Release 3363

Bugs solved

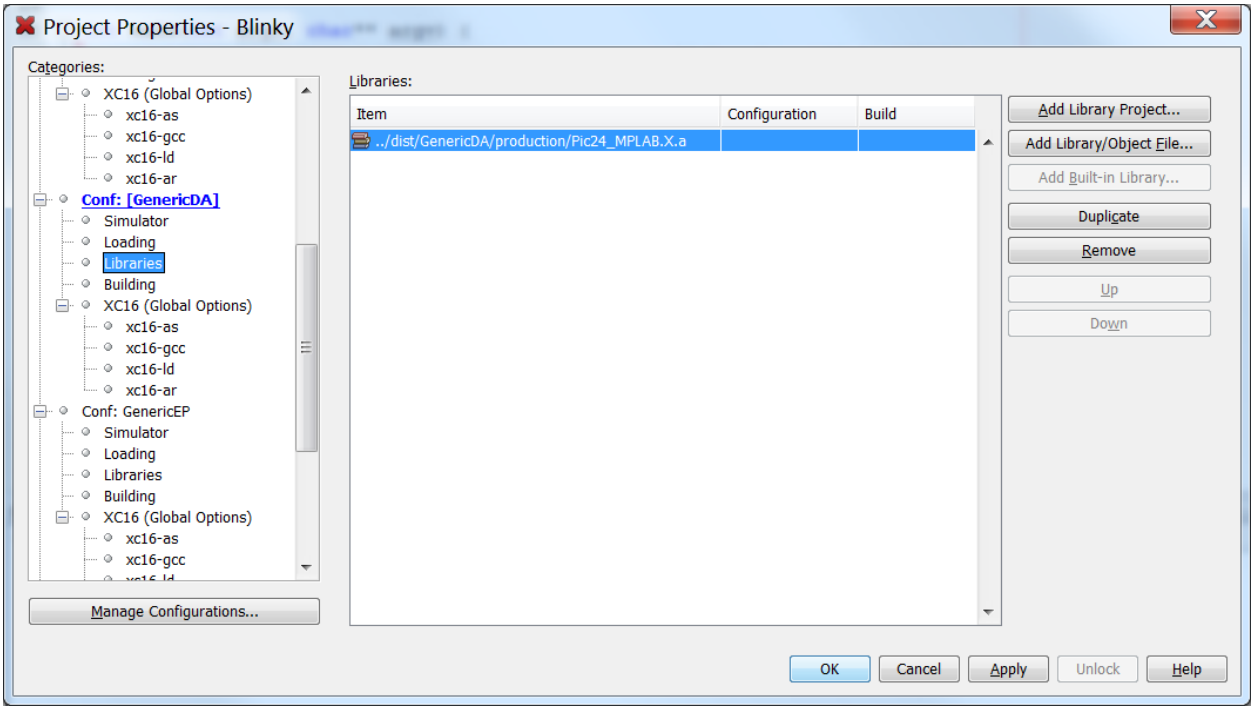
- All functions with the TO suffix don't handle RTCC wait-times correctly.
- The yRtccX.c examples were not distributed in release 3356. Those examples handle the low power functions of the hardware RTCC and some other RTCC implementations.
- The file qKernelInternal.h was distributed but is not used.
- The version number was incorrect
- The qThrSleep() function doesn't handle RTCC wait-times correctly.
- The timer functions don't handle RTCC timing correctly.
- The qKrnStack function does not return the stack size correctly.
- The qErrNotify() function can exhaust resources if called before the RTOS is initialized.
- Some variables have the "volatile" keyword because at the highest optimize levels the generated code cannot prevent race conditions.
- Bugs in the qThrEvt*() functions have been solved.
- The PIC32 implementation uses inline assembler code for atomic operations and with some compiler versions the inline registers are used incorrectly. We have solved this by specifying registers in the inline assembler.
- Under certain circumstances and load a register is corrupted in the PIC32 versions. This has been solved.

