C816 document, *GettingStartedC816_Ride7.pdf*, before continuing.

Debugging is limited to 4K instructions if you have the demonstration version of Ride7-C816. It is not limited if you have the PRO version of Ride7-C816. This limitation applies to both software simulation and hardware debugging using RLink.

2.2 SX8810 debugger

This is the graphical user interface for debugging SX8810 applications using the simulator or an RLink connected to an SX8810 target board. See Chapter 3 of this doc for detailed usage instructions. The same user interface is used for all debuggers, including the simulator and the RLink hardware debugger.

See *GettingStartedC816_Ride7.pdf* for instructions on using the debugger.

2.3 SX8810 simulator

This software simulates the CPU, including the C816 core, the entire memory space and some peripherals. Complex peripherals and less common peripherals are not simulated.

See GettingStartedC816_Ride7.pdf for instructions on using the simulator.

2.4 RLink-SX

The RLink-SX hardware dongle, designed by Raisonance, programs and debugs the target SX8810 chip from a PC. Technically, it is a bridge between a PC USB and many serial protocols. The JTAG + SX8810 debug protocol used by SX8810 devices is one of the protocols supported by an RLink-SX.

Only the RLink-SX model can use the JTAG protocol for debugging SX8810 targets.

- RLink-SX can program without limitation.
- Debugging is not limited by the RLink-SX itself, but could be limited by Ride7.
- Other RLinks cannot use the JTAG protocol for debugging SX8810 targets.

The RLink SX8810 adaptor features the standard JTAG connector defined by Semtech, which allows communication with the SX8810.

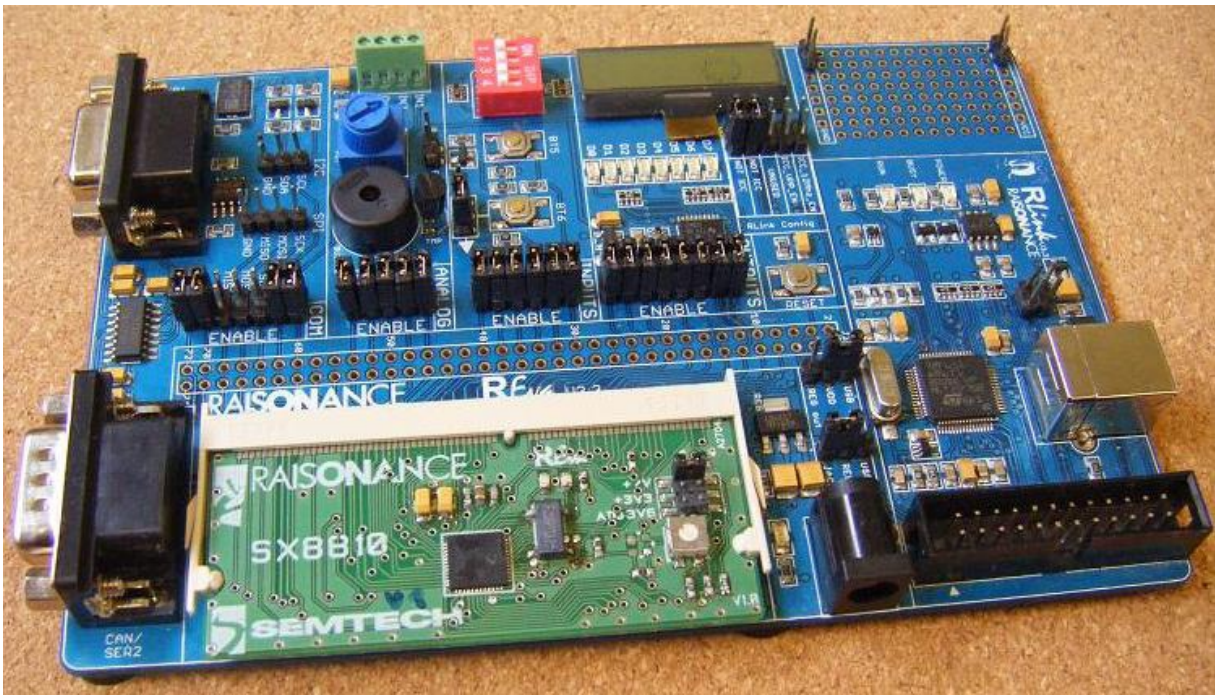
The RLink-SX can be used with commercial demonstration boards like the REva, and also with any custom board featuring an SX8810 chip and a JTAG connector.

> You must install Ride7, which installs the RLink USB driver, before you connect an RLink to your PC.

Picture of RLink-SX with SX8810 ADP:

2.5 SX8810 REva board

The SX8810 REva board consists of an SX8810 daughterboard plugged on a REva motherboard.
Picture of REva board with SX8810 daughterboard:



- The REva motherboard (blue in the above picture) is a generic demonstration board that provides most of the standard features required of a demonstration board. It includes an RLink for debugging and programming the SX8810 device on the daughterboard. The whole board can be powered by the USB of the RLink, or by the Jack connector. Please refer to the REva documentation for more information. You should find it in <*Ride*>*Doc**C816*\.
- The REva daughterboard (Semtech part number: SX8810EVKADB; green in the above picture) includes the target SX8810 device to be programmed and debugged. It is plugged into the SO-DIMM connector of the REva board. Please refer to the daughterboard's schematics for more information. You should find them in <*Ride*>*Doc**C816*\.

Note: The embedded RLink and some REva peripherals do not support voltages lower than 2.5V. Therefore, while using the REva board, you cannot set the power lower than 2.5V. To do this you should use your own board and a stand-alone RLink with an SX8810 ADP.

2.5.1 Power configuration

The power of the board is configured using jumpers on both the motherboard and the daughterboard. See the motherboard documentation for the three possible ways to configure it:

- Direct power from USB
- Power from USB through regulator
- Power from external Jack through regulator

If the motherboard is configured for one of the two modes that use the regulator, then the voltage output by the regulator is configured by the jumpers on the daughterboard:

- “+2V”: configures the regulator for a 2V output.
- “+3V3”: configures the regulator for a 3V3 output.
- “ADJ3V6”: configures the regulator for a voltage that depends on the state of the potentiometer, between 0V and 3V6.

2.5.2 Inventory of connections between SX8810 CPU and REva board components

This table lists the connections between the evaluation part of the motherboard and the SX8810 CPU on the daughterboard. You can use it to design your application's firmware. See the board schematics and CPU datasheet for more details.

