
Technical Notes

Renesas 78K0/78K0R/RL78 Family On-Chip Emulation

Contents

Contents.....	1
1 Introduction	2
2 Emulation options.....	3
2.1 Hardware Options	3
2.2 Initialization Sequence	3
2.3 Initialization After Download Sequence	5
3 CPU Setup	6
3.1 General Options	6
3.2 Debugging Options	7
3.3 Reset.....	8
3.4 Advanced Options.....	9
3.4.1 78k0.....	9
3.4.2 78k0R	10
3.4.3 RL78.....	12
4 Internal FLASH Programming	13
5 Real-Time Memory Access	13
6 Access Breakpoints	13
7 Getting Started.....	15
8 Troubleshooting.....	15

1 Introduction

This documents covers following families: 78K0R, 78K0 and RL78.

These Renesas families use proprietary serial interface for in-circuit debugging of microcontrollers and FLASH programming. Such firmware is implemented on the CPU silicon providing a comprehensive set of debug functionalities. Communication interface uses:

- CPU clock lines X1 and X2 on 78K0
- TOOL0 and TOOL1 or TOOL0 (single wire communication) on 78K0R
- TOOL0 on RL78

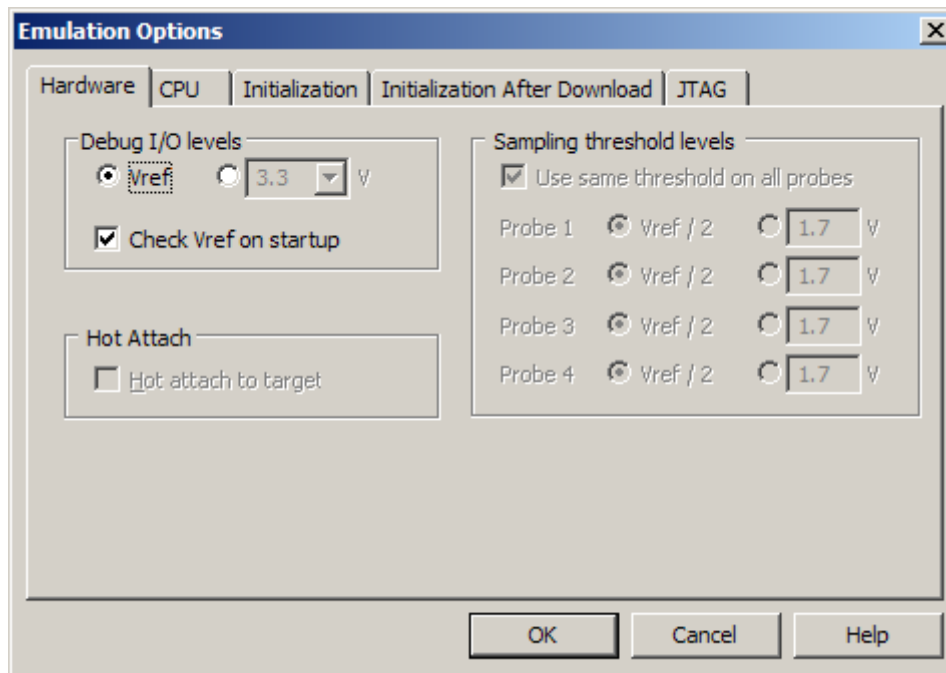
Debug Features

The emulation system features:

- 1 execution hardware breakpoint
- Unlimited software breakpoints, including in the internal FLASH
- 1 access breakpoint
- Internal FLASH programming

2 Emulation options

2.1 Hardware Options



Emulation options, Hardware pane

Debug I/O levels

The development system can be configured in a way that the debug signals are driven at 3.3V, 5V or target voltage level (Vref: 1.8V -5V).

When 'Vref' Debug I/O level is selected, a voltage applied to the belonging reference voltage pin on the target debug connector is used as a reference voltage for voltage follower, which powers buffers, driving the debug signals. The user must ensure that the target power supply is connected to the Vref pin on the target connector and that it is switched on before the debug session is started. If these two conditions are not met, it is highly probably that the initial debug connection will fail already. However in some cases it may succeed but then the system will behave abnormal.

Check Vref on startup

This option is available for iC5000 development system only. When checked, the system will check the presence of voltage on the Vref pin on the target debug connector. If no voltage or too low voltage is detected, a warning message is pop up.

