
Technical Notes

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

This document is intended to be used together with the CPU reference manual provided by the silicon vendor. This document assumes knowledge of the CPU functionality and the terminology and concepts defined and explained in the CPU reference manual. Basic knowledge of winIDEA is also necessary. This document deals with specifics and advanced details and it is not meant as a basic or introductory text.

Contents

Contents	1
1 Introduction	2
2 Emulation options	3
2.1 Hardware Options	3
2.2 Initialization Sequence	3
3 CPU Setup	4
3.1 General Options	4
3.2 Debugging Options	5
3.3 Reset	6
3.4 Advanced Options	7
3.4.1 78K0	7
3.4.2 78K0R	8
3.4.3 RL78	10
4 Internal FLASH Programming	11
5 Emulation Notes	12
5.1 Starting the debug session with Reset command	12
6 Real-Time Memory Access	12
7 Access Breakpoints	14
8 Getting Started	15
9 Trace	16
10 Troubleshooting	17

1 Introduction

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

Renesas on-chip debug support features a proprietary serial debug interface and a debug firmware providing a comprehensive set of debug functionalities. Communication interface uses:

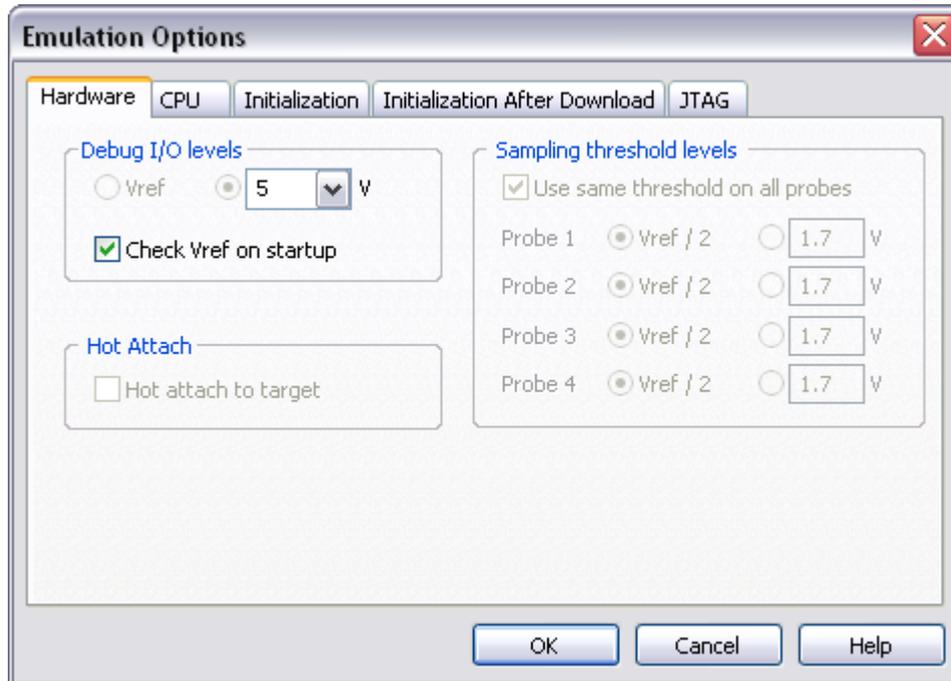
- CPU clock lines X1 and X2 on 78K0
- TOOL0 and TOOL1 or TOOL0 (single wire communication) on 78K0R
- TOOL0 on RL78

Debug Features

- 1 execution hardware breakpoint on 78K0 and no exec. breakpoints on 78K0R and RL78, where breakpoints act post-execute
- Unlimited software breakpoints including in the internal program flash
- 1 or 2 access breakpoints (depending on the device)
- 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.