Signal name	REva board pin(s)	SX8810 pin(s)	Comments
GND	J2.1 & others	GND	General ground
VCC3	J2.2	VDD & VDDA	SX8810 CPU power
REG_OUT	J3.1 J3.2 J3.3 J4.1	-	Used for motherboard power configuration. See motherboard documentation.
REG_ADJ	J3.4 J4.3	-	Used for motherboard power configuration. See motherboard documentation.
RST_HARD	J2.4	-	Input of STM6315 reset manager, controlled by push-button on motherboard.
TDO	J2.5	PB4	JTAG TDO. Used by RLink.
TCK	J2.9	PB6	JTAG TCK. Used by RLink.
TMS	J2.11	TMS	JTAG TMS. Used by RLink.
nSRST	J2.13	NRESET	SX8810 reset input. Can be tied low by RLink or STM6315 (when pressing push-button on MB).
TDI	J2.17	PB5	JTAG TDI. Used by RLink.
RTRST	J2.21	PB7	JTAG TRST. Used by RLink.
LED7	J2.10	PA7	LED
LED6	J2.12	PA6	LED
LED5	J2.14	PA5	LED
LED4	J2.16	PA4	LED
LED3_SPI_NSS	J2.18 & J2.65	PA3	Use either for LED or for SPI.
LED2_SPI_SCL	J2.20 & J2.64	PA2	Use either for LED or for SPI.
LED1_SPI_MOSI	J2.22 & J2.66	PA1	Use either for LED or for SPI.
LED0_SPI_MISO	J2.24 & J2.68	PA0	Use either for LED or for SPI.
BT6	J2.30	PB2	Push button
BT5	J2.32	PB1	Push button
SW4	J2.34	PC5	Switch
SW3	J2.36	PC4	Switch
SW2	J2.38	PC1	Switch

Signal name	REva board pin(s)	SX8810 pin(s)	Comments
SW1	J2.40	PC0	Switch
AN_1	J2.46	AC_A1	Analog Input
AN_2	J2.48	AC_A2	Analog Input
PWM	J2.50	PB0	Filtered to Analog Output
POT	J2.52	AC_A0	Analog Input from potentiometer
TEMP	J2.54	AC_A3	Analog Input from temperature sensor
I2C-SCL	J2.60	PC3	I2C clock
I2C-SDA	J2.62	PC2	I2C data
LED2_SPI_SCL	J2.20 & J2.64	PA2	Use either for LED or for SPI.
LED1_SPI_MOSI	J2.22 & J2.66	PA1	Use either for LED or for SPI.
LED0_SPI_MISO	J2.24 & J2.68	PA0	Use either for LED or for SPI.
UART-TX	J2.70	PC6	UART Transmit line
UART-RX	J2.72	PC7	UART Receive line
PB3	J2.29	PB3	No dedicated feature.
AC_R0	J2.33	AC_R0	No dedicated feature.
AC_R1	J2.35	AC_R1	No dedicated feature.
AC_R2	J2.37	AC_R2	No dedicated feature.
AC_R3	J2.39	AC_R3	No dedicated feature.
AN_4	J2.45	AC_4	No dedicated feature.
AN_5	J2.47	AC_5	No dedicated feature.
AN_6	J2.49	AC_6	No dedicated feature.
AN_7	J2.51	AC_7	No dedicated feature.
AN_8	J2.53	AC_8	No dedicated feature.
AN_9	J2.55	AC_9	No dedicated feature.
LED3_SPI_NSS	J2.18 & J2.65	PA3	Use either for LED or for SPI.

2.6 SX8810 programming board

This small board, initially designed for production programming, can also be used for prototyping small applications.

It includes a connector for a stand-alone RLink, a socket for an SX8810 device, and a wrapping zone where you can add your own hardware for testing on a production line or for prototyping.

This board is designed and maintained by Semtech. Please contact them directly for more information, datasheets, schematics, etc. The board's schematic is also available in *<Ride>\Doc\C816*.

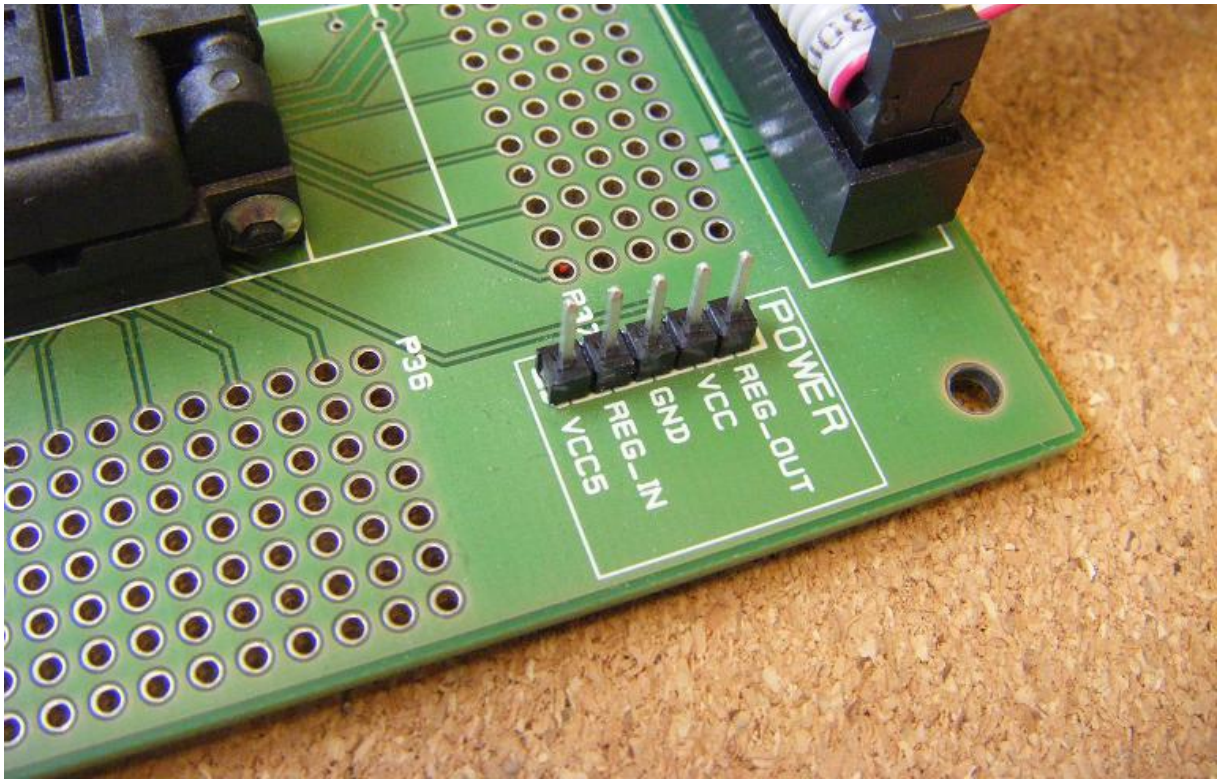
On the software (PC) side, using this board for programming and debugging is the same as using a REva board or your own board with an RLink connected to it. See Chapter 3 of this document for instructions on how to program and debug the target SX8810 from the software (PC) side when these connections are made (usually with *SX8810_pgm.exe*).

The rest of this section contains board-specific power details.