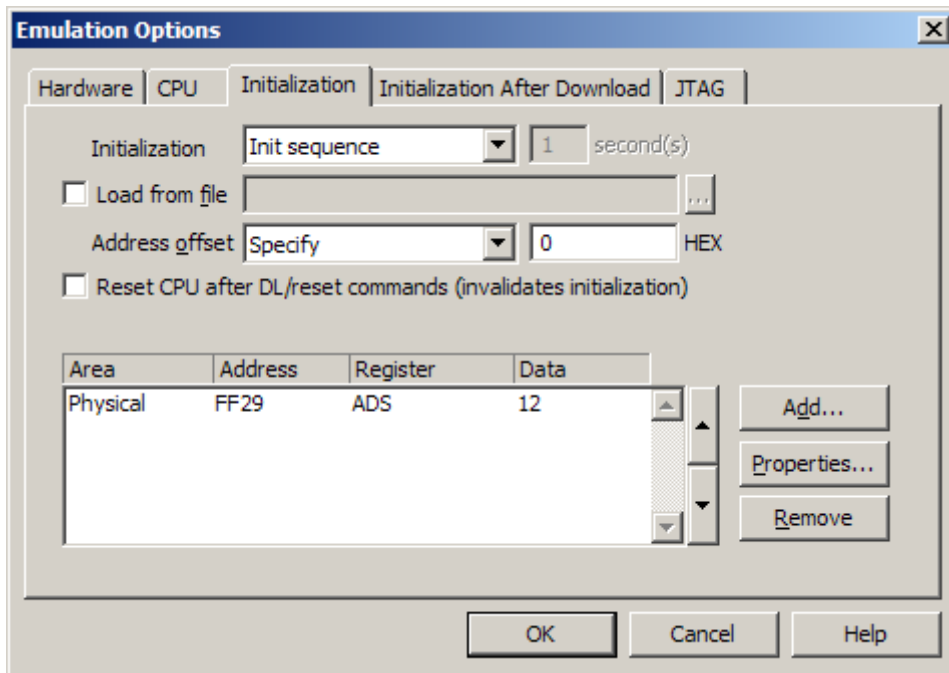
2.2 Initialization Sequence

Before the flash programming or download can take place, the user must ensure that the memory is accessible. This is very important since there are many applications using memory resources (e.g. external RAM, external flash), which are not accessible after the CPU reset. In that case, the debugger must execute after the CPU reset a so called initialization sequence, which configures necessary CPU chip selects and then the download or flash programming can actually take place. The user must set up the initialization sequence based on his application.

Note: Normally, there is no need for initialization sequence in case of a single chip application/CPU.

The initialization sequence can be set up in two ways:

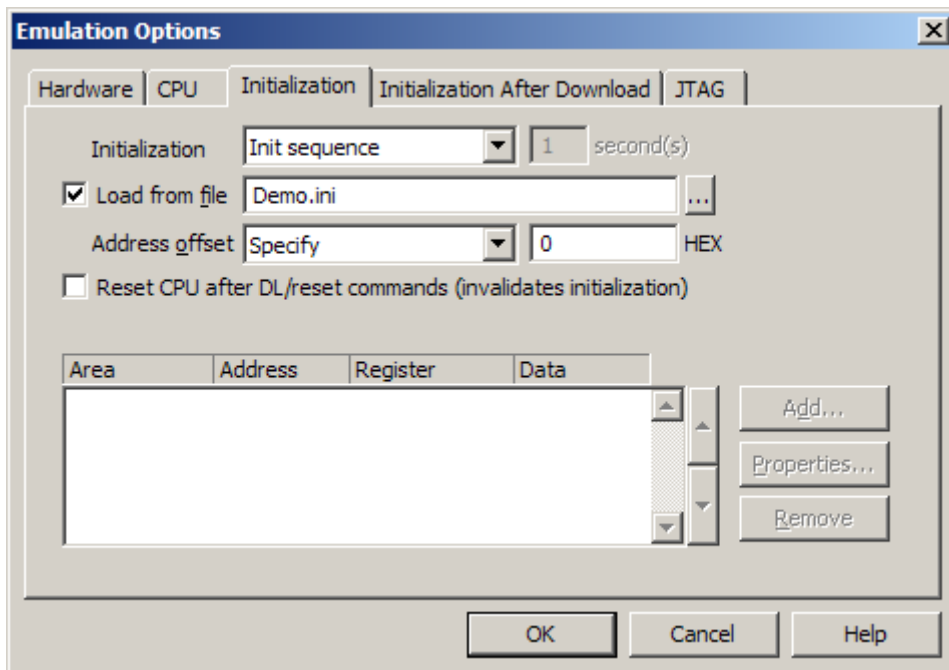
1. Set up the initialization sequence by adding necessary register writes directly in the Initialization page within winIDEA.



2. WinIDEA accepts initialization sequence as a text file with .ini extension. The file must be written according to the syntax specified in the appendix in the hardware user's guide.