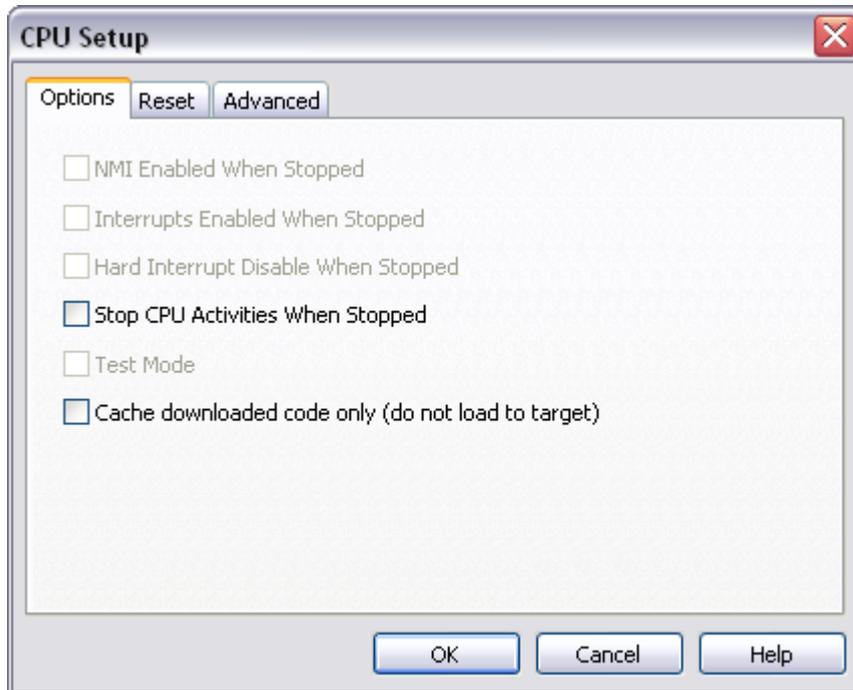
2.2 Initialization Sequence

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. Detailed information may be found in the [Initialization Sequence](#) help topic.

Normally, there is no need to use the initialization sequence with 78K0, 78K0R and RL78 microcontrollers.

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.

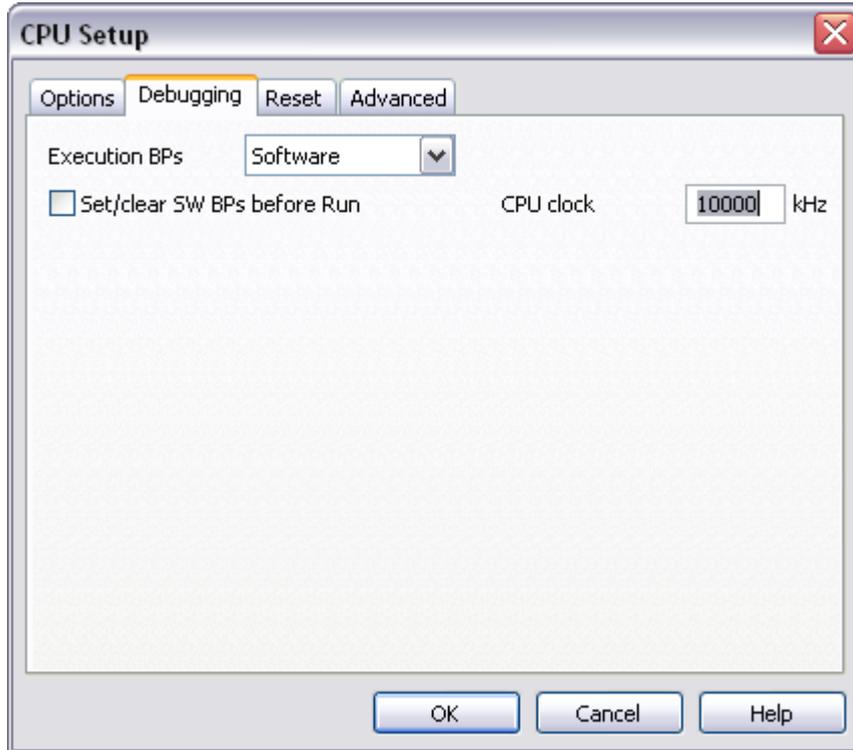
Cache downloaded code only (do not load to target)

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 hardware execution breakpoints. 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.

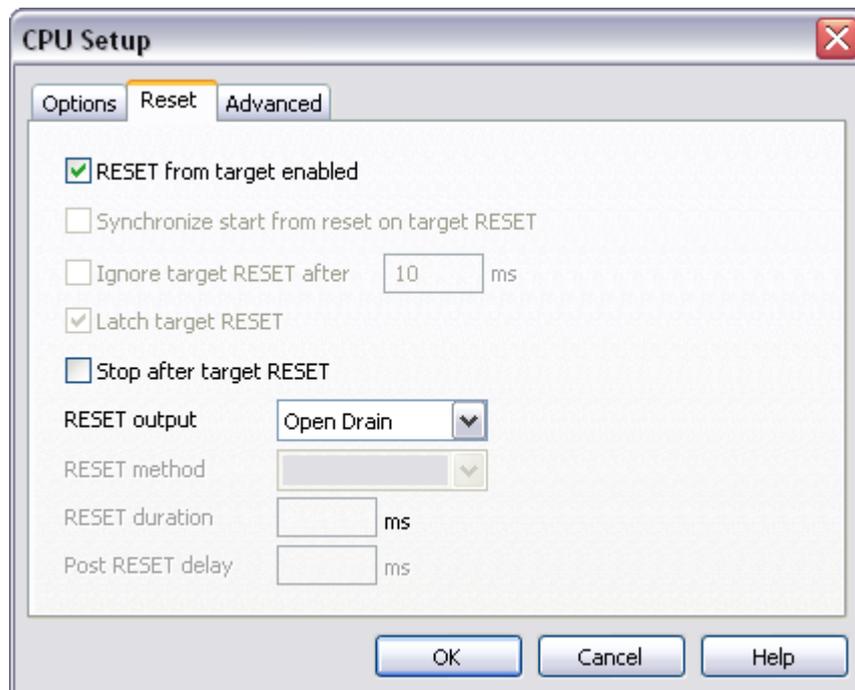
Set/clear SW BPs before Run

When the option is checked, then a software breakpoint is not set/cleared immediately, but is just remembered. Only when the CPU is set to running are the breakpoints committed. This way several breakpoints can be changed but only one re-FLASH operation takes place. This is especially noticeable in testIDEA operation with many stubs and also during a regular debugging session when several breakpoints are set/cleared within the same flash erase block.

CPU Clock

The target microcontroller clock frequency must be specified. This frequency is used by the serial debug interface (X1, X2) and is also required for 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.

