Bugs and their fixes

**Code hangs when running xTaskNotifyFromISR from within ISR:**

The ISR must have a higher priority than the task it is notifying. If the task has a higher priority, the code will switch to the task immediately and will not complete the ISR. Fix is to change the ISR priority to be a low number.

**Code hangs at configASSERT( ( portAIRCR\_REG & portPRIORITY\_GROUP\_MASK ) <= ulMaxPRIGROUPValue );**

Ensure that all preemption priorities are zero. If they are non-zero, FreeRTOS does not work properly and will configAssert (error and loop forever). If all priorities are zero then NVIC\_SetPriorityGrouping( 0 ); can be used before the scheduler starts to ensure they are all zero.

**ASM330 not responding**

The ASM330 initially was not replying over SPI. After writing the NSS pin high during initialisation the sensor started working. Since the code used is from a generic library, a library agnostic solution is to set the default state of the NSS pin to HIGH in the .ioc configuration by setting the pin’s GPIO output level to HIGH.

**Cannot run STM32H743 past 192MHz. I get an error in HAL clock configuration:**

**f\_mkdir function hangs forever in SDIO 4 bit mode:**

This problem seemed to be fix by increasing the clk divide factor to 3. Any value lower and the code breaks.

**FreeRTOS gives up priority of data logging task after f\_stat:**

The cause of this is still unknown however, it appears to be specific to when using FreeRTOS. The SD card words correctly with a bare metal implementation

Update: Changing clk divide factor to 3 appeared to fix this issue.

**USB device recognised but mass storage not initialised:**

This was caused because the SDIO clk speed was higher than the USB clk speed. This was fixed to lowing the SDIO clk divider to 4 (for 192mhz sys clk speed).

**MS5611 did not respond to SPI commands:**

Either during the soldering (too hot) or otherwise, the sensor appears to have been damaged. A new sