

Release Notes: LLVM FOR RENESAS RL78 10.0.0.202301

31st of January, 2023

CyberThor Studios Ltd. is releasing the LLVM for Renesas RL78 10.0.0.202301, a cross compiler tool for Renesas RL78 micro-controllers.

SALIENT FEATURES

The LLVM for Renesas RL78 10.0.0.202301 toolchain is based on:

- ❖ LLVM 10.0.0 [released]
- ❖ Compiler-rt 10.0.0 [released]
- ❖ Libcxx 10.0.0 [released]
- ❖ Libcxx-abi 10.0.0 [released]
- ❖ Newlib 4.1.0 [released]
- ❖ GDB 7.8.2 [released]

LLVM RL78 comes with significant performance improvements (both code size and speed) compared to GCC RL78. It also comes with support for latest language standards: full support for C17 and C++17 and experimental support for the C2x (next C standard) and partial support for C++20.

The latest patches are applied to the LLVM sources.



ABOUT LLVM FOR RENESAS RL78 10.0.0.202301

Release Version:	LLVM for Renesas RL78 10.0.0.202301
Release Date:	31st of January, 2023
Platforms Supported:	Ubuntu 18.04 or later (or compatible distribution) Windows 7 or later
Language:	C, C++
Targets:	G23, G1X, I1X, D1X, LIN MCP, F1X, and L1X
Object File Format:	ELF



This toolchain is the successor of GCC RL78 toolchain and it is meant as a direct replacement to GCC RL78. This section describes the fixes made in the LLVM for Renesas RL78 10.0.0.202301 release.

With the implementation of the CC-RL syntax support, a GCC assembler syntax parsing bug was introduced in LLVM 202212.

Due to this bug, the LLVM 202212 release is no longer available for download.

1. [Improvement] Various codegen and runtime optimizations

Code speed and size optimizations were made in the compiler and runtime library.

2. [Improvement] Partial CC-RL ASM expression/directive support (ver. 202212)

Support for most CC-RL assembly expressions, operators and directives was added. Relocatable symbol difference expressions are now supported.

Ex:

```
mov a, #LOW(FOO - BAR)
```

As a part of this change, %lo16() and %hi16() were replaced by LOWW() and HIGHW().

For details please refer to LLVM for Renesas RL78 User Manual Section. 4.1.

3. [Improvement] CC-RL assembler syntax (ver. 202301)

Directives in the CC-RL assembler syntax are no longer case sensitive, both upper and lowercase versions are accepted.

CC-RL syntax directives are no longer enabled by default, they require the -frenesas-extensions commandline option.

4. [Improvement] ABI Compatibility with CC-RL (ver. 202212)

LLVM RL78 aims to be ABI compatible with CC-RL.

For details please refer to LLVM for Renesas RL78 User Manual Section. 4.2.

5. [Bug fix] Fixed asm parser ignoring closing conditional directives (ver. 202301)

After encountering a conditional assembly directive (.if, .ifdef, ...), the parser failed to interpret any other directives that followed, including other conditional directives (.else, .endif, ...), resulting in the "error: unmatched .ifs or .elses" being displayed.

This was fixed in the current release.

6. [Bug fix] Interrupt function and far code interaction (ver. 202301)

Interrupt function declaration and definition merging when using explicit/implicit __far address spaces was fixed.

Far, non-vectorized interrupts are correctly being placed in .textf now.



Notes:

This installer does not provide an option to integrate the LLVM RL78 toolchain with e2 studio, as the e2 studio IDE will automatically detect the LLVM RL78 toolchain installation on start-up for integration. Alternatively, you may use the 'Toolchain Management' feature in e2 studio to achieve this.

For details on e2 studio please visit the following link below:

<https://www.renesas.com/eu/en/software-tool/e-studio>

There is no support in this installer to integrate the toolchain with the HEW IDE.



The following is a list of known issues for the tools we include for the LLVM for Renesas RL78 10.0.0.202301 toolchain:

1. Assembly parsing issue of callt instruction operand.

The assembly parser cannot handle any complex expressions as operands for callt instruction, only constant values are allowed at the moment.

2. Missing assembly listing support (-a[cdhlms] option in GNU AS).

There's no equivalent in the LLVM for the -a[cdhlms] GNU AS option.

The alternative solution is to use llvm-objdump to obtain source code interleaved with assembly (-S, --source option).

3. Binding of references to packed fields.

Creating references to struct members which are declared as packed, e.g. `__attribute__((packed))`, is incorrect as it can cause unaligned access issues. The compiler should return an error in this case however this is not currently the case. This will be fixed in a future release.

4. Inline assembly missing constraints and modifiers support for parameters.

Clang supports, in the case of RL78, inline assembly constructs without parameters, e.g. `asm (AssembleTemplate)`.

The extended syntax available in GCC, `asm (AssemblerTemplate : OutputOperands : InputOperands : Clobbers : GotoLabels)`, is not currently supported. This will be fixed in the next release, however the constraints will not be the same as in GCC.