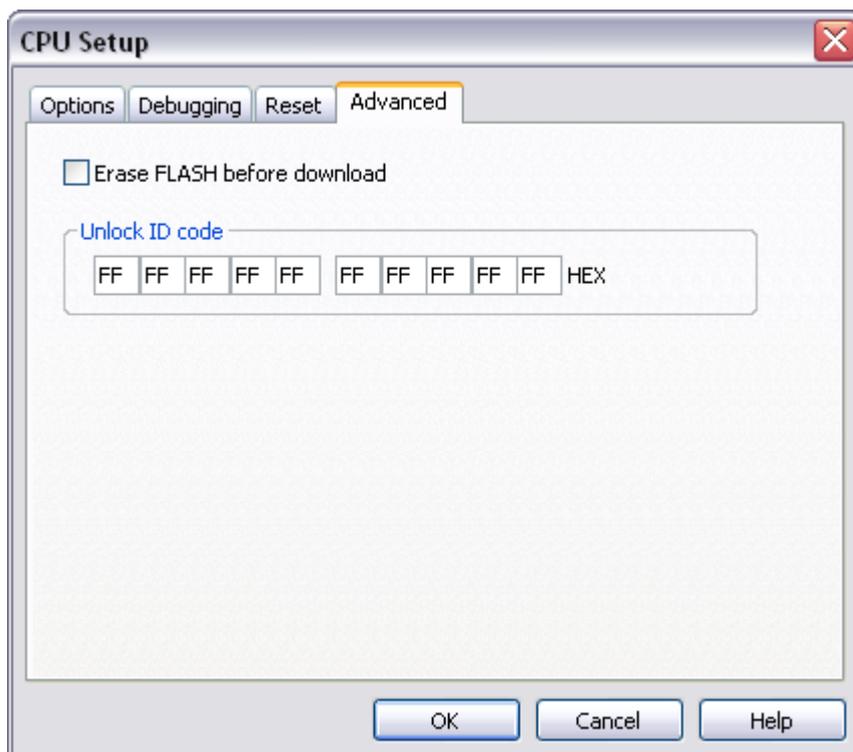
This option is not available on 78K0 family.

