

## Release Notes: LLVM FOR RENESAS RL78 10.0.0.202206

30th of June, 2022

CyberThor Studios Ltd. is releasing the LLVM for Renesas RL78 10.0.0.202206, a cross compiler tool for Renesas RL78 micro-controllers.

### SALIENT FEATURES

The LLVM for Renesas RL78 10.0.0.202206 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 10.0.0.202206

Release Version:	LLVM for Renesas RL78 10.0.0.202206
Release Date:	30th of June, 2022
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 10.0.0.202206 release.

### **1. [Improvement] Various library and codegen optimizations**

Code speed and size optimizations were made both in the included libraries and in the compiler itself.

### **2. [Improvement] Added support for additional character encodings**

The compiler accepts the `-finput-charset=`/`-fexec-charset=` options with all the encodings supported by the libiconv library.

### **3. [Change] libg.a was removed**

Since libg.a was a copy of libc.a, it was removed from the multilib supplied with the toolchain.

### **4. [Change] Using variable length arrays(VLA) will issue warnings**

The current release of LLVM for RL78 will issue warnings when variable length arrays are used. In a future release, usage of VLA-s will be disabled.

### **5. [Bug Fix] Build fails when the resulting command line length is over a certain limit**

Even when using @command files, linking would fail if the user tried to link too many files, with long filenames.

Current release fixes this, allowing the usage of command files to bypass the command line length limitations.

### **6. [Bug Fix] Warnings generated using -mfar-rom option**

When using -mfar-rom the linker will no longer generate warnings like "warning: incompatible rodata area flags...".

The following functions had their far rom equivalents added: `bsearch` `atol` `atof` `atoff` `atoi` `atoll` `gets` `perror` `puts` `qsort` `strtod` `strtof` `strtoull` `strtoul` `strtoll` `strtol` `strtoumax` `strtoimax` `strcspn` `strtold`

Please note that in order to call these -mfar-rom versions, users must include the corresponding headers (`stdlib.h`, `stdio.h`, `inttypes.h` or `string.h`).

### **7. [Bug Fix] When using typedefs of functions and mfar-code together, address space qualifiers are rejected**

In the previous release, the compiler did not allow changing a function type's address space if it was defined in a typedef with the default address space.



This is fixed in the currently release, resulting the following code being accepted when using the -mfar-code option:

```
typedef void(FT) (void);  
__near FT * f_np = (__near FT *)0xabcd;  
__far FT * f_fp;
```

**8. [Bug Fix] Address space of a function is decided at first declaration, silently ignoring later definitions having different address space**

The compiler now correctly issues an error if a function is re-declared or defined with a different address space.

**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 toolchain with the HEW IDE.



The following is a list of known issues for the tools we include for the LLVM for Renesas RL78 10.0.0.202206 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. Assembly parsing of %lo16, %hi16 limited support.**

When using %lo16, %hi16 operators only 2 basic operations are currently allowed, symbol + constant and symbol - constant.

For example:

```
.short %lo16(_my_symbol) + %lo16(my_const)
```

The parse will be enhanced in future versions to support more complex expressions.

### **3. Assembly parsing missing operand.**

There is currently no assembly operand to extract 8-bit data from symbols in order to be used with MOV ES, #byte, MOV CS, #byte etc. Only immediate values can be used with those instructions at the moment. The current workaround is to use %lo16, %hi16 to set the value in 16-bit register and then move the relevant part in ES/CS. A new operand will be implemented in next release.

### **4. 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).

### **5. Unsigned bitfields not supported.**

Currently, there is no support for unsigned bitfields in LLVM. This will be implemented in a future release.

### **6. 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.



## 7. llvm-objdump missing features

llvm-objdump does not currently support -i and -g options supported by r178-elf-objdump. This will be fixed in a future release.

## 8. Inline assembly missing constraints and modifiers support for parameters.

Clang supports, in 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.

## 9. 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](#)

## 10. 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=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-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-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-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-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-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-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=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=---](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 10.0.0.202206**

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/>

