

liblepto - Feature #367

log: implement IDebugAssert

22.02.2023 12:26 - Maximilian Seesslen

Status:	Erledigt	Beginn:	22.02.2023
Priorität:	Normal	Abgabedatum:	
Zugewiesen an:	Maximilian Seesslen	% erledigt:	0%
Kategorie:		Geschätzter Aufwand:	0.00 Stunde
Zielversion:	v0.2.0	Aufgewendete Zeit:	0.00 Stunde
CS Zielversion:			
Beschreibung			
Biwak classes should be able to indicate errors in debug mode that may be handled in application layer later.			
<pre>lDebugAssert(sta == 2, "Could not write (%d)", sta);</pre>			
• remove lAssertSta(); it can be done in an single macro, see below. • implement pretty-print, see below. • Only use lLog and lLogImpl; others are macros • Implement lLogImpl in biwak • Used macros: LEPTO_LOG_PRETTY_PRINT, FLASH_SIZE, LEPTO_LOG_PRETTY_PARAMS, __FILENAME__, Ansi-Codes, LEPTO_LOG_DEBUG, LEPTO_LOG_DEBUG_ASSERT • Use different function names for pretty-print to force an linker error when configurations of compile units do not match • adjust documentation • measure 'cost' of feature			
So lepto should also set __FILENAME__			
<pre>set (CMAKE_CXX_FLAGS "\${CMAKE_CXX_FLAGS} -D__FILENAME__='\"\$(subst \${CMAKE_SOURCE_DIR}/,,\$(abspath \$<))\"'")</pre>			
Put the pretty logic into biwak cmake file in order not to put biwak logic into lepto:			
<pre>set (MCU_FLASH_SIZE 32768) if(MCU_FLASH_SIZE GREATER_EQUAL 65536) add_definitions(-DLEPTO_LOG_PRETTY_PRINT) endif(MCU_FLASH_SIZE GREATER_EQUAL 65536)</pre>			

Historie

#1 - 22.02.2023 12:27 - Maximilian Seesslen

```
#include <QCoreApplication>
#include <main.hpp>
#include <stdarg.h>

//LEPTO_LOG_DEBUG_ASSERTIONS
#define FLASH_SIZE 65536

#if FLASH_SIZE >= 32768
#define LEPTO_LOG_PRETTY_PRINT
#endif

#if defined LEPTO_LOG_USE_PRETTY_PRINT
    #define PRETTY_PARAMS    const char*file, int line,
    #if ! defined __FILENAME__
        #define __FILENAME__ __FILE__
        #warning "__FILENAME__" is not defined. Implement it in CMake to save memory.
    #endif
#else
```

```

#define PRETTY_PARAMS
#endif

void lDebugImpl( PRETTY_PARAMS const char *format, ... )
{
    va_list list;
    va_start(list, format);

#if defined LEPTO_LOG_USE_PRETTY_PRINT
    printf("%s(%d): ", file, line);
#endif

    vprintf( format, list);
    printf( "\n" );
    return;
}

#if defined LEPTO_LOG_USE_PRETTY_PRINT

// #define lDebugAssert(a, b, ...) if(! (a)){lDebugImpl(__FILENAME__, __LINE__, "E: Assertion '" #a "' not true. "
. " b , ##__VA_ARGS__);}

#define lDebugAssert(a, ...) if(! (a)) { lDebugImpl(__FILENAME__, __LINE__, "E: Assertion '" #a "' not true. "
__VA_ARGS__ ) ; }
#else
#define lDebugAssert(a, ...) if(! (a)){lDebugImpl("E: Assertion '" #a "' not true. " __VA_ARGS__ );}
#endif

#define lAssert( assertion, ... ) if ( ! (assertion) ) \
    lFatal( "Assertion wrong" __VA_ARGS__ );

int main(int argc, char *argv[])
{
    QCoreApplication app(argc, argv);
    int sta=1;
    lDebugAssert( sta == 2, "Could not write (%d)", sta);
    lDebugAssert( sta == 2 );

    return (0); //app.exec();
}

```

#2 - 22.02.2023 12:47 - Maximilian Seesslen

- Beschreibung aktualisiert

#3 - 22.02.2023 12:48 - Maximilian Seesslen

- Beschreibung aktualisiert

#4 - 22.02.2023 13:06 - Maximilian Seesslen

- Beschreibung aktualisiert

#5 - 22.02.2023 13:31 - Maximilian Seesslen

Alignment would be also ok in the "pretty-print" version;

```

#if defined LEPTO_LOG_PRETTY_PRINT
    int len=printf("%s(%d)" , file, line);
    for(int i1=0; i1<20-len; i1++)
    {
        printf(".");
    }
    printf(": " ANSI_ORANGE);
#endif

```

#6 - 22.02.2023 13:36 - Maximilian Seesslen

- Beschreibung aktualisiert

#7 - 22.02.2023 13:38 - Maximilian Seesslen

- Beschreibung aktualisiert

#8 - 22.02.2023 13:39 - Maximilian Seesslen

- Beschreibung aktualisiert

#9 - 22.02.2023 18:25 - Maximilian Seesslen

- Beschreibung aktualisiert

How to measure the 'cost';

- Build debug (CANRec), release; track sizes
- create feature-branch
- implement it
- Build debug (CANRec) without PP, release without PP, both with PP; track size *
- campo.sh trackSizes

#10 - 02.03.2023 11:38 - Maximilian Seesslen

- Status wurde von Neu zu Erledigt geändert