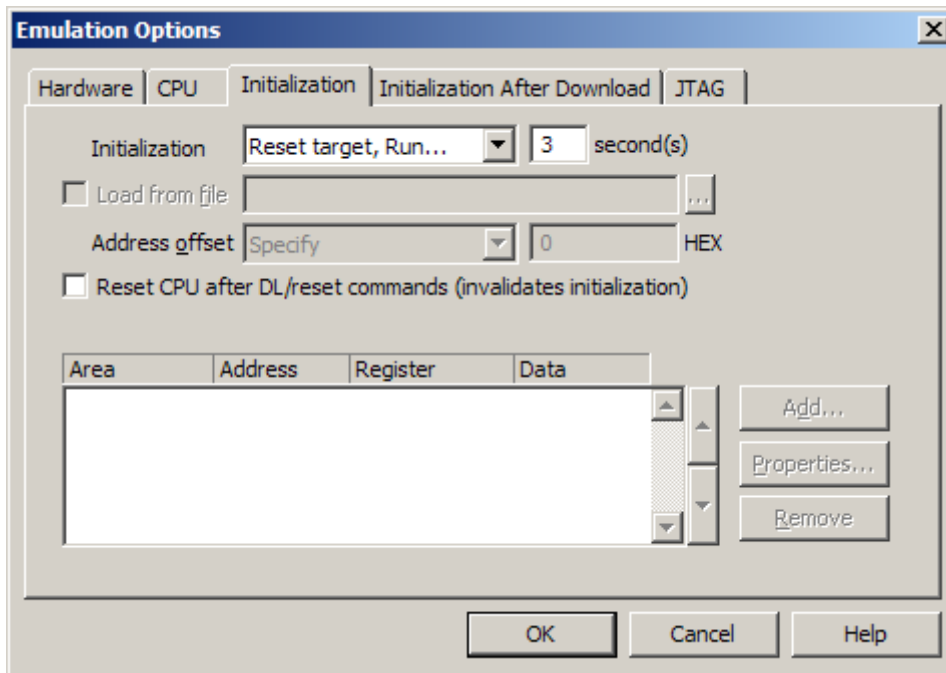
Excerpt from the sample Demo.ini file:

```
S PTBD B 12          //comment
S PTBDD B FF
```



The advantage of the second method is that you can simply distribute your .ini file among different workspaces and users. Additionally, you can easily comment out some line while debugging the initialization sequence itself.

There is also a third method, which can be used too but it's not highly recommended for the start up. The user can initialize the CPU by executing part of the code in the target ROM for X seconds by using 'Reset and run for X sec' option.



Reset CPU after DL/reset commands (invalidates initialization)

When the option is checked, the configured initialization, which is probably required for debug download (flash programming), is invalidated after the debug download and debug reset.

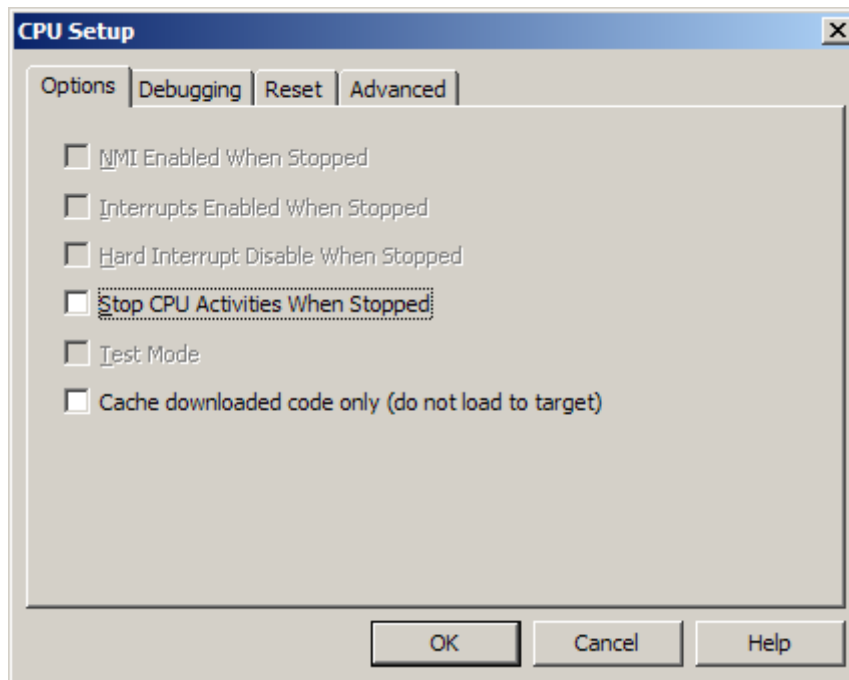
2.3 Initialization After Download Sequence

There are rare cases when a different initialization of the CPU is required for debug download and after the debug download. When Initialization After Download is used, typically the 'Reset CPU after DL/reset commands (invalidates initialization)' option described in the previous chapter is checked too.