5. Far address space handling in C++ code

Currently far data, far rom and far code handling in C++ are not supported.

6. Newlib format specifiers

The pre-built newlib library included with the toolchain is built without using the `-enable-newlib-io-c99-formats` configure flag.

Without this flag, handling for some format specifiers will not be included in the resulting library.

Using libgen, users can build their own newlib, specifying the `-D_WANT_IO_C99_FORMATS=1` option to include the extra format specifiers, at the cost of code size.



7. Other issues, non-specific to RL78

LLVM tries to be a complete replacement. As such there are still a couple of missing features from GCC which will be implemented in future releases. In particular, the following issues should be noted:

lld: Information printed using `--print-gc-sections` is not as nice as when using the GNU ld. [Bug 46783](#)

llvm-ar: Errors when printing multiple members with the same name. [Bug 42521](#)

llvm-dwarfdump: Does not print section attribute flags yet. [Bug 38488](#)

llvm-nm: Unable to understand symbols built with gcc-lto [Bug 41437](#)

llvm-nm: Needs support for `--line-numbers` to `llvm-nm` [Bug 40001](#)

llvm-objcopy: Unknown argument '`--change-section-address`'. [Bug 45217](#)

llvm-objcopy: Objcopy zero-size section might cause huge binaries. [Bug 46299](#)

llvm-objdump: Prints wrong line number info for obj file compiled with `-ffunction-sections`. [Bug 40703](#)

llvm-objdump: Wrong behavior for non-relocatable objects when using `llvm-objdump` with `-r` option. [Bug 41901](#)

llvm-readobj: Make GNU style symbol printing invalid symbol section indexes match GNU `readelf` [Bug 43850](#)

llvm-readelf: Relocation addends printed style does not match GNU `readelf` [Bug 45235](#)

llvm-string: Short option with argument grouping not GNU compatible [Bug 42942](#)

llvm-string: Allow "`-<integer>`" as an alias for "`-n <integer>`" [Bug 42964](#)

llvm-symbolizer: Shows incorrect source line info if `--gc-sections` used [Bug 41124](#)

llvm-symbolizer: `llvm-addr2line` does not exit when passed a non-existent file [Bug 42754](#)



8. Other issues

Finally, for better understanding regarding the status of the toolchain please visit <https://bugs.llvm.org/> . In particular, the following queries will help better understand the status of each tool.

https://bugs.llvm.org/buglist.cgi?bug_status=UNCONFIRMED&bug_status=NEW&bug_status=CONFIRMED&bug_status=REOPENED&component=ELF&product=lld&query_format=advanced&resolution=---

https://bugs.llvm.org/buglist.cgi?bug_status=UNCONFIRMED&bug_status=NEW&bug_status=CONFIRMED&bug_status=REOPENED&component=llvm-ar&product=tools&query_format=advanced&resolution=---

https://bugs.llvm.org/buglist.cgi?bug_status=UNCONFIRMED&bug_status=NEW&bug_status=CONFIRMED&bug_status=REOPENED&component=llvm-dwarfdump&product=tools&query_format=advanced&resolution=---

https://bugs.llvm.org/buglist.cgi?bug_status=UNCONFIRMED&bug_status=NEW&bug_status=CONFIRMED&bug_status=REOPENED&component=llvm-nm&product=tools&query_format=advanced&resolution=---

https://bugs.llvm.org/buglist.cgi?bug_status=UNCONFIRMED&bug_status=NEW&bug_status=CONFIRMED&bug_status=REOPENED&component=llvm-objcopy%2Fstrip&product=tools&query_format=advanced&resolution=---

https://bugs.llvm.org/buglist.cgi?bug_status=UNCONFIRMED&bug_status=NEW&bug_status=CONFIRMED&bug_status=REOPENED&component=llvm-objdump&product=tools&query_format=advanced&resolution=---

https://bugs.llvm.org/buglist.cgi?bug_status=UNCONFIRMED&bug_status=NEW&bug_status=CONFIRMED&bug_status=REOPENED&component=llvm-readobj&product=tools&query_format=advanced&resolution=---

https://bugs.llvm.org/buglist.cgi?bug_status=UNCONFIRMED&bug_status=NEW&bug_status=CONFIRMED&bug_status=REOPENED&component=llvm-size&product=tools&query_format=advanced&resolution=---

https://bugs.llvm.org/buglist.cgi?bug_status=NEW&bug_status=CONFIRMED&bug_status=REOPENED&component=llvm-symbolizer&product=tools&query_format=advanced&resolution=---



FREE SUPPORT FOR LLVM FOR RENESAS RL78 10.0.0.202301

For free technical support, please register at
<https://llvm-gcc-renesas.com>

For your feedback and suggestions, please visit
<https://llvm-gcc-renesas.com/help/contact-us/>