SX8810 programming board with an RLink-SX connected to it:

2.6.1 Power section of the programming board

Power section with no jumpers:



- **REG_OUT**: Output of the programming board's 3V3 regulator.
- **VCC**: Input power of the SX8810 CPU. It also provides power to the RLink I/Os.
- **GND**: Voltage reference (0V) for the whole board, SX8810 CPU, regulator and RLink connector.
- **REG_IN**: Input of the programming board's 3V3 regulator.
- **VCC5**: 5V input from the RLink connector.

WARNING: When you connect the RLink to the board, the GND signal of the board is:
- connected to the ground of the RLink, which is
- connected to the GND of the USB, which is
- connected to the GND of the host PC, which is
- connected to the GND of the building. If you use one of the modes in which power comes from an external source, you must make sure that this power is correctly isolated from the mains.

Physical damage to the board, RLink, host PC and your personal health can occur if you do not follow this advice. This also applies when using your own board.

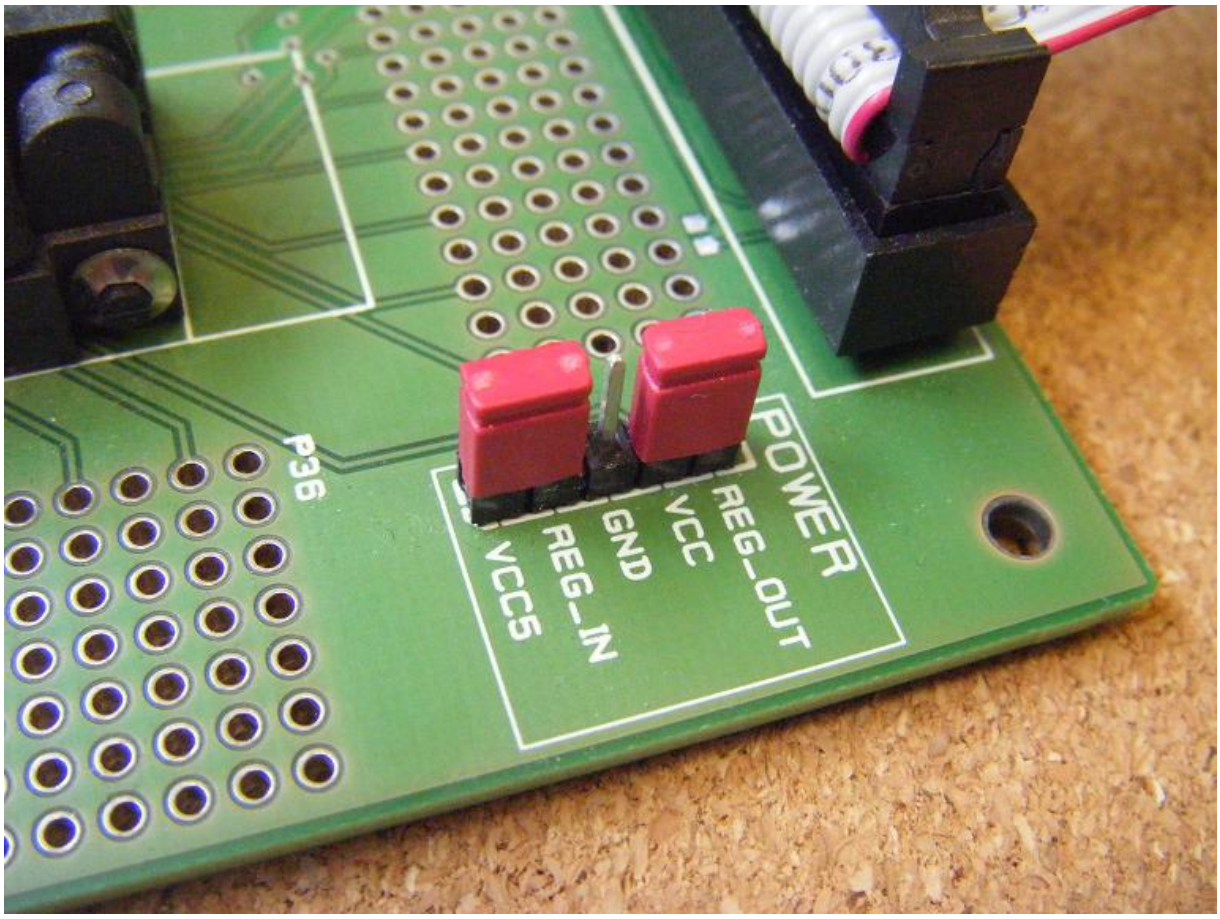
The next subsections detail the three valid ways the programming board can be powered:

- Power from RLink through regulator (default)
- Power from outside source through regulator
- Power from outside source directly to SX8810

2.6.2 Power from RLink through regulator

This is the default mode. In this mode, the RLink provides the 5V power from the USB of the host PC to power the regulator of the programming board, which then powers the target SX8810 CPU.

SX8810 programming board with jumpers for power from RLink through regulator:



In this mode, the following voltages are applied:

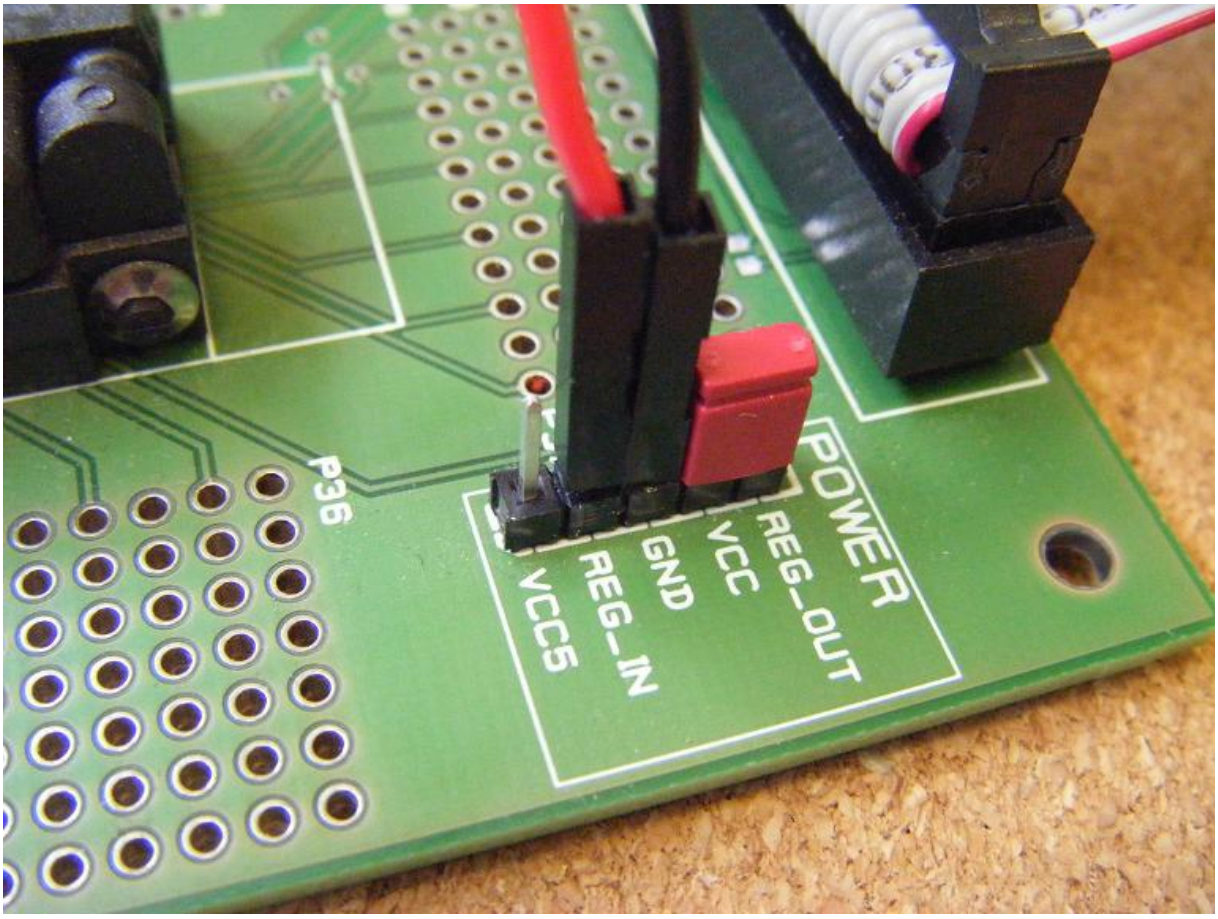
- GND = 0V
- REG_IN = VCC5 = 4V8 (if board, RLink, USB and PC are connected)
- VCC = REG_OUT = 3V3