Note that this dialog is available for 78K0 family only

3 CPU Setup

3.1 General Options



Stop CPU Activities When Stopped

When the option is checked, all internal peripherals like timers and counters are stopped when the application is stopped. Otherwise, timers and counters remain running while the program is stopped. Usually, when the option is checked, the emulation system behaves more consistently while stepping through the program. While being aware of the consequences, it is up to the user whether the option is checked or not.

For instance, it's recommended that a timer, which generates interrupts, is stopped when the application is stopped. Otherwise, the CPU would first service all pending interrupts (generated by the timer while the application was stopped) after the application is resumed. Such behavior is far away from the actual behavior of the target application.

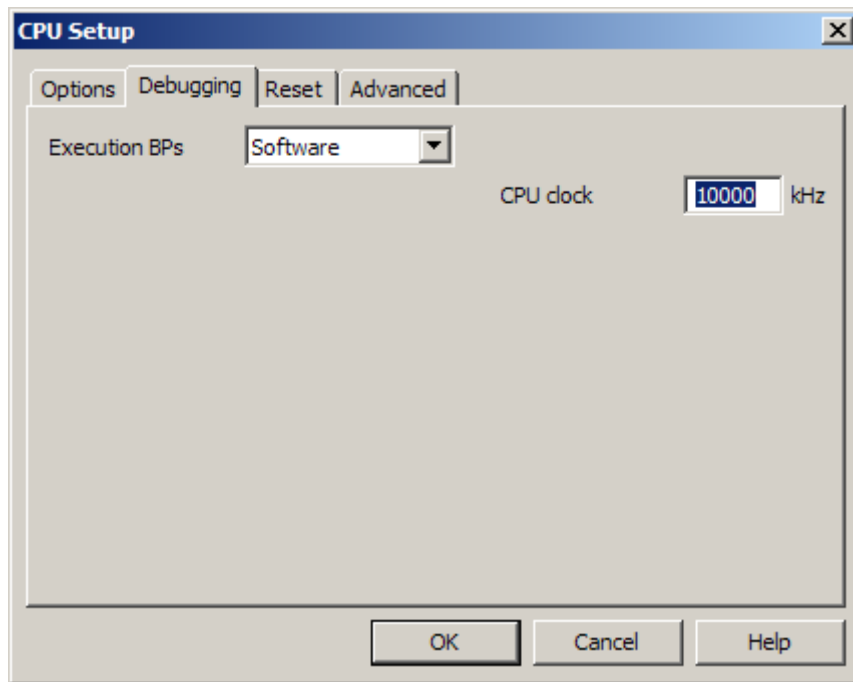
Cache downloaded code only (do not load to target)

When this option is checked, the download files will not propagate to the target using standard debug download but the Target download files will.

In cases, where the application is previously programmed in the target or it's programmed through the flash programming dialog, the user may uncheck 'Load code' in the 'Properties' dialog when specifying the debug download file(s). By doing so, the debugger loads only the necessary debug information for high level debugging while it doesn't load any code. However, debug functionalities like ETM and Nexus trace will not work then since an exact code image of the executed code is required as a prerequisite for the correct trace program flow reconstruction. This applies also for the call stack on some CPU platforms. In such applications, 'Load code' option should remain checked and 'Cache downloaded code only (do not load to target)' option checked instead. This will yield in debug information and code image loaded to the debugger but no memory writes will propagate to the target, which otherwise normally load the code to the target.

3.2 Debugging Options

Note: This dialog is available for 78K0 family only.



Debugging Options

Execution Breakpoints

Hardware Breakpoints

Hardware breakpoints are breakpoints that are already provided by the CPU. The number of hardware breakpoints is limited to one. The advantage is that they function anywhere in the CPU space, which is not the case for software breakpoints, which normally cannot be used in the FLASH memory, non-writeable memory (ROM) or self-modifying code. If the option 'Use hardware breakpoints' is selected, only hardware breakpoints are used for execution breakpoints.

Note that the debugger, when executing source step debug command, uses one breakpoint. Hence, when all available hardware breakpoints are used as execution breakpoints, the debugger may fail to execute debug step. The debugger offers 'Reserve one breakpoint for high-level debugging' option in the Debug/Debug Options/Debugging' tab to circumvent this. By default this option is checked and the user can uncheck it anytime.

Note: 78K0R and RL78 families don't provide a hardware execution breakpoint. However, the on-chip debug logic provides a post execution breakpoint, which can be used and configured through the access breakpoints dialog.

Software Breakpoints

Available hardware breakpoints often prove to be insufficient. Then the debugger can use unlimited software breakpoints to work around this limitation. Note that the debugger features unlimited software breakpoints in the internal flash too.

When a software breakpoint is being used, the program first attempts to modify the source code by placing a break instruction into the code. If setting software breakpoint fails, a hardware breakpoint is used instead.

