



RFlasher7

Getting Started and Overview

Document version
121025

Release date
October 2012

RAISONANCE: a KEOLABS brand 

Contents

1. INTRODUCTION.....	4
1.1 Overview.....	4
2. FIRST STEPS WITH RFLASHER.....	5
2.1 Project options.....	6
2.2 File loading.....	7
2.3 Code view.....	7
2.4 Programmer-specific options.....	8
2.5 Blank check.....	8
2.6 Erase.....	8
2.7 Program.....	8
2.8 Reset and run the program.....	9
2.9 Verify.....	9
2.10 Statistics.....	9
2.11 Reload memory	9
2.12 Read.....	9
2.13 Save the memory dump.....	10
3. AUTO MODE.....	11
3.1 Hardware configuration and file loading.....	11
3.2 Auto mode for one device.....	11
3.3 Auto mode for a batch of the same devices.....	12
4. USING PROJECTS.....	14
4.1 Create a new project.....	14



RFlasher

4.2 Add nodes to the project.....	16
5. PROGRAMMER OPTIONS.....	18
5.1 CPU families GettingStarted docs.....	18
5.2 Additional help or information.....	18
6. GLOSSARY.....	19



1. Introduction

1.1 Overview

- RFlasher is a user-friendly Windows interface that allows you to program object files to a microcontroller's flash memory.
- You can connect to microcontrollers in order to manage their non-volatile memory: erase, blank-check, program from the PC memory used by RFlasher, or read-out to the PC memory used by RFlasher.
- You can load object files into the PC memory that is used by RFlasher and then view, edit them and save them.
- RFlasher also provides a configurable automatic mode for grouping commands (for example, erase and program in one single click) and for mass programming (for example programming multiple devices in sequence).
- RFlasher provides a third mode of operation: project mode. The project mode is slightly more complex to use but offers more possibilities such as allowing you to save the configuration for each application in case you have several different applications to handle. It also allows you to create projects from several object files.
- RFlasher can be used to program microcontrollers in any of several supported families using the connection protocol specific to each. The list of supported devices is not fixed and will increase as time passes; it is available in the 'Project Options' section.
- This document provides information about RFlasher features that are common to all supported targets and/or programming tools, general user information for the supported tools and device families. For details about features that are specific to your target and/or programming tool, please refer to the on-line help, after having selected your target device and programming tool.

Note: RIDE users

RFlasher is simply a special, limited mode of Ride7 (Raisonance Integrated Development Environment). Therefore, everything that RFlasher does can also be done with Ride7. Among other things, Ride7 can create and open RFlasher projects, program and erase devices, etc. The opposite, however, is not true.