2.6.3 Power from outside source through regulator

In this mode, an external source powers the regulator of the programming board, which then powers the target SX8810 CPU.

This mode can be used to execute an application without the presence of an RLink (after soldering an LED or other test hardware).

SX8810 programming board with jumpers and wires for power from external source through regulator:



In this mode, the following voltages are applied:

- GND = 0V
- *(unused)* VCC5 = 4V8 (if board, RLink, USB and PC are connected)
- REG_IN = External_Power (should be around 5V)
- VCC = REG_OUT = 3V3

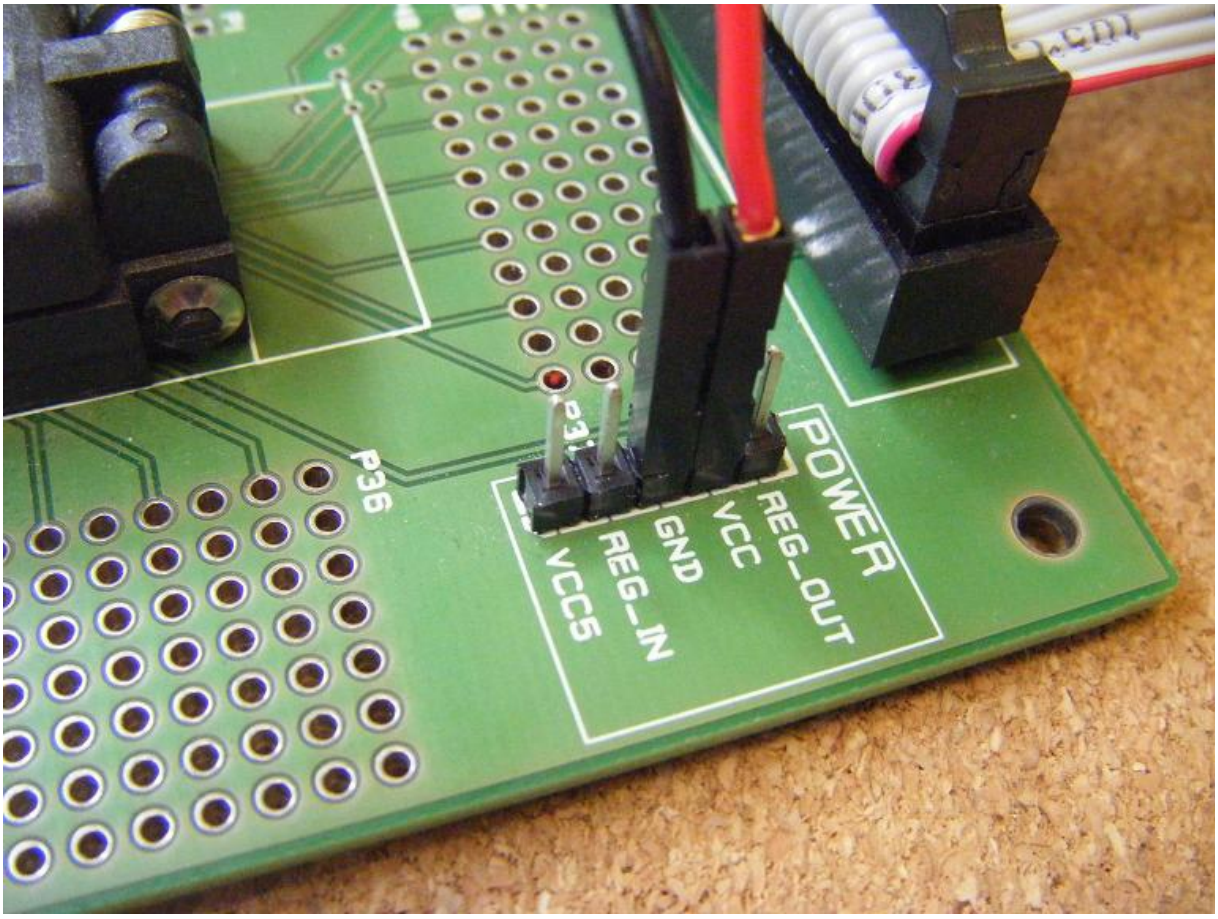
2.6.4 Power from outside source directly to SX8810

In this mode, an external source is used to power the target SX8810 CPU directly.

This mode can also be used to execute an application without the presence of an RLink (after soldering an LED or other test hardware).

In this mode the regulator is not used at all. If you suspect that your board's regulator might be burned, try connecting the board in this mode to check if the problem really comes from the regulator.

SX8810 programming board with wires for direct power from external source:



In this mode, the following voltages are applied:

- GND = 0V
- *(unused)* VCC5 = 4V8 (if board, RLink, USB and PC are connected)
- *(unused)* REG_IN = floating
- *(unused)* REG_OUT = 0V
- VCC = External_Power (see SX8810 documentation from Semtech for valid voltage range)

3. Debugging an example project

This part of the document shows the features of the monitor and their use through a quick tutorial. We will debug one of the examples for the REva board provided with Ride7.

