GNU tools for STM32 patch list

Patch	Description	Affected component	Category	Use case	Affects target binary?
Fix for long path issues on Windows	Windows has a limit of the number of characters in paths to files. This fix allows up to 248 characters in paths to GCC tool chain binaries and up to 4096 chars for all files processed by the GCC tools. Without the patch the latter limit is about 150 characters.	gcc, make, busybox	OS limitation	Run gcc, make, busybox	No
Provide newlib string function compatible with all platforms	Adds aliases for **newlib** string functions. Enables the functions to be called on all target platforms without changing the target source code. Useful for unit testing of target source code on Windows.	newlib	Target platform compatibility	Execution and unit test of target binary	No
Provide compatibility with IAR EW projects	Adds pre-processor symbol __FILENAME__ which is used in IAR EW. Will be required for import of IAR EW projects.	gcc	IAR EW project compatibility	Import of IAR EW project	No
Enable debugging of functions in target libraries libg or libg_nano	Updates the GCC build scripts for libg and libg_nano in **newlib**, so that debug symbols are not stripped.	newlib	Debug limitation	Debug of target binary	No
Correct stack usage for functions with inline assembler	Required by Stack Analyzer advanced debug function in CubeIDE.	gcc	Debug limitation	Debug of target binary	No
Reduce newlib code size by 10-30%	Updates the GCC build scripts for **newlib** to use -Os instead of -O2. Beneficial in most embedded projects.	newlib	Code size	Build and load target binary	Yes, reduced flash size
Enable user config of malloc() pagesize in newlib	Provides the ability to set the page size used when allocating memory in malloc(). Done by implementing sysconfig. Without the fix, the default page size is 4 Kbyte which may consume a lot memory in some applications. Applies to the build of the C standard library **newlib**.	newlib	Data size	Build and execute target binary	Yes, reduced RAM size

Prepare for calculation of cyclomatic complexity	Provides the ability to calculate cyclomatic complexity of the target source code processed by GCC. The patch is available in the GCC code base. It is a preparation for future added functionality in CubeIDE.	gcc	Functional enhancement	Calculations of cyclomatic complexity of target source code	No
Fix for "Empty if - statements" warning	Eliminates warning emitted for an empty if- statement	gcc	Warning	Build gcc	No
Fix for “Missing function prototype” warning	Eliminates warning by adding missing function prototypes.	gcc	Warning	Build gcc	No
Fix for "Unused arguments" warning in __GLIBCXX_THROW_OR_ABORT__	Eliminates warning in cases were the arguments to __GLIBCXX_THROW_OR_ABORT__ are not used.	gcc	Warning	Build gcc	No
Fix for "Set but not used" warning	Eliminates warning for variables that are initialized but not further referenced.	gcc	Warning	Build gcc	No
Fix for "Maybe used uninitialized" warning	Eliminates warning for variables that are not initialized before first use.	gcc	Warning	Build gcc	No
Removed single character argument for functions	Fixes: __isinff, __isinfd, __isnanf, __isnand, __fpclassifyf, __fpclassifyd, __signbitf, __signbitd, imaxabs, _getwchar_r, _getwchar_unlocked_r	newlib	libstdc++ naming rules enforcement	Test suite	No
Renamed single character member of struct __tzrule_struct	From m to *month*, from n to *week*, from d to *day* and from s to *secs*	newlib	libstdc++ naming rules enforcement	Test suite	No