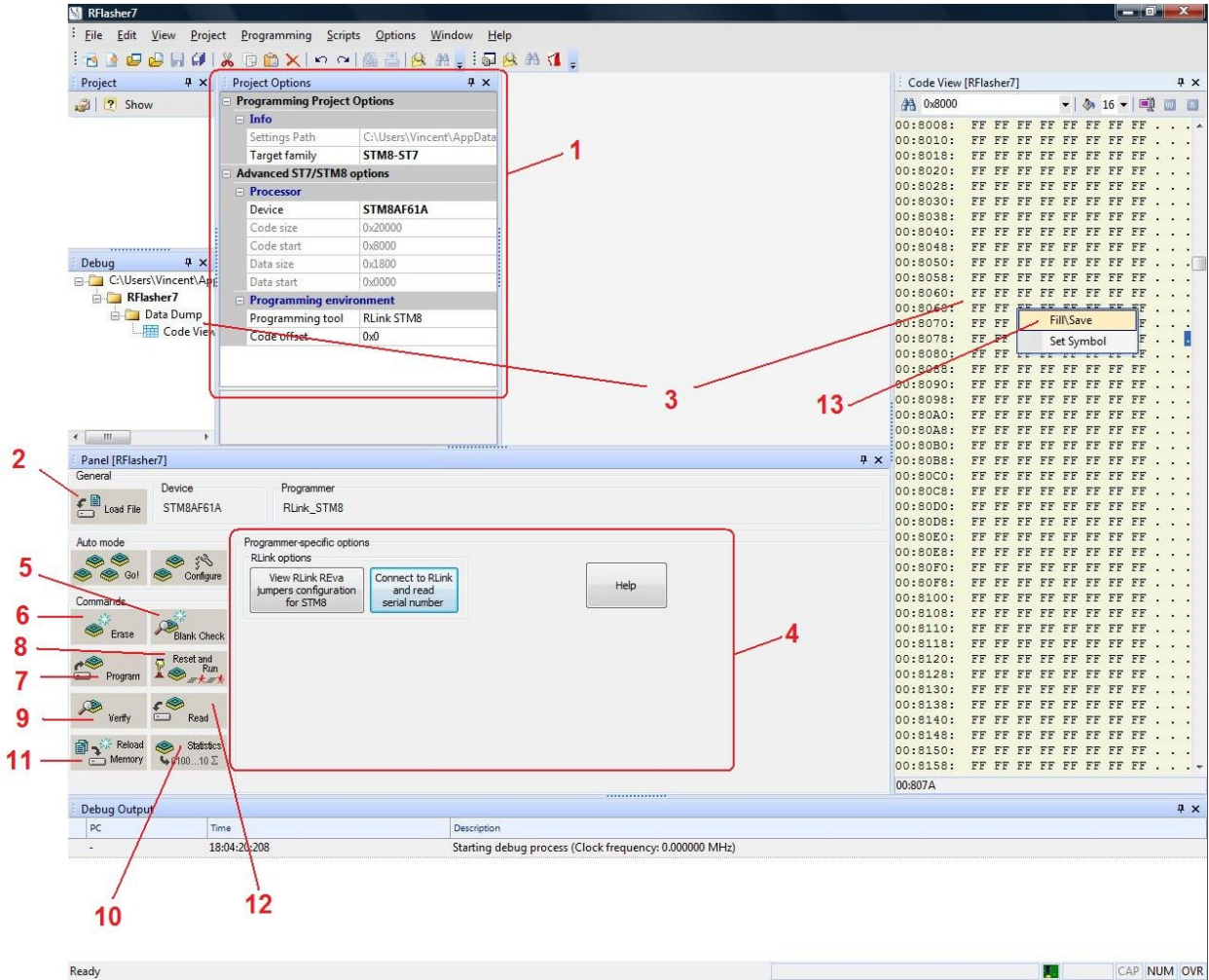


2. First steps with RFlasher

This chapter takes you through the steps 1-13 for programming your target with Rflasher.

Note: Numbers on the figure

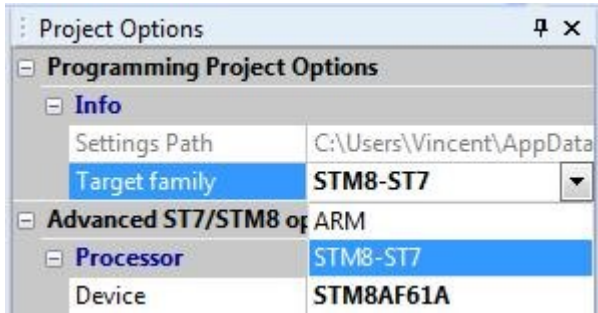
The numbers on Figure 2.1 represent both the order of the steps and the section number reference in this document. For example, Step 4 refers to section 2.4.