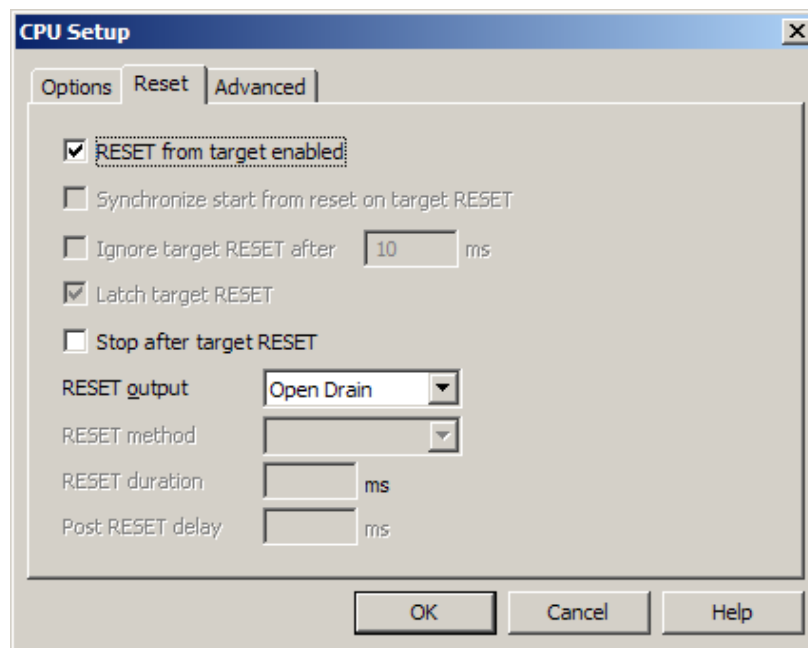
Using flash software breakpoints

A flash device has a limited number of programming cycles. Belonging flash sector is erased and programmed every time when a software breakpoint is set or removed. The debugger sets breakpoints hidden from the user also when a source step is executed. In worst case, a flash may become worn out due to intense and long lasting debugging using flash software breakpoints.

CPU Clock

Here you specify CPU clock frequency. This frequency is used by serial debug interface (X1, X2) and is also required by the internal flash programming.

3.3 Reset



Reset Options

RESET From Target Enabled

Beside the debugger, the target can have additional external reset sources, like power-on reset, watchdog circuitry or even reset push-button. In general, it's recommended to disable all external reset sources in the target, which may disturb the debugger in a way that serial communication is lost and complete system needs to be reinitialized.

It's recommended that all reset sources are designed as an open drain type. 'Reset from Target Enabled' option in the 'CPU Setup/Options' tab must be normally checked to assure safer debugging. Then the debugger can detect any reset source and service it properly.

Since target reset lines are designed as an open drain type, the debugger can detect all resets, even if they have been initiated by hardware other than the emulator itself. In certain applications, though, the requirement to disable this type of checking is required.

To disable reset sources from the target to be detected by the debugger, uncheck the 'RESET From Target Enabled' option. In this case, only the emulator will be able to generate a reset and the debugger will ignore all reset sources from the target.

Note: Wrong setting of this option can significantly change the operation of the target!

Stop after target RESET

CPU can be optionally stopped after the CPU reset is detected and handled. If the option is unchecked, the application is resumed upon reset release.

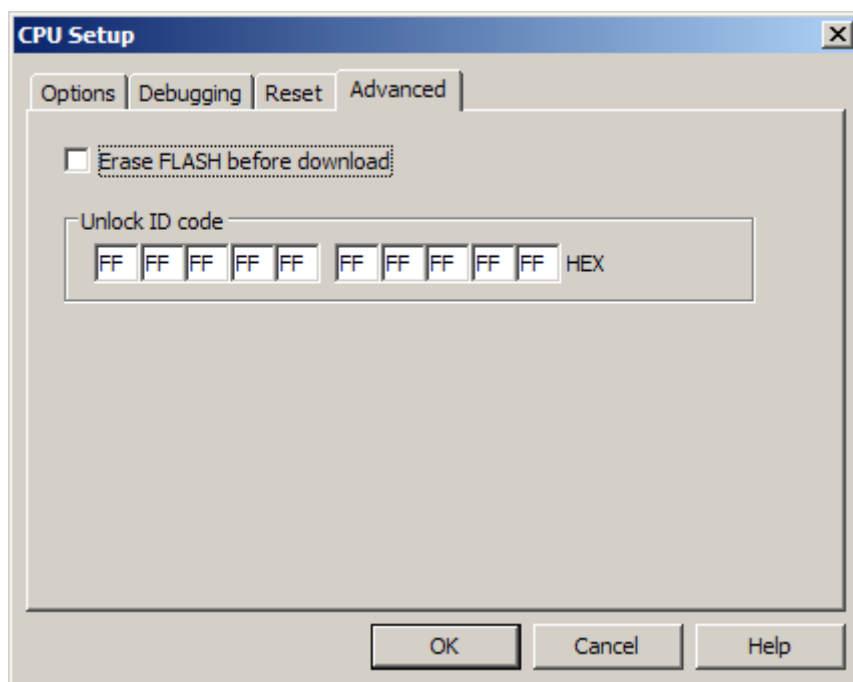
This option is not available on 78K0 family.