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

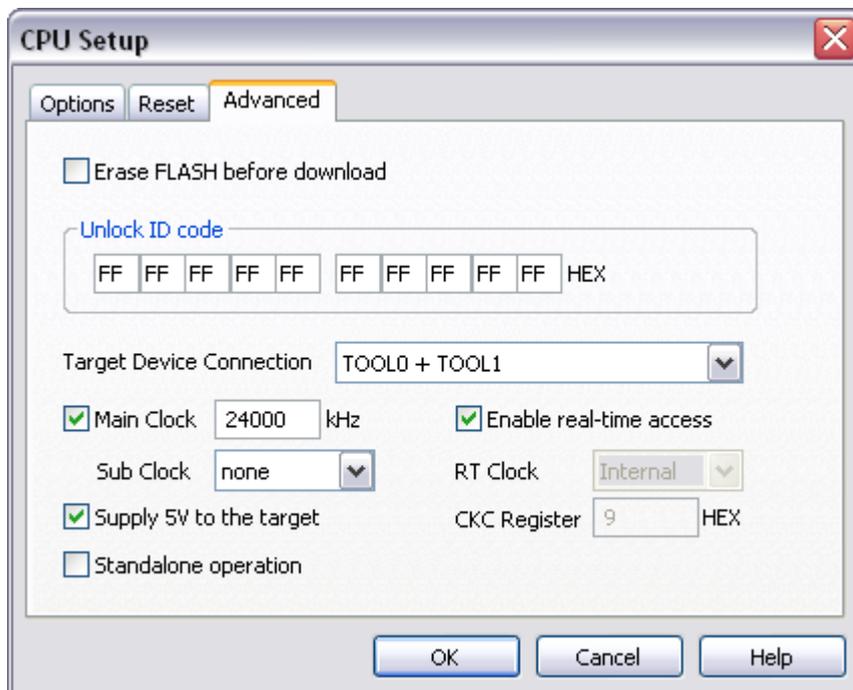
Erase FLASH before download

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

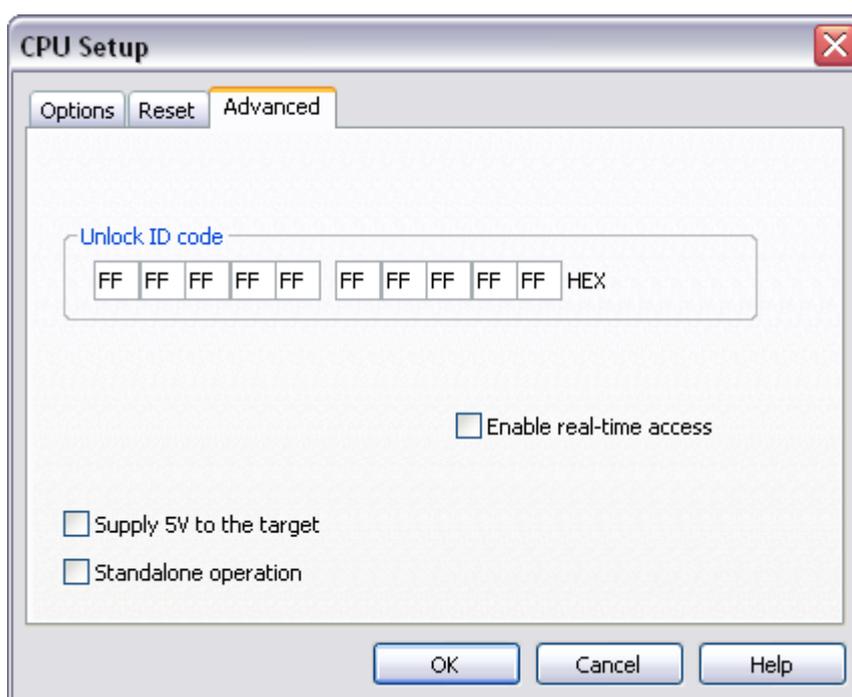
When enabled in winIDEA, emulator supplies 5V at pin 4 (VCC) on debug connector.

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 program and data flash, which are both programmed through the standard debug download. The debugger identifies which code from the download file fits in the program and data flash and then programs it accordingly. Flash programming related settings are in the 'Hardware/Emulation Options/CPU Setup/Advanced' dialog.

Manual debug command 'Mass erase' under 'Hardware/FLASH' performs mass erase on program flash. Data flash is not affected.

Renesas has also built a proprietary EEPROM emulation software layer, which is managed by their FSL libraries, on top of the data flash. Direct access to the "EEPROM emulation" through the debugger is not supported.

5 Emulation Notes

Debugging is based on Renesas OCD monitor, which is loaded into the last block of the internal program flash. For this reason this block is reserved for the debugger.

Additionally, following microcontroller resources are affected during the debugging:

- 4 bytes at address 0x0 (reset and NMI vectors)
- OCD option byte at 0xC3
- 8 bytes at address 0xD0 (jump table)
- 6 bytes of user stack are used by the OCD monitor

Note that performing the Verify Download debug command will yield errors in the listed memory areas.

5.1 Starting the debug session on RL78 with Reset command

If the debug session is started with a reset command (in order to preserve the flash content), debug monitor is expected to be already programmed to flash. If it is present, debug session will be established without any changes to the flash content. In case debug monitor is not previously loaded to the target flash, debug session will fail. In such case new download (with the debug monitor) needs to be performed to establish the debug session.

Note: 78K and 78K0R debug behaves differently – in case reset is issued and the debug monitor is not already programmed, then debug monitor is programmed by the emulator. In this case flash content is not preserved.

6 Real-Time Memory Access

Renesas 78K0 family does not support real-time memory access while 78K0R and RL78 do support real-time access.

Real-time access is implemented through the Renesas OCD monitor implementing memory read and write commands with minimum possible overhead. Nevertheless it's obvious that every real-time access stalls the application for certain amount of time. Additionally, real-time access is operational only when no execution or access breakpoints are set. This is the restriction of Renesas OCD (debug) communication protocol.

If any of the mentioned restrictions is not acceptable for debugging or testing, or it's just an annoying matter, use alternative iSYSTEM ActiveGT POD (ICE) system which doesn't have these restrictions.

In general, it is not recommended to use real-time access for Special Function Registers (SFRs) window. In practice, real-time access still means stealing some CPU cycles. As long as the number of real-time access requests stays low, it shouldn't noticeably affect the application. However, if you update all SFRs or memory window via real-time access, it is most likely that the application behavior will be affected to the extent that it won't run the same way it does without the debugger connected.