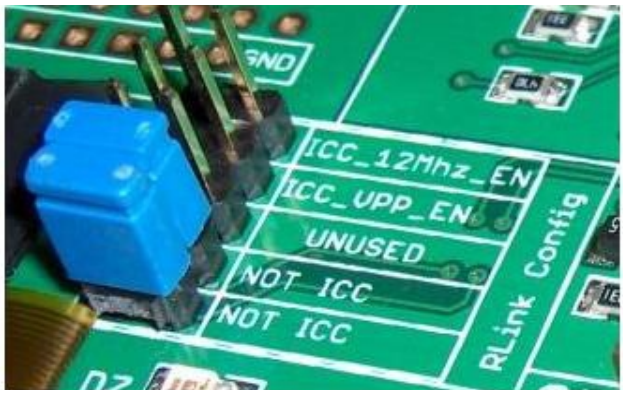
3.1 Connecting the hardware

Doing all these checks seriously is very important! It is the best way to make sure that your whole system is working and is correctly configured, powered and connected (RLink, USB driver, target CPU, target board, connections, etc.).

1. Install the Ride7 and RKit-C816.
2. Remove all CDs from all drives, including the installation CD of Ride7.

!!! WARNING !!! Remove all CDs, from all drives. **Especially the installation CD of Ride7.** Some versions of Windows search CDs for drivers without asking you, so they may take the wrong RLink USB driver version (for another Windows version). Make sure you use the driver that was registered in Windows by the Ride7 installation program.

3. Connect your stand-alone RLink or REva-embedded RLink to your PC using the USB cable. When using a REva starter kit's embedded RLink with an SX8810, ensure that your jumpers are set correctly on the RLink for JTAG communication: Plug two jumpers in the "NOT_ICC" section. REva jumpers positioned for JTAG communication protocol:



4. Check that the POWER and BUSY LEDs of the RLink are lit.
5. Follow the driver installation process by Windows.

Note: The first time you plug a given RLink on a given PC, loading the driver may take a while. This driver was installed by the Ride7 installation program, so let Windows search for it automatically, in your Ride7 installation directory and Windows system directories, but not on the web. The next time you plug in the RLink, it will be much faster.

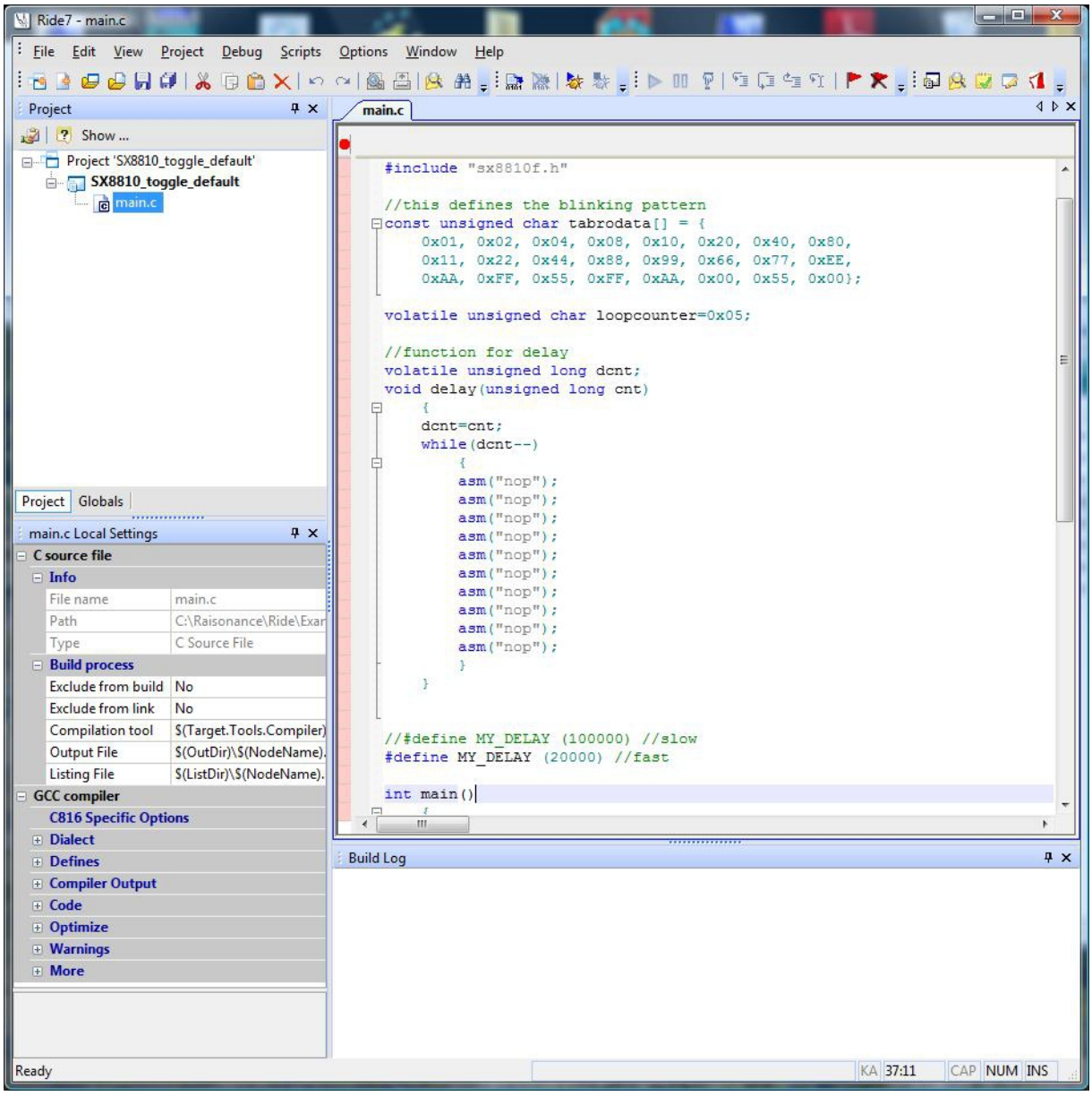
6. After the driver is loaded and Windows has enumerated the RLink correctly, the BUSY LED of the RLink should turn OFF. Check this before continuing.
7. If you are using a stand-alone RLink, connect it to the target board, and power the target board.
8. Check that the POWER LED is lit on the target board (for the REva board, it is located between the Jack and the daughterboard).

Note: You might need to change the jumpers to select Jack or USB power depending on the power source you choose to use. See the REva documentation for more information.

From this point, you should not need to physically touch the board until the end of the tutorial.