RESET Output

Depending on the target application requirement, RESET output can be configured for either Open Drain or Push-Pull operation type.

3.4 Advanced Options

3.4.1 78k0



Renesas 78k0 Family Advanced CPU Options

Beside of the user's code an OCD monitor and monitor parameters are always loaded to the internal flash.

Reserved memory locations for monitor are:

- 0x0002 - 0x0003: NMI vector
- 0x007E - 0x007F:CALLT [1F] vector
- 0x0084: Monitor startup configuration byte
- 0x008F – 0x01FF:Monitor

If you try to load user code or data to these reserved memory locations, winIDEA will display a warning message.

Additional memory locations which are used by monitor but are not reserved are:

0x0085 – 0x008E: 10 bytes ID code

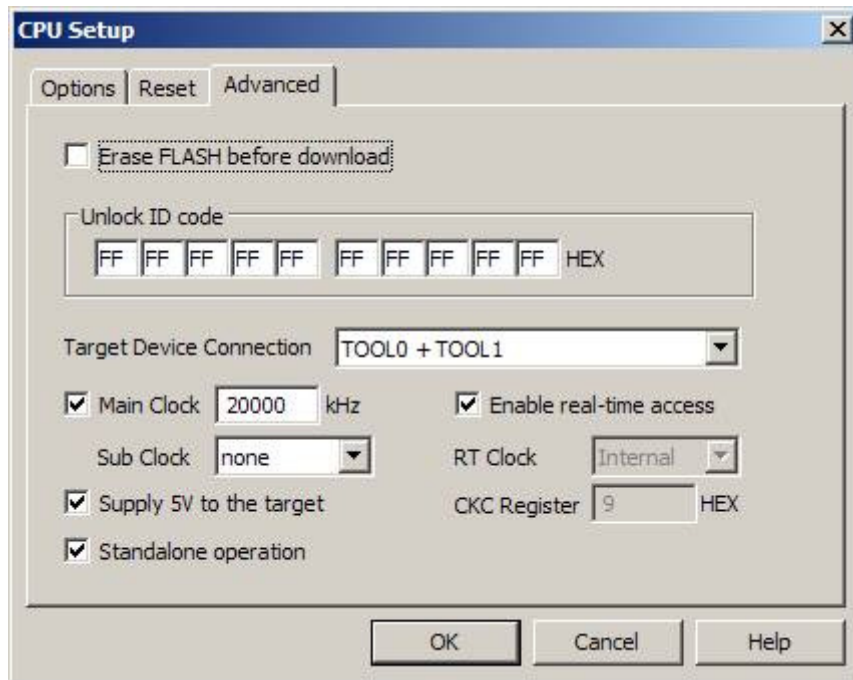
Erase FLASH before download

Check the option if complete flash should be erased prior to the debug download. Otherwise, only the necessary sectors are erased.

Unlock ID code

After reset 10 bytes ID code specified in winIDEA is compared with 10 bytes from memory location (0x0085 – 0x008E). Based on ID check result and Monitor startup configuration byte (address 0x0084) emulation is enabled or not.

3.4.2 78k0R



Renesas 78k0R Family Advanced CPU Options

Beside of the user's code an OCD monitor and monitor parameters are always loaded to the internal flash. Reserved memory locations are:

- 0x00000 – 0x00004 (Reset and NMI vectors)
- 0x000C3 - (OCD option byte)
- 0x000D0 – 0x000D7 (jump table)
- 0x008F - Monitor code

Monitor code is always loaded to the last block of the internal flash and uses 6 bytes of user stack. Two different monitors can be loaded:

- min size monitor (0x58 bytes)
- full size monitor (0x400 bytes)

Code size depends of available monitor functions.

Real time access or 1 wire tool interface requires full monitor, while minimum monitor is used when 2 wire tool interface (without RT access) is selected.

Erase FLASH before download

Check the option if complete flash should be erased prior to the debug download. Otherwise, only the necessary sectors are erased.

Unlock ID code

After reset 10 bytes ID code specified in winIDEA is compared with 10 bytes from memory location (0x0085 – 0x008E). Based on ID check result and Monitor startup configuration byte (address 0x0084) emulation is enabled or not.