Changes

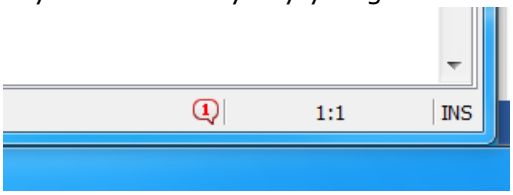
- The system now detects if power management functions qPwrPermitIdle(), qPwrPreventIdle(), qPwrPermitSleep() and qPwrPreventSleep() are out of sync. In that case it throws the error qERR_PWR_IDLE_OUT_OF_SYNC or qERR_PWR_SLEEP_OUT_OF_SYNC
- The system now detects if there is more than 0x8000 bytes of memory in some PIC24F devices of the DA type, PIC24E and dsPIC24E and handles the memory automatically.
- A new option is available to specify if a thread is added at the beginning of the list of threads with the same priority or at the end. This helps with porting existing co-operative scheduled applications.
- Some code has been optimized to handle 32-bit multiplications on PIC24 devices.
- The PIP structure has been extended to support the high load framed UART driver which will be released later. Existing software that uses the PIP structure don't need any change.
-

Other

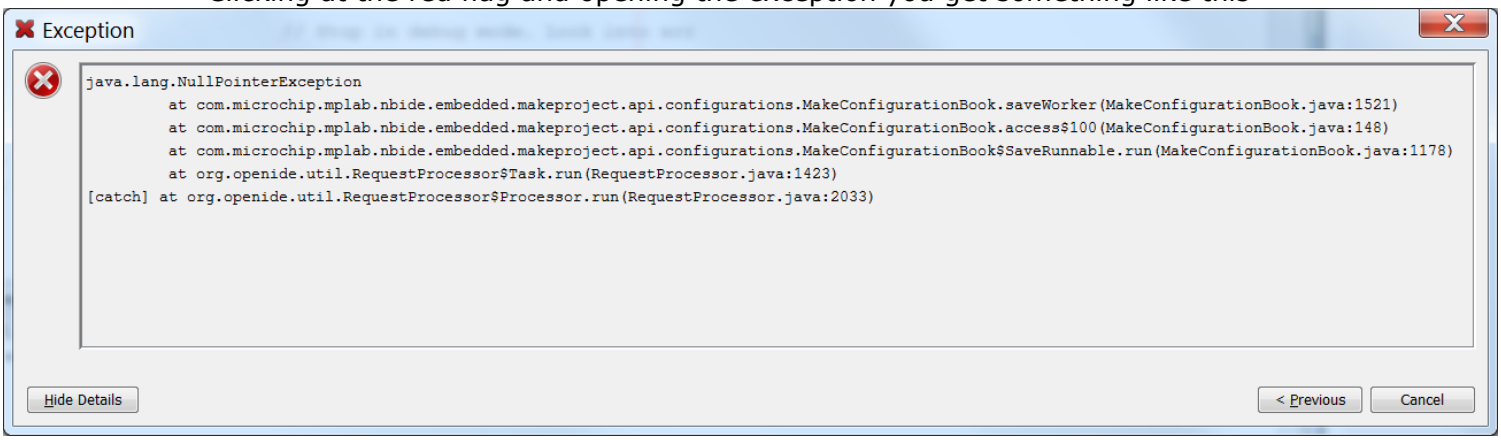
- The minimum version of the C32 compiler is V1.12. Versions before that don't work and generate a compiler error in the compiler test.
- This version is tested with version V3.00 from MPLAB-X. This version fails to load a project library file in the main project. You have to specify the object library (extension ".a") as shown in the following screenshot:



If you do that anyway you get a little red flag in the bottom of the screen



Clicking at the red flag and opening the exception you get something like this



Q-Viewer Bug solved

- Idle Task Name was incorrectly displayed in the qViewer

Q-Viewer Bug not solved

- In the future the viewer will be part of the Microchip MPLAB-X plugins. Until then a manual install is required.
- To install the viewer download the manual from the website and follow the installation instructions. The *.nmb file is located in the qViewer directory under the version V3361/qViewer

Release 3356

Bugs

- The PIC32 port was shipped without yPip.c.
- The function qPipGet() called qPipPutFast() instead of qPipGetFast()