2.1 Project options

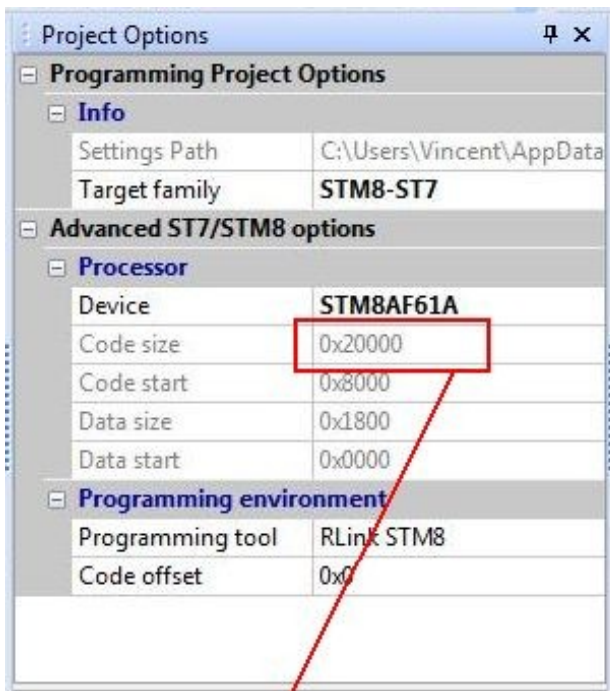
The **'Project options'** section allows you to configure the software. In this section, you must select the target CPU family (ARM, STM8-ST7, ...) and derivative (STR711FR2, ST7LITE39, etc.), and the programming tool. (RLink, ...)

Target family selection:



If you close this pane, you can open it again by selecting the menu **'View'->'View Properties Window'**.

If a device is not present in the list, then it means that RFlasher does not support it. Check for software updates on the Raisonance website, as it might have been added since your software was built.



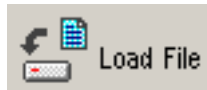
FLASH memory start address

Please note the Flash memory start address for future reference:



2.2 File loading

This button allows you to load the content of an object file into the PC memory that is used by RFlasher prior to programming the target device's memory.



You can load a file in any of the formats that Raisonance tools recognise as ready to program (HEX, ELF, BIN, AOF ...).

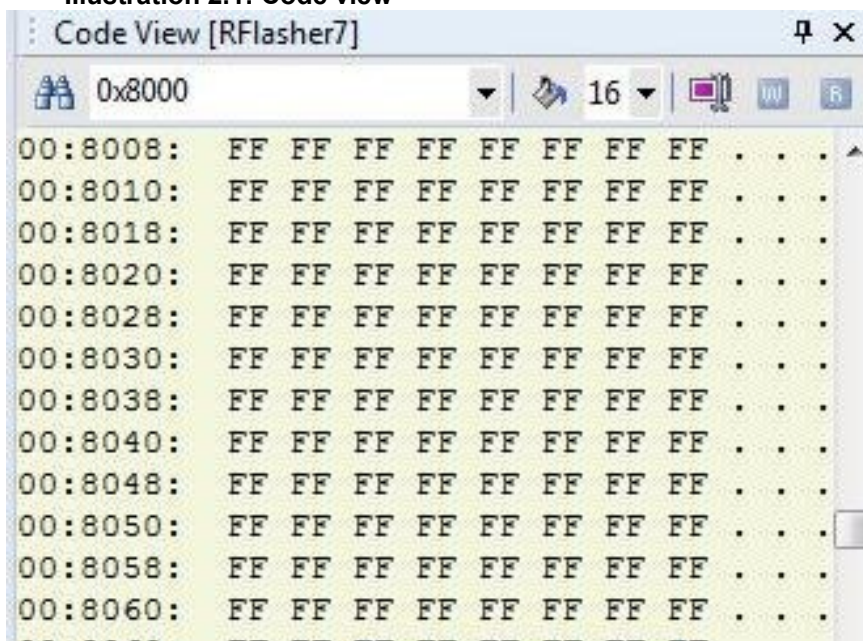
Select your file, then click **'Open'**.

The file's content is now displayed in the code view described below.

2.3 Code view

Code view is open by default in the main window, situated on the right of the screen. It displays the

Illustration 2.1: Code view



code used in the project.

This window shows the code loaded in the PC memory that is used by RFlasher prior to programming the target.

Note: Code view before programming

At this stage, the display is not an image of your device's memory as the device has not yet been programmed. The file has only been loaded in the PC memory used by RFlasher.

2.4 Programmer-specific options

A part of the main window panel is reserved for programmer-specific options. You can refer to chapter 5 page 18 for your programming tool. There is often a help button in this section that you can click to get further information about programming-tool specific options.

Note: Set options before communicating with device

You must make sure that these options are correctly set before attempting **any** communication with the device.

You are about to communicate with the chip for the first time, so please make sure that the Programmer is connected to the PC and the target board, and that the target board is powered. Some programmers provide connection tests in their programmer-specific options. If this is the case with yours, please use them to check the connections and options.

2.5 Blank check

This button allows you to check if the device is blank:



After communication with the target device, a dialog box informs you of the result of the operation. If the device is blank, you can jump directly to step 2.7; otherwise you must erase your device first.

2.6 Erase

Click on this button to erase your device's Flash memory prior to programming it:



You can do a blank check again after erasing to make sure your device is ready.

2.7 Program

You can program your device by clicking on this button:





The program operation consists of loading the content of the PC memory that is used by RFlasher (initialized with "Load File") into the target device's flash. After communicating with the device, a dialog box informs you of the success of the operation.

2.8 Reset and run the program



Click on this button to make the program run on your device.

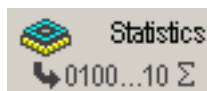
2.9 Verify

Click on the "verify" button to perform a strict verification of the program integrity in the Flash memory by comparing its content with the program loaded in the PC memory, which is visible in the Memory View:



The verification status is given in a dialog box at the end of verification.

2.10 Statistics



Depending on your hardware, you can click on this button to obtain statistics on the target. Usually this includes the CheckSum.

2.11 Reload memory

Close and reopen RFlasher to clear its memory, or click on the "Reload Memory" button, which has the same effect:



If you go to the start address of Flash memory in the memory view, you should not longer see any data.

2.12 Read

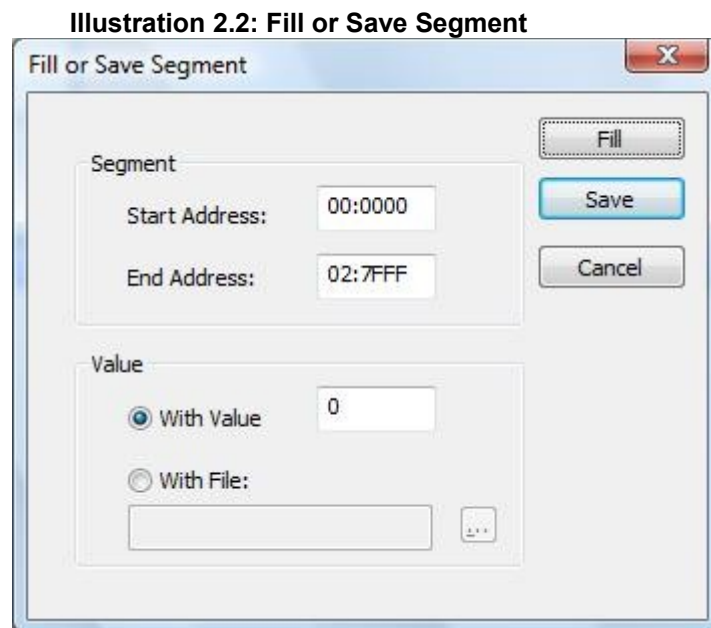
Click on this button to copy the device's Flash memory content to the PC memory used by RFlasher:



This button allows RFlasher to dump (make an exact copy of) the device Flash memory into the RFlasher memory view. To see the Flash memory, go to its start address using the "Search" field.

2.13 Save the memory dump

You can then save the memory dump to a file for future reference. To do this, note the start address and the end address of the memory block you want to save, then right-click on the memory view and choose "Fill/Save". The following dialog box appears.



Enter the start and end addresses, then select "with file" and click on "Browse...". Here you can enter a file name such as "dump.hex", and click on "Open". Back in the "Fill or Save segment" dialog box, click on the "save" button. The file `dump.hex` is created, containing the memory dump in Intel HEX format.



3. Auto Mode

RFlasher provides an Automatic Mode in which you can define a sequence of operations to be executed on a single click. This way, you can save a lot of time when processing a batch of devices for instance.

3.1 Hardware configuration and file loading

Check that your programmer and device are correctly connected, and follow the steps Error: Reference source not found to 2.3 beginning on page Error: Reference source not found, to prepare the PC memory with the object file that you will load in your device.

3.2 Auto mode for one device



Click on the "Configure" button to select the operations to include in your sequence.

The following dialog box appears:

Illustration 3.1: Automatic mode configuration



Select the options you want, but **uncheck the last one** ("Loop") for now. Click "OK" to validate your choices.





Click on the "Go!" button to launch the auto mode sequence. Once the operations are completed, a dialog box indicates the success or failure of the operations:

Illustration 3.2: Automatic mode end message



3.3 Auto mode for a batch of the same devices



Select the "Configure" button again, and this time check the "Loop" option as if you had several devices to program sequentially with the same program, and click "OK".

When you select the "Go!" button, it launches the process. At the end of the sequence, a dialog box asks you whether you want to program another device.

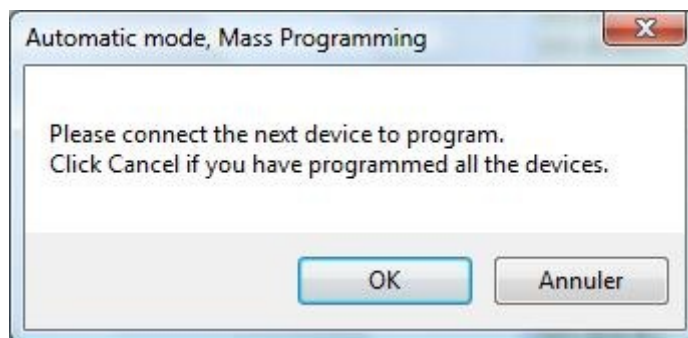


Illustration 3.3.:

Click "OK" to program each device in the batch.

When you have programmed all the devices, click "cancel" to end the sequence loop.

The dialog box indicates, this time, the statistics of success or failure in programming your devices:





Illustration 3.4.:



4. Using projects

You may need to organize your work when using a programming software: for example to handle different HEX files to load in the same device at different addresses, to insert references to documents, etc.

In this case, you can gather your information and program files in a project for management within RFlasher.

Important remark:

One of the main goals of Project management in RFlasher is to provide a way for restoring your configuration: chosen target and associated options, object files to include, etc.

If you open a previously created project, all your options will be recovered and your Memory content will be reloaded (i.e. your object files will be mapped in memory).

4.1 Create a new project

In order to create a new project, select "New -> Project" from the "File" menu. The New Project dialog box appears:



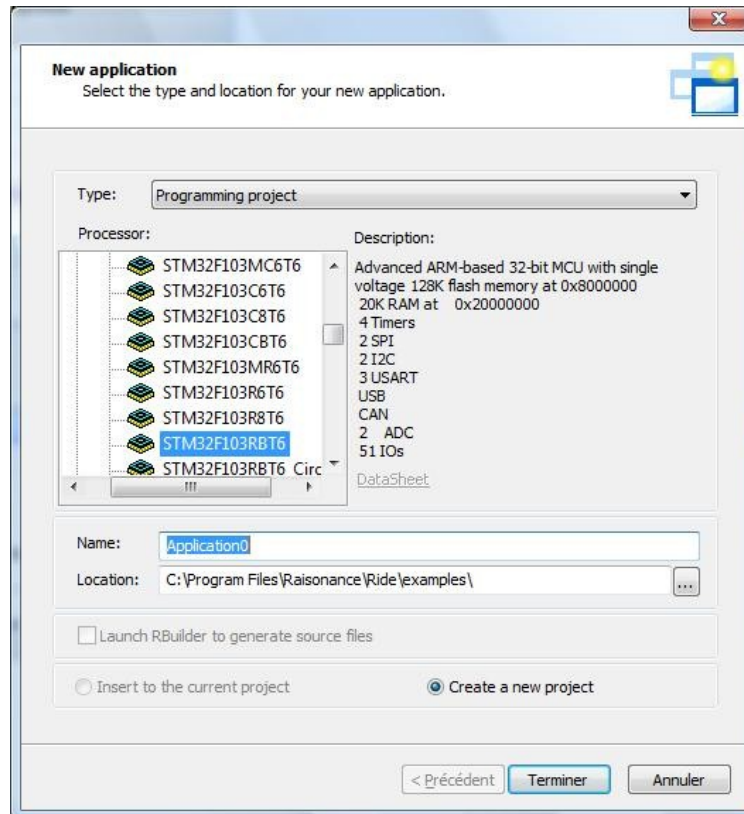


Illustration 4.1.:

Choose a project name such as "test1". Select a working directory (for example "C:\tmp\RFlasher7\test1") and a Target family, then click "Finish".