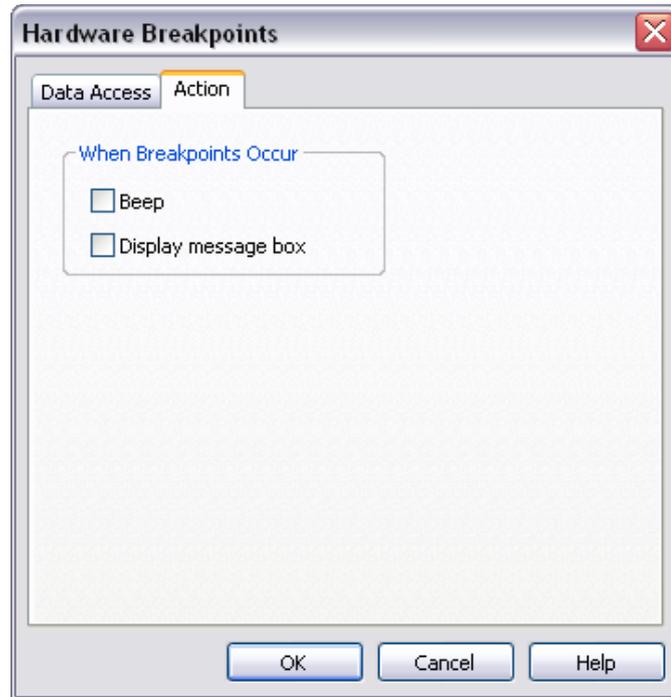
When a particular special function register needs to be updated in real-time, put it in the real-time watch window (don't forget to enable real-time access in the SFRs window but keep SFRs window closed, or opened but with SFRs collapsed). This allows observing a special function register in real-time with minimum intrusion on the application.

Using "alternative" monitor access to update a memory location or a memory mapped special function register while the application is running works like this: the application is stopped, the memory is read and then the application is resumed. Hence the impact on real time

execution is severe and use monitor access for 'update while running' only if you are aware of the consequences and can work with them.

7 Access Breakpoints

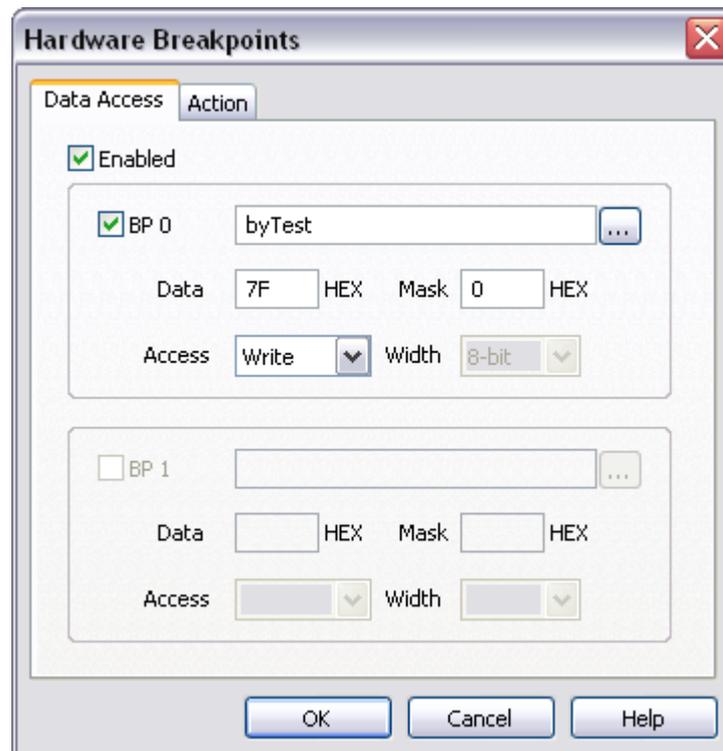
Available access breakpoints configuration is defined by the on-chip debug resources.



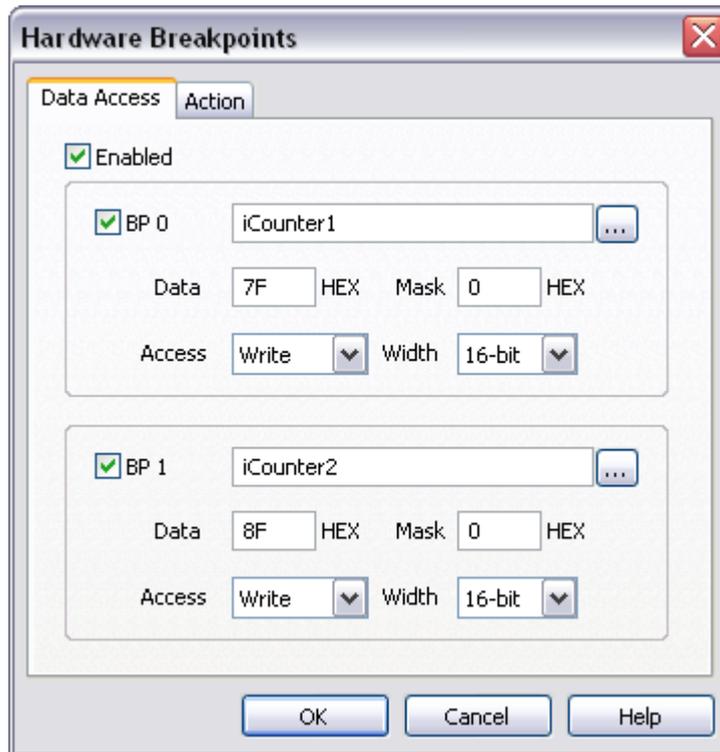
Hardware Breakpoints Configuration – Action tab

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

78K0 devices feature one access breakpoint, which is limited to the 0xF800-0xFFFF address range.



78K0R, RL78/F12 and RL78/G13 devices feature one access breakpoint too while RL78/F13, RL78/F14 and RL78/G14 devices, which additionally feature on-chip trace, have two access breakpoints available.



8 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.

9 Trace

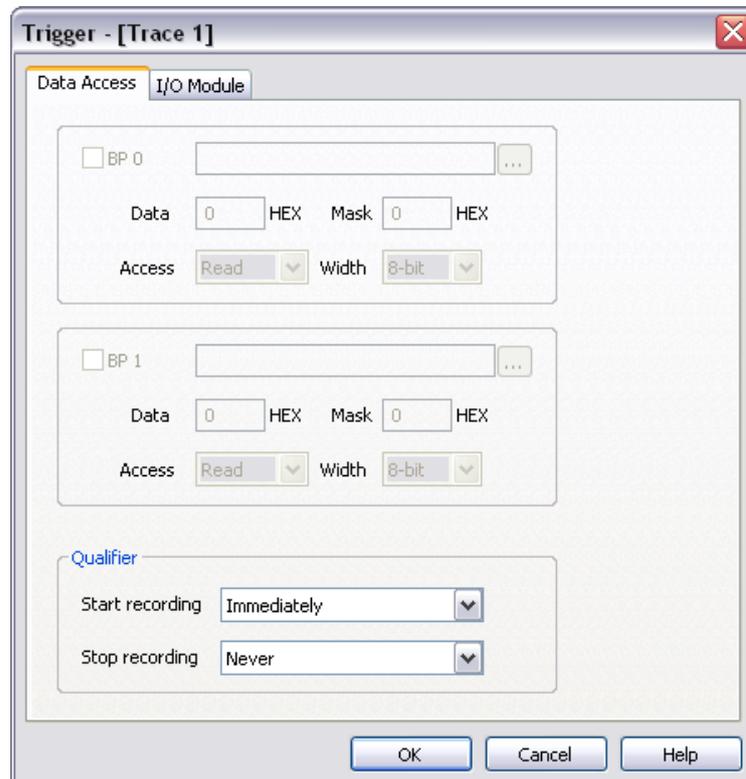
RL78/F13, RL78/F14 and RL78/G14 devices feature a simple on-chip trace without time stamp information, which provides some basic insight in the application behavior besides the regular debug functionalities. It features program trace recording with start and stop event condition and a trace buffer which can capture up to 256 successive “Change of flow” program events.

