

RELEASE NOTES: GNURL78 v15.01

2nd April 2015

KPIT Technologies Limited is releasing the GNURL78 v15.01, a cross compiler tool for Renesas RL78 micro-controllers.

SALIENT FEATURES:

1. The GNURL78 v15.01 toolchain is based on
GCC 4.8.1 [snapshot dated 10th December 2014],
Binutils 2.23 [snapshot dated 10th December 2014],
Newlib 2.0.0 [snapshot dated 10th December 2014] and
GDB 7.8.2 [released].
2. The GNURL78 toolchain comes with bug fixes and more code size optimization.
3. The GNURL78 toolchain now supports new command line options:
-mcpu=g10, -mcpu=g13 and -mcpu=g14 to differentiate the S1, S2 and S3 cores respectively.

ABOUT RL78 v15.01:

Release Version:	GNURL78 v15.01
Release Date:	2 nd April 2015
Platforms Supported	Red Hat GNU/Linux v8.0 or later (or compatible distribution) Windows XP, Windows 7, Windows 8
Language:	C, C99, C++
Targets	G1X, I1X, D1X, LIN MCP, F1X, and L1X
Object File Format	ELF

CHANGES IN THIS RELEASE:

This section describes the enhancements made and the issues fixed in this release.

GCC/Binutils:

1. The GNURL78 toolchain now generates the DIVHU and DIVWU instructions for the S3 core (e.g., G14/F13/F14/G1G targets) with "-mcpu=g14" option. The following compiler options are added to differentiate instructions generation for S1, S2 and S3 cores:

'-mcpu=g10' - This supports the generation of software emulation for S1 cores.

'-mcpu=g13' - This supports the generation of hardware divide peripheral instructions for S2 cores.

'-mcpu=g14' - This supports the generation of instructions DIVHU and DIVWU for S3 cores.

The compiler accepts -mcpu=g10, -mcpu=g13 and -mcpu=g14 as command line options.

The -mg10, -mg13, and -mrl78 options are still supported, although they are now aliases for the -mcpu= options.

2. The GNURL78 toolchain generated large code for manipulating the stack pointer when a large local variable was declared.

This issue has been fixed.

3. The GNURL78 toolchain generated large exception handling code by default for the C++ projects when a call-by-reference was used. The GNURL78 toolchain is now configured to eliminate the generation of exception handling (-fno-exceptions) code by default to make the behaviour consistent across the other GNU tools (e.g., GNURX and GNUARM-NONE - which are based on GCC 4.8.3 or higher)

4. The option '-mrelax' along with optimization option '-O2' caused execution of certain applications to fail on simulator and hardware.

This issue has been fixed.

5. The GNURL78 toolchain generated segmentation fault when '-mrelax' was used along with optimization options '-O1' or '-Os' for certain test cases.

This issue has been fixed.

6. The GNURL78 toolchain optimizes the access to SFR register which is marked as volatile for multiple access to same location for certain test cases. This causes incorrect hardware behaviour when the same SFR is modified in the interrupt routine during execution.

This issue has been fixed.

7. The GNURL78 toolchain generates an internal compiler error (ICE) when the option '-fcall-used-r0' is used for registers r0 to r7 or 'a' to 'l' (ax, bc, de, hl) along with optimization '-O2' for a specific test case.

This issue has been fixed.

8. The GNURL78 toolchain generated incorrect opcode for index addressing i.e., the assembler parser generated NOP instructions and therefore causing a mismatch between the list file and the objdump (dis-assembly) for certain assembly test cases as shown below:

```

/*test.s*/
.text
addw    ax, [hl+0]
addw    ax, [hl]
.end

$rl78-elf-gcc -c -x assembler-with-cpp test.s -g -Wa,-ahl=test.lst

0000 61 09 00    addw ax, [hl+0]
0003 61 09 00 00  addw ax, [hl] ---> expected 3 bytes opcode

```

This issue has been fixed by eliminating the generation of 'NOP' instructions.

9. The GNURL78 objdump utility generated incorrect disassembly for index addressing such as below for certain test cases: list file displayed "a, [hl+0]", while the object dump showed only "a, [hl]", Here "+ 0" was omitted by the toolchain.

This issue has been fixed for following instructions:

ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, XCH. MOV, MOVW, CMPS, ADDW, SUBW, CMPW, INC, DEC, INCW and DECW instructions.

10. The GNURL78 toolchain generated incorrect disassembly for 'sp' register i.e., the opcode 'mov sp, ax' was disassembled as 'mov 0xffff8, ax'.

This issue has been fixed.

11. The GNURL78 assembler generated wrong location counter for the below test case:

```

/*test.s*/
.org 0x1234
0: .short 0b, 0f @ 0:
1: .short 1b, 1f @ 1:
2: .short 2b, 2f @ 2:
.end
//$ rl78-elf-as test.s -a

```

This issue has been fixed.