Target Device Connection

Specifies with how many wires the emulator is connected to the target:

- 2 wire (TOOL0+TOOL1)
- 1 wire (TOOL0) (full monitor is selected)

Main clock and Sub Clock

Different clock sources can be used while CPU is running:

- internal high speed oscillator (4MHz/8MHz)
- Main clock
- Sub clock

Main clock and sub clock are optional. When used, frequency must be specified.

Enable real-time access.

When checked RT access is enabled (complete monitor is used & loaded then).

RT Clock and CKC Register

Clock at which CPU is running must be selected when 1 wire interface is selected. At the same time the value of CKC register must be specified.

Supply 5V to the target

When enabled in winIDEA, emulator supplies 5V at pin 4 (VCC) on debug connector.

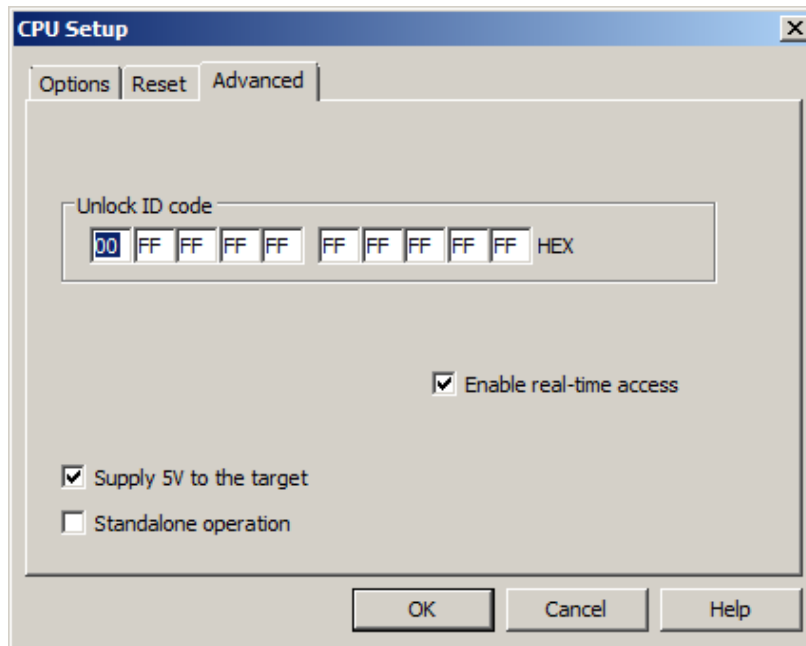
Standalone operation

When this option is enabled, only user's code without OCD monitor is loaded to internal flash during download. Emulator is working as flash programmer.

After the debug download is finished 'NO CONNECTION' debug status is displayed.



3.4.3 RL78



Renesas RL78 Family Advanced CPU Options

Beside of BFA's monitor that is already programmed into the chip, emulation requires additional monitors which are loaded to user's flash area.

Reserved memory locations:

- 0x00002 - 0x00003
- 0x000CE - 0x000D7

Monitor code is always loaded to last block of internal flash and uses 6 bytes of user stack. Two different monitors can be loaded:

- 0x100 flash locations at the end of program flash if real-time access is disabled
- 0x200 flash locations at the end of program flash if real-time access is enabled

Reset vector at 0x00000-0x00001 is redirected to monitor, therefore verify download reports error.

RL78 chip doesn't support chip (mass) erase command any more. Because of that only flash blocks where new code and monitor fits in are programmed during download. Unused flash blocks are not erased during download. To completely erase the chip use 'Hardware/Flash/Mass erase' command in winIDEA. After mass erase debug monitor is programmed into flash.

Unlock ID code

After reset 10 bytes ID code specified in winIDEA is compared with 10 bytes from memory location (0x00C4 – 0x00CD). If invalid ID code is specified emulation will not start. CPU considers 10 0xFFx as invalid code.

Enable real-time access.

When checked RT access is enabled (complete monitor is used & loaded then).

Supply 5V to the target

When enabled in winIDEA, emulator supplies 5V at pin 8 (VDD) on debug connector.

Standalone operation

When this option is enabled, only user's code without OCD monitor is loaded to internal flash during download. Emulator is working as flash programmer. Debugging is impossible in this case. The only debug control is RESET.

After the debug download is finished 'NO CONNECTION' debug status is displayed.