3.2 Opening the example project

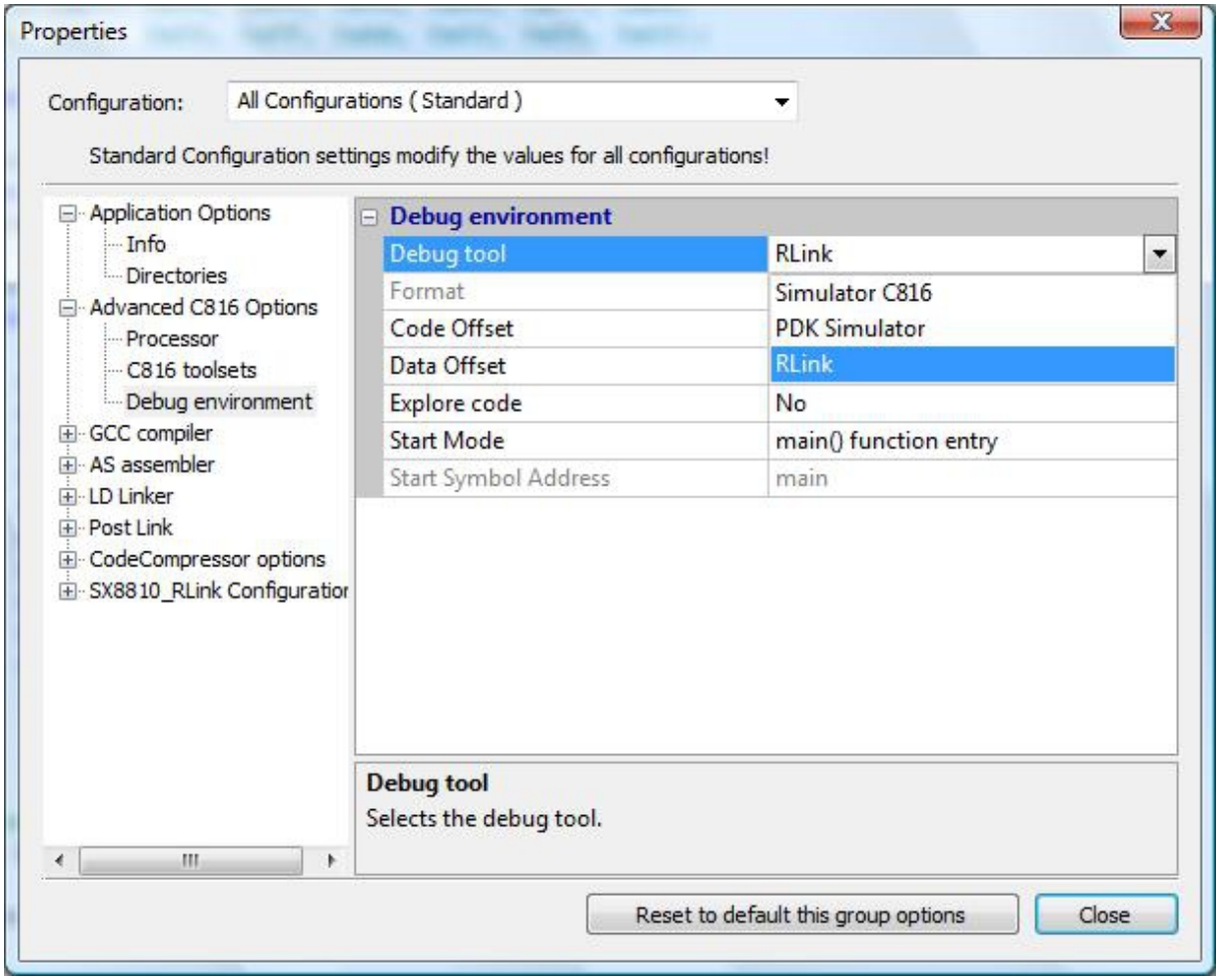
1. Open Ride7 using Windows: **Start > Programs > Raisonance Tools > Ride7 > Ride7**.
2. Open the project, using **Project > Open Project** in Ride7 and select file *<Ride>\Examples\C816\SX8810\SX8810_toggle_default\SX8810_toggle_default.rprj*
3. Now the project is open and ready to be used.



3.3 Setting the debug options

Before debugging, check that the project options are correctly set.

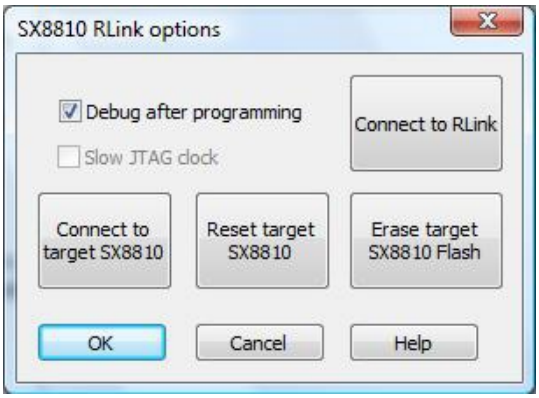
Open the debug options window, in which you can tell Ride7 to use the RLink instead of the software simulator for debugging (**Options > Project Properties > Advanced C816 Options > Debug environment**).



3.4 Testing the connections

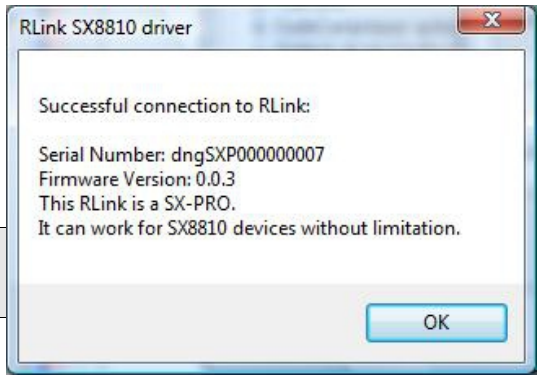
- When **SX8810-Monitor** is selected you can select **SX8810_RLink Configuration** and click **Click here to open options dialogue box** to open the **SX8810 RLink options** box.

  This window allows you to test and configure the debugger, the target SX8810 and their connections.
  It also allows you to erase the Flash.



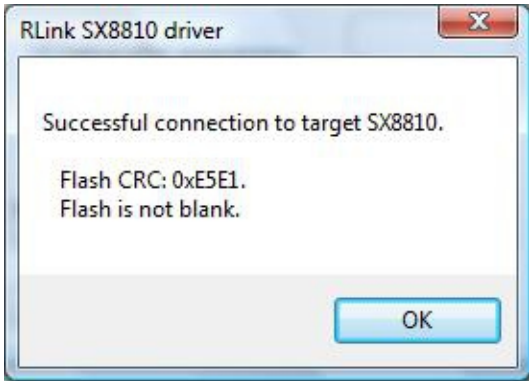
- Click **Connect to RLink** to check the connection between the PC and the RLink.

- If successful, this displays the RLink Serial Number, which Raisonance needs for any support request (unless of course, reading the Serial Number fails).
  This also says whether your RLink can be used for SX8810 devices or not.



Note: The RLink Serial Number is NOT written on the RLink. The only way to find it is with this test.

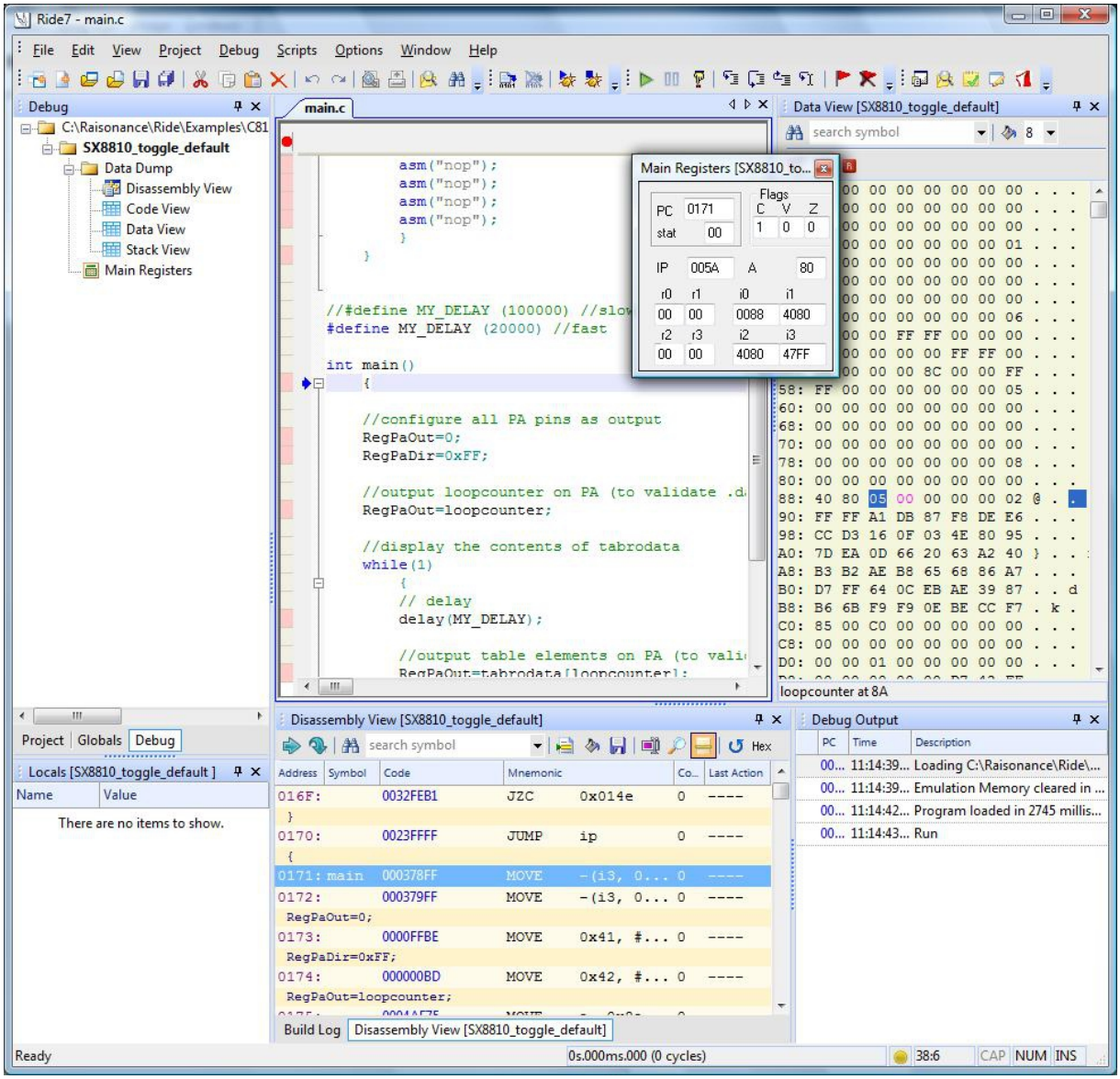
- Click **Connect to target SX8810** to check the connection between the RLink and the SX8810, and the SX8810 power.
- If successful, this also displays information about the target SX8810, like the CRC of the program currently loaded in the Flash, and whether or not this CRC corresponds to a blank Flash. If the CRC is different from the CRC expected for a blank Flash, then you can be sure that the Flash is not blank. The reported CRC is computed by the SX8810 on the full content of the Flash.
  On rare occasions software may have the same signature as an erased Flash.



If you get any errors while doing the checks above, you must understand and solve them before going on, because the next steps cannot work if these checks fail. The error messages are usually clear enough for you to understand and solve any problem. If not, please contact our support team.

3.5 Starting the debug session

To start the debug session, select **Debug > Start**.
Your application will be loaded, run and stopped on `main()` function.
The Ride7 environment should look similar to the one shown below:



Note: From a user interface point of view, basic debugging commands (stopping and resuming CPU execution, setting a breakpoint, single-stepping, checking memory and registers, etc.) are identical, whether you are using the simulator or a hardware debug tool.
You must refer to the *Getting Started with Ride7* document to get familiar with the simulator before starting to work with the hardware debuggers.

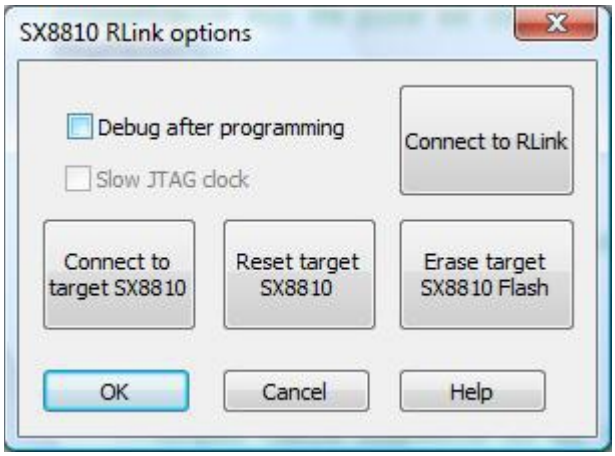
3.6 Other examples

Ride7 is installed with at least one other example that deserves your attention. It can be found in *<Ride>\C816\SX8810\SX8810_toggle_custom*.

The toggle_custom example is similar to toggle_default, but it also shows how to use custom startup and linker script files for the SX8810. Use it as a starting point for your applications if you need to modify one of these template files. See the *Getting Started with Ride7* document for more details about startup and linker script files, when and how to use a custom version of them.