The next steps are related to your hardware configuration, please refer to chapter Error: Reference source not found, section Error: Reference source not found, page Error: Reference source not found.

After configuring the hardware and clicking on Finish (section Error: Reference source not found), the project panel should appear on the left side of the main window.

It is composed of three tabs, accessible via their titles on top of the active view:

- Project
- Programmer Views
- Documentation

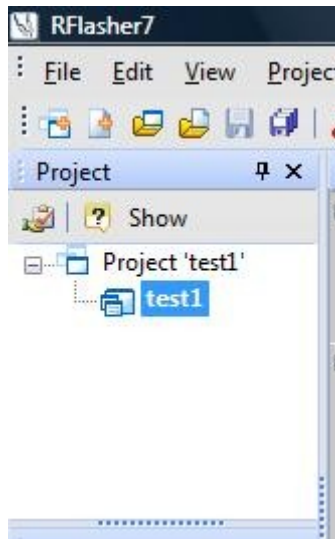


Illustration 4.2.:

4.2 Add nodes to the project

In order to include programs in your project, follow this procedure:

1. Click on the project file (`test1`) to select it.
2. Select the menu "Project" -> "Add Existing Item". A file-selection dialog box opens.
3. Choose the object file to include from anywhere on your disk.
4. Click "Open"

Repeat these steps for any object file or documentation file you want to include.

You can change the order of objects in the project display by selecting an object and pressing Shift+↑ or Shift+↓ to place your nodes in the expected order.



Note about object file inclusion:

Including several object files can be useful if you have, for example a HEX file containing the program code and another HEX file containing initialization values to store in Flash memory. In this case, you must ensure (using the Memory View) that your memory blocks don't overlap and that the stored values are in the expected address so that the program can retrieve them.

Note about documentation files:

You can include documentation files in the **PDF**, **HTML** (.HTM extension only), **DOC** and **TXT** formats in the Project View (using “**Add node**”). These files, once included, can also be found under the Documentation View. To open them, double-click on their node and the appropriate application for viewing will be automatically opened.

Note about Memory view:

When using projects, the default application memory content is not “blank” but contains the mapping of the object files included in your project. As a consequence, if you click the “**Reload memory**” button, the memory will not become “blank” but your object files will be reloaded. This is not the same behavior as when programming without a project. Without a project the default application memory is blank.



5. Programmer options

5.1 CPU families GettingStarted docs

The device-specific and programmer-specific configuration and features are the same in RFlasher7 as the respective specific debugging options in Ride7.

See the Ride7 documentation for your target CPU's family: GettingStartedARM, GettingStartedSTM8, GettingStartedPPC, etc.

5.2 Additional help or information

You may find additional documentation in the DOCS folder inside the Ride installation. In addition, help is available via the Help menu in Ride.

You can also visit the Raisonance website: <http://www.raisonance.com/>

Address: KEOLABS S.A.S.
17, Avenue Jean Kuntzmann,
38330 Montbonnot Saint Martin
France

Telephone: +33 4 76 61 02 30

Fax: +33 4 76 41 81 68

Email: info@raisonance.com

If you find any errors or omissions, or if you have suggestions for improving this manual, please let us know by emailing: support@raisonance.com



6. Glossary

Programmer / programming tool

The hardware used to send code from a workstation to the target microcontroller, ie to write the code into the microcontroller Flash memory.

Target

The microcontroller or embedded device on which the current program is meant to run. The target is connected to the workstation running RFlasher via the programming tool.

History

Author	Date	Modification
Vincent	Mar 2008	Creation
Vincent	Oct 2012	Reference GettingStarted for specific options. Raisonance is a KEOLABS brand



Disclaimer

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.

Please check the printed version against the .pdf installed on the computer in the Ride installation directory, 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 © Raisonance 1987-2008 All rights reserved.