4 Internal FLASH Programming

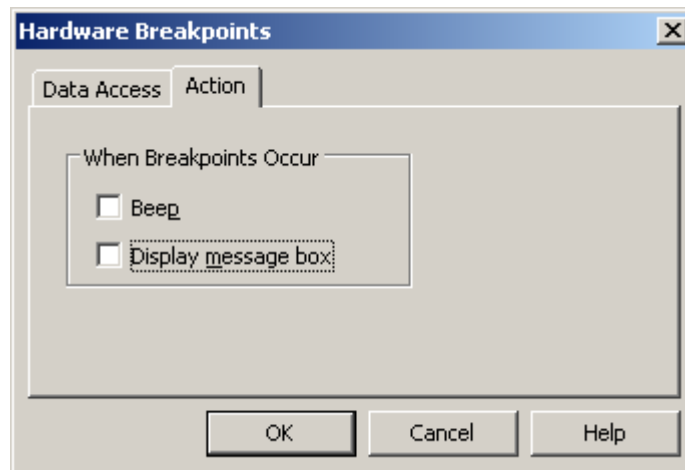
Renesas microcontrollers have internal flash, which is programmed through the standard debug download; thereby no standard FLASH setup dialog is available. The debugger recognizes which code from the download file fits in the FLASH. All necessary FLASH programming settings are done in the 'CPU Setup/Advanced' dialog.

5 Real-Time Memory Access

Renesas 78k0 family does not support real-time memory access, while 78k0R and RL78 do support real-time access.

6 Access Breakpoints

Access breakpoints are breakpoints that are already provided by the CPU.

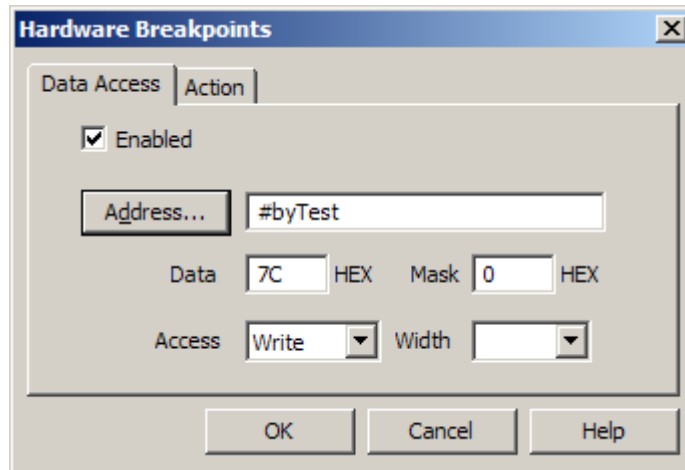


Hardware Breakpoints Configuration – Action tab

When access breakpoint is hit a display message can be shown or you can set a PC to play a sound.

78K0

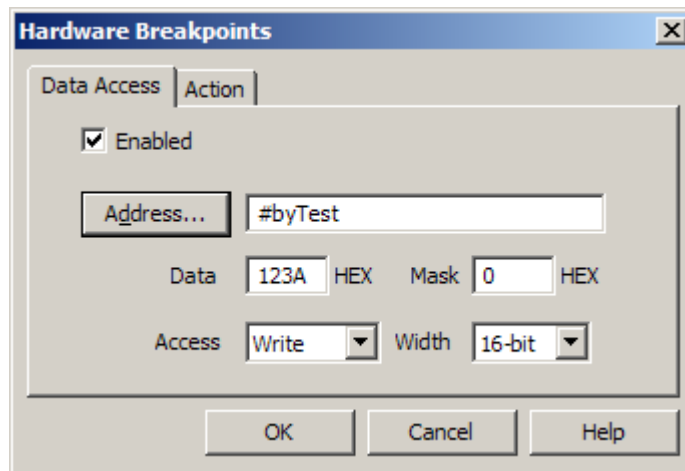
The number of access breakpoints is limited to one and can be set in a range 0xF800-0xFFFF.



Data value, data mask and access type can be specified.

78K0R & RL78

The number of access breakpoints is limited to one.



Data value, data mask, access type and bus width can be specified.

7 Getting Started

- 1) Connect the development system with the target.
- 2) Make sure that the target debug connector pinout matches with the one requested by a debug tool. If it doesn't, make some adaptation to comply with the standard connector otherwise the target or the debug tool may be damaged.
- 3) Power up the emulator first and then the target.
- 4) Execute the debug reset.
- 5) The application should stop on location to which the reset vector points.
- 6) Open memory window at the internal microcontroller RAM location and check whether you are able to modify its content.
- 7) If you passed all 6 steps successfully, the debugger is operational and you may proceed to download the code in the internal microcontroller flash.
- 8) Check 'Erase FLASH before download' options in the 'CPU Setup/Advanced' tab.
- 9) Specify the download in the 'Debug/Files for download/Download files' tab.
- 10) Execute the Debug download, which should download the code in the internal microcontroller flash.

8 Troubleshooting

- Make sure that the power supply is applied to the target serial connector when 'Vref' is selected for Debug I/O levels in the Hardware/Emulator Options/Hardware tab, otherwise emulation fails or may behave unpredictably.
- When performing any kind of checksum, remove all software breakpoints since they may impact the checksum result.

Disclaimer: iSYSTEM assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information herein.

© iSYSTEM. All rights reserved.