Q-Viewer

- In the future the viewer will be part of the Microchip MPLAB-X plugins. Until then a manual install is required.
- To install the viewer download the manual from the website and follow the installation instructions. The *.nmb file is located in the qViewer directory under the version V3356/qViewer

Release 3353

New

- The Q-Viewer is included as a beta version in this release
- New pipe function for fast access to the pipe. This provides faster access for hardware devices with fixed length block size like SPI or UARTS. The functions are:
    - qPipGetBytFast Reads one byte from the pipe without synchronization
    - qPipGetWrdFast Reads one word from the pipe without synchronization
    - qPipPutBytFast Writes one byte into the pipe without synchronization
    - qPipPutWrdFast Writes one word into the pipe without synchronization
    - qPipReadFast Reads one block from pipe with full synchronization
    - qPipWriteFast Writes one block into pipe with full synchronization
- The PIC24 distribution has changed the name of the device groups from GenericEDS to GenericDA which follows the device group naming of the compiler

Bugs

- The problem with the message use-count in the Q-Viewer required a change the message structure.
- Label in inline code with a value of 10$: and higher have been changed to lower than 10$: because it creates under some circumstances compiler errors.
- Several bugs in the PIC24 assembler pipe functions have been solved
- Under heavy interrupt load the Entries variable in the pipe structure could become negative while the variable is unsigned. The variable and the atomic word functions are now signed.

Documentation

- Only these release notes and the porting guides are in the documentation directory. All other documentation can be downloaded from the website.

Q-Viewer

- To install the viewer download the manual from the website and follow the installation instructions. The *.nmb file is located in the qViewer directory under the version V3353/qViewer

Release 3352

This release includes the PIC32 port, changes to the error handling, some small changes in naming and some bugs are solved.

PIC32

The PIC32 implementation is function complete with the following exceptions:

- Statistics is not yet implemented
- Tracking not implemented
- New pipe functions are not yet implemented

There is serious problems with some of the compiler versions and for that reason we advise to use XC 1.31 only. Exceptions have serious issues during debugging and for that reason we have removed the error handling of exceptions out of the standard Q-Kernel handling. Include the following code to find exceptions:

```
static unsigned qvTrapAddress;
void _general_exception_handler() {
    asm volatile("mfc0 %0,$14" : "=r" (qvTrapAddress));
    qBREAKPOINT;
}
```

Error handling.

The error handling is functional the same as in previous versions but has been implemented differently to handle the RCON register better. The function qKrnError() reset the processor in software and stores the error information in persistent memory. After the reset the function qKrnInit() returns a pointer to this information. What has been changed is that this only occurs if the processor has been reset in software. It also set the RCON register to zero so the user has to test RCON before a call to qKrnInit().

We have also included the critical status in the error information. A value of -1 (all F's) indicates not in a critical section, 0 in first critical section, 1 in second critical section, etc.

Bugs.

- The function qSemAcquire() contained a bug that prevented it from functioning when not compiled with__NO_CHECK__. The bug is fixed.
- Priority fibers are incorrectly activated. The bug is fixed.
- The name BREAKPOINT does not follow the standards. Should be qBREAKPOINT. This has been fixed. People that want to keep using the name BREAKPOINT have to create a #define for it.

Small changes in internal functions

- The name of the function xCrtEnterFirst() has been changed to qCrtEnterFirst()

- The name of the function xIntDecIfNotZero() has been changed in qWrdDecIfNotZero()
- The name of the function xKrnSetInterrupt has been changed to qKrnSetInterrupt()
- Temporary variables in qFormat() are fixed for 32-bit implementations of Q-Kernel
- The structure of header files has been changed. The main file qKernel.h is the main include file and exists for every port while the file qKernelCommon.h is referenced from qKernel.h.
- The function qKrnInit() has been changed to call the new function yErrorInit().
- The implementation of the function yUSec2Cycles() has been changed and is now a non-port dependent function. Because of this the name has been changed to xUSec2Cycles()

Release 3343

First version 6 release