

**Release Notes: GNUARM-NONE-EABI v16.01**

31<sup>st</sup> March 2016

KPIT Technologies Limited is pleased to release the GNUARM-NONE-EABI v16.01, a cross compiler toolchain for ARM based Renesas RZ micro-controllers.

The GNUARM-RZ toolchain has been superseded by GNUARM-NONE-EABI toolchain. The functionality is same as that of GNUARM-RZ, this is done to maintain the universal naming convention for ARM toolchains.

**SALIENT FEATURES**

1. The GNUARM-NONE-EABI v16.01 toolchain is based on,

GCC 5.2.1 [snapshot dated 2<sup>nd</sup> December 2015],

Binutils 2.25.90 [snapshot dated 17<sup>th</sup> December 2015],

Newlib 2.2.0 [released] and

GDB 7.8.2 [released].

2. The GNUARM-NONE-EABI v16.01 toolchain provides multi-lib for Cortex-A/R/M targets.

Please refer to the following link for the correct command line options to be passed to GCC/G++ for the Cortex-A/R/M architecture variants,

<http://www.kpitznutools.com/phpmyfaq/index.php?aktion=artikel&rubrik=013001&id=553&lang=en>

3. This toolchain has been tested for Cortex-A9 target only.  
(**Note:** Toolchain has been tested only for 'little' endian mode.)

**ABOUT GNUARM-NONE-EABI v16.01**

Release Version:	GNUARM-NONE-EABI v16.01
Release Date:	31 <sup>st</sup> March 2016
Platforms Supported:	Red Hat GNU/Linux v8.0 or later (or compatible distribution), Windows 7, Windows 8, Windows 10.
Language:	C, C99, C++
Targets:	Cortex-A9, Cortex-R4 and Cortex-M3
Object File Format:	ELF

## CHANGES IN THE GNUARM-NONE-EABI v16.01 RELEASE

This section describes the enhancements made and the issues fixed in the v16.01 release.

### GCC:

1. The GNUARM-NONE-EABI v16.01 toolchain adds basic support for the ARMv8 Baseline (armv8-m.base) and Mainline (armv8-m.main) architectures.
2. The 'armv7e-m' architecture is now provided with additional multilibs for floating point FPU support, 'fpv5-d16' and 'fpv5-sp-d16'.

Also, the cortex-m7 core is mapped to use 'armv7e-m' multilib and the previously existing separate multilib 'cortex-m7' directory has been removed.

### LIBRARIES:

1. The GNUARM-NONE-EABI libgen utility is now enhanced to support dynamic building of libgcc library.

For example, following command can be used to build optlib and libgcc dynamically during project build:

```
$arm-none-eabi-libgen --select-lib=optlib,libgcc --header-files=all --compiler-
options=-mcpu=cortex-a9,-march=armv7-a,-mlittle-endian, -mthumb-interwork,-mfloat-
abi=soft,-mfpu=vfp,-Os,-ffunction-sections, -fdata-sections,-fno-function-cse,-
funit-at-a-time,-falign-jumps --assembler-options=-mcpu=cortex-a9,-march=armv7-a,-
mlittle-endian, -mthumb-interwork,-mfloat-abi=soft,-mfpu=vfp -o "libProjectName.a"
```

The compiler options mentioned above will also be passed while building libgcc. Two libraries will be generated in the current working directory i.e., the libProjectName.a and the libgcc.a.

Similarly the newlib and libgcc libraries can also be built.

Note that, some warnings are observed on the console while building libgcc dynamically, similar warnings are also observed while statically building libgcc as part of toolchain build process.

2. The GNUARM-NONE toolchain generated incorrect values for the macros UINT\_MAX and INT\_MAX when optlib libraries were used.

This issue has been fixed.

### INSTALLER:

1. The link to GNUARM ABI (Application Binary Interface) is made available on <http://www.kpitgnutools.com> website and also provided along with Linux RPM and Windows installer.

## KNOWN LIMITATIONS IN GNUARM-NONE-EABI

This section describes the known limitations in this release. We intend to fix these issues in our future releases. We occasionally release maintenance packs for critical bug fixes.

### WINDOWS AND GNU/LINUX:

1. Assembler does not support ARM architectures 'armv5e' and 'ep9312'.
2. For following CPUs, compiler generates an instruction 'bx lr'.  
arm2, arm250, arm3, arm6, arm60, arm600, arm610, arm620, arm7, arm70, arm700, arm700i, arm710, arm7100, arm710c, arm720, arm7500, arm7500fe, arm7d, arm7di, arm7dm, arm7dmi, arm7m. Assembler does not support this instruction for these CPUs, resulting in application build failure with assembler errors.  
Similar problem has been observed with the following architectures as well:  
armv2, armv2a, armv3, armv3m.

### NEWLIB:

1. The 'C' library functions, 'strcat' and 'strncat' do not generate correct results with GDB simulator.

### GDB:

1. For 'big endian' targets debugger fails to start the debugging process.
2. For 'thumb' mode:
  - a. Debugger shows incorrect values for local and global variables.
  - b. While stepping over the multiplication operation, debugging process goes in an infinite loop.
  - c. While debugging floating point operations, execution halts with an error message:

```
Unhandled v6 thumb insn: 4603
0x00000004 in ?? ()
```

3. On using float-abi 'hard':
  - a. Debugging process halts with an error message:  
0x00000004 in ?? ()  
Cannot find bounds of current function

### GDB-SIMULATOR:

1. The run simulator fails to simulate the 'mul' instruction correctly when destination register is identical to the first operand.

### LIBGEN:

1. While building the Newlib library using 'libgen' tool, many compiler warnings are generated by the 'libgen' tool. These warnings are displayed on 'stdout' while building the runtime libraries. These warnings are observed while building the pre-built libraries as well.

**TOOLS INFORMATION**

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>