3.7 Programming without debugging

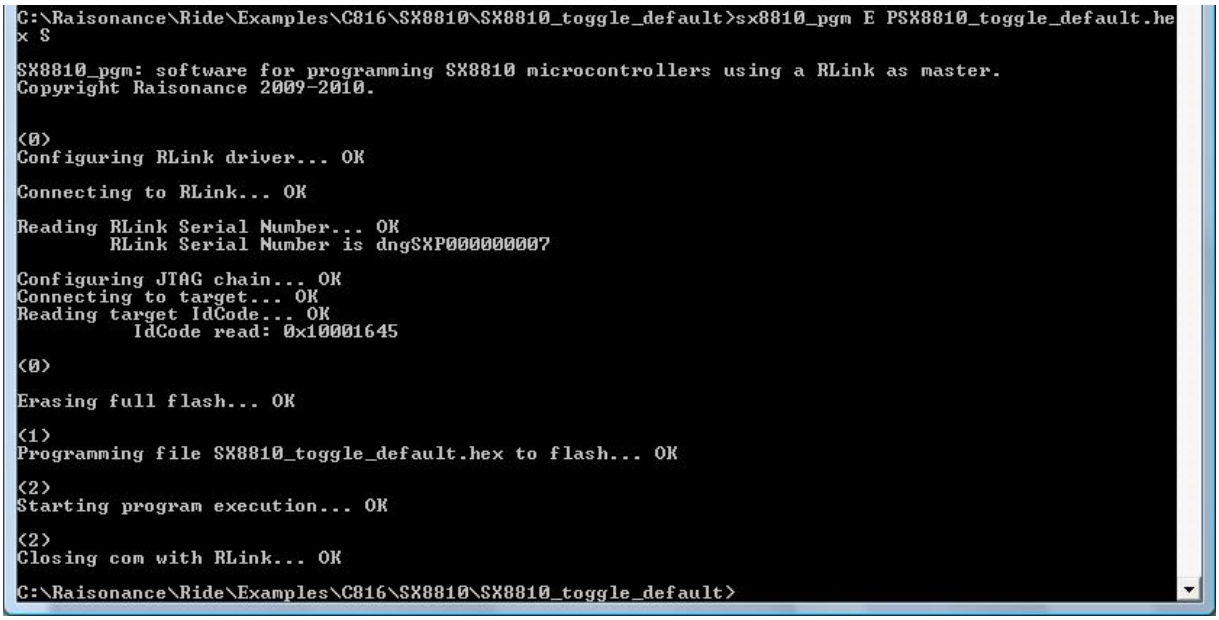
It is sometimes preferable, instead of debugging, to simply program the Flash and let the application run. To do this, terminate the debug session, go back to the advanced debug options and uncheck the **Debug after programming** checkbox.



3.8 Using SX8810_pgm.exe

Once your application is designed, debugged and validated, you will probably need to produce it in high volume. This means that you will want to program a lot of SX8810 devices with the same code very quickly. If the operator is a subcontractor from another company, you might not want him to have your source code. For these situations, Raisonance provides an executable called *SX8810_pgm.exe* that allows you to directly program hex files (generated by Ride7 during the link of applications) to the SX8810 devices without using Ride7. It can also easily be called by higher-end production and test programs, or just a batch file, in order to avoid mistakes from the production operator.

- Open a command prompt using Windows **Start** menu
- Use CD command to go to the example project's folder.
- Execute `SX8810_pgm`, to see the most recent Help for using it.
- Execute `SX8810_pgm E PSX8810_toggle_default.hex S`. This erases the SX8810 Flash, programs it with the example's code from the hex file and starts its execution.



4. Designing your own application

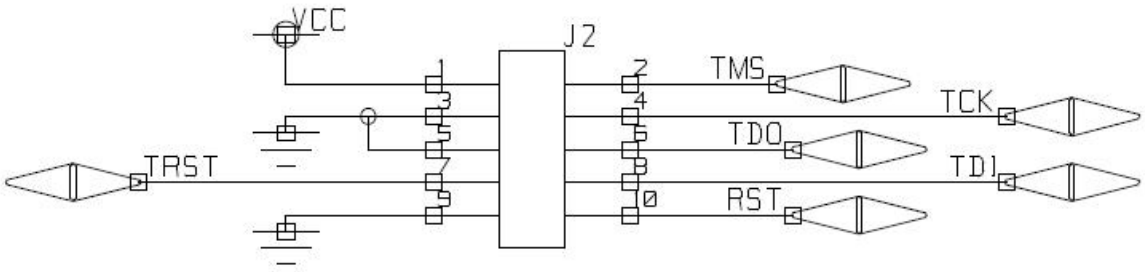
Using the debugger with your own application board is exactly the same procedure as described above for debugging an example. Except, you must include the JTAG connector on your board, as described in the next section. See also the Limitations chapter.

The programming and REva board schematics can serve as starting point for your board's design.

4.1 Including the JTAG connector on the board

When you design your system's PCB, you must take care that it includes a JTAG connector, and that this connector is correctly connected to 8 signals; the SX8810 (6 signals), ground and power.

The lines must be short to avoid disturbances. We cannot guarantee that the RLink and SX8810 will be able to communicate using JTAG if the lines are longer than 10cm between the JTAG ADP and the SX8810 CPU.



4.2 Power Warning

Some care must be taken in the design of the power part of your board, especially if your board is designed to be powered from mains:

WARNING: When you connect the RLink to the board, the GND signal of the board is:
- connected to the ground of the RLink, which is
- connected to the GND of the USB, which is
- connected to the GND of the host PC, which is
- connected to the GND of the building.

When using your own target board, you must make sure that the board's power is correctly isolated from the mains. Physical damage to the board, RLink, host PC and your personal health can occur if you do not follow this advice.

5. Limitations

5.1 Use of JTAG signals

The RLink communicates with the SX8810 using JTAG and NRESET signals.

Since JTAG functionality takes priority over the pins primary function, pins PB4 to PB7 are not available for the application during the debug session.

Additionally, pins NRESET and TMS, as well as pins PB4 to PB7, must not be driven by any external hardware during debugging or programming since it will interfere with the JTAG communication.

5.2 Power measurement

The RLink is designed so that its I/Os are powered by the target board (through the VCC signal of the adapter) so that the RLink I/Os follow the target board's voltage. This implies that any power consumption measurement will be inconsistent if performed on the target board while the RLink is plugged (even if not debugging).

6. Conformity

ROHS Compliance (Restriction of Hazardous Substances)

KEOLABS 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)

KEOLABS 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)

KEOLABS 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)

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

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

For more information on conformity and recycling, please visit the KEOLABS website *www.keolabs.com*

7. Glossary

Term	Description
ARM	Advanced RISC Machine
C816	CoolRISC™ 816
DoC	Debug on Chip
JTAG	Lower-level serial protocol used for the RLink to communicate with the SX8810
RBuilder	Application builder that allows users to configure device peripherals and out put the required C code automatically for their applications. Code is based on libraries provided by the manufacturer.
REva	Raisonance evaluation platform – modular evaluation boards with main evaluation board (motherboard) and daughter boards featuring different microcontrollers
RFlasher	Raisonance Flasher: Programming interface for user-friendly flash programming
Ride7	Raisonance Integrated Development Environment
Ride7-C816	Combined installation of Ride7 and RKit-C816
RLink	Raisonance's versatile in-circuit debugger and programmer for 8-bit and 32-bit microcontrollers
SFR	Special Function Register
SX8810	C816 derivative designed by Semtech

8. Index

9. History

Date	Description
09 Sept 10	Initial version
7 May 2013	Modified cover page, final page and section 1.3 Additional help or information for KEOLABS

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 licence 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.