12. The GNURL78 toolchain supports a new linker option, '--warn-orphan'. This option tells the linker to generate a warning message whenever it has to place an orphan section into the output file.

e.g., for the below test case, linker did not generate any warning for the 'RL78_DESC' section, even though it was not defined in the linker script,

```
int test_array[4]__attribute__((section("RL78_DESC")))= {1,2,3,4};
int main (void)
{
    return 0;
}
```

13. The following test case generated internal compiler error (ICE) for default compiler options,

```
#include <stdio.h>
typedef unsigned long int uint32_t;
uint32_t `__far` * ptr1;
int main (void)
{
    printf ("ptr1 is %lx\n", (char * `__far`) ptr1);
    return 0;
}

$rl78-elf-gcc test.c
```

This ICE has been fixed. Additionally, the compiler generates the following warning on passing the '-Waddress' option, when user casts a far pointer to a near pointer:

```
"warning: converting far pointer to near pointer [-Waddress]"
```

14. The GNURL78 toolchain generates incorrect prologue/epilogue for interrupt service routine and does not save/restore the 'es' register. This issue is observed when a ' __far' pointer is used inside interrupt service routine.

This issue has been fixed.

15. The GNURL78 toolchain generates an internal compiler error when the ' __builtin_rl78_clr1' or ' __builtin_rl78_set1' function tries to set the bit 0.

```
char reg1;
void test0()__attribute__((noinline));
void test0()
{
    __builtin_rl78_clr1((char *)&reg1, 0);
    __builtin_rl78_set1((char *)&reg1, 0);
}
```

This issue has been fixed.

16. The RL78 Application Binary Interface (ABI) document for GCC is now updated for G10 variants at, <http://www.kpitgnutools.com/manuals/RL78-ABI-Specification.html>

Libraries:

1. The GNURL78 toolchain supports multi-libs for the options such as '-mg14' and '-mg13' in addition to previously supported '-mg10' variants.
2. The GNURL78 toolchain was setting incorrect value for the macros 'LDBL_MANT_DIG' and 'LDBL_MAX_EXP' when optlib was used. i.e., the value of 24 and 128 was set respectively instead of 53 and 1024. This value should be updated in Optlib, based on the option, '-m64bit-doubles' or '-m32bit-doubles' as in case of Newlib libraries.

This issue has been fixed.

KNOWN LIMITATIONS:

This section describes the known limitations in this release. We intend to fix these limitations in our future releases.

1. The GNURL78 toolchain calculates wrong higher 8/16 bits for operation hi8/hi16 respectively for below test case:

```
int main ()
{
    asm ("MOVW HL, #(%hi16(0x0EFFE2))"); // FAILS => FFE2 in place of 0EFF
    asm ("MOVW AX, #(%hi8(0x0EFFE2))"); // FAILS => E2 in place of 0E
    asm ("MOVW HL, #(%lo16(0x0EFFE2))"); // Work as expected => FFE2
    return 0;
}
```

2. The GNURL78 toolchain does not handle relocation related calculations when a combination of different data types is used, for example
`.hword %hi16(0xF0A80) + %lo16(Sym1)`
3. The GNURL78 Assembler does not check the ranges of relative jumps performed by the 'br' instruction correctly. The RL78 processors allow 8-bit and 16-bit relative branches (so -128/+127 or -32768/+32767 bytes from PC) but gas seems to be calculating the distance using unsigned values and therefore allowing a forward jump of, say, 200 bytes without complaint which actually ends up as a backwards jump of 56 bytes (i.e. -56). A build-time error only occurs once the distance is too large to fit in an 8 or 16-bit value (as appropriate).

For Windows OS only:

1. The GNURL78 v14.03 Installer, "Default Installation" mode will install the toolchain to a default location "C:\Program Files\KPIT".
The "Custom Installation" mode allows user to install toolchain at custom directory.
User will not be prompted for IDE selection page while installing a toolchain. Instead, the e2 studio IDE will automatically detect the GNURL78 v14.03 toolchain on start-up for registration. You may also use the Toolchain Management feature in e2 studio IDE to achieve this.
2. In-order to integrate the toolchain with Renesas e2 studio version 2.0 and above, please use the Renesas toolchain management feature in e2 studio IDE ('Help' > Add Renesas Toolchains')
3. The registry entry for Windows-7 64-bit system differs to Windows-7 32-bit system.

NOTE:**WINDOWS and GNU/LINUX:**

1. The optimized libraries provided along with the Newlib libraries in the toolchain do not require a separate download.
2. The optimized libraries ('liboptm.a' and 'liboptc.a') are not provided under GNU GPL. The source code of these optimized libraries is neither released nor available on request.
3. The 'libgen' utility is not provided under GNU GPL. The source code of the "libgen" utility is neither released nor available on request.

For free technical support, please register at <http://www.kpitgnutools.com>

For your feedback and suggestions, please visit <http://www.kpitgnutools.com/feedback.php>.