All sequential instructions between the two program “change of flow” events are inserted by the debugger, which extracts this information from the download file. It is assumed the same file has been also programmed to the microcontroller program flash. ‘Unknown return address’ or ‘Unknown destination address’ can get displayed for a frame in the trace display when under certain conditions the debugger has no means to calculate the right address.

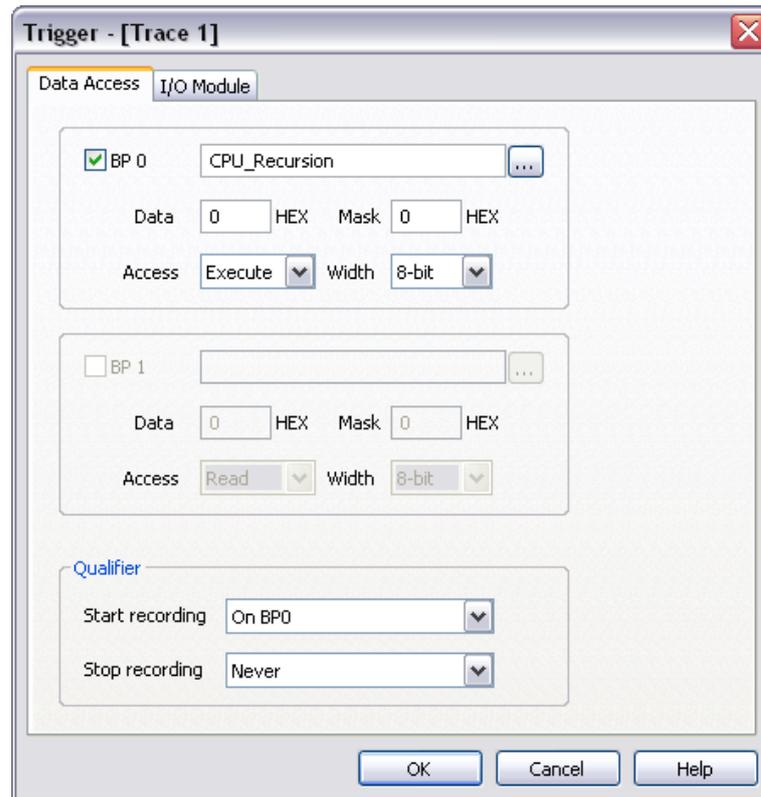
Size of on-chip trace buffer varies among different RL78 devices:

Device series	On-chip trace buffer size
RL78/G14	0x100 “Change of flow” addresses
RL78/F13 64KB flash RL78/F14 64KB flash	0x40 “Change of flow” addresses
RL78/F13 128KB flash RL78/F14 128KB flash RL78/F13 256KB flash RL78/F14 256KB flash	0x80 “Change of flow” addresses

Per default, trace is configured for record immediately and never stop recording, which stops recording after the on-chip trace buffer gets full.



Due to a limited trace buffer size, it is recommended to either define a start recording condition (BP0) or stop recording condition (BP1) in order to capture code execution of interest.



Start and stop recording condition can be a program address or a data access. Note that data accesses are not recorded by the on-chip trace.

Refer to winIDEA Contents Help, [Analyzer Window](#) section (or alternatively to the standalone Analyzer.pdf document) for more details on trace configuration, use and handling.

High end trace features and functionalities like profiler and code coverage are available on iSYSTEM Active PRO/GT PODs (ICE). Please contact iSYSTEM sales for more details on these tools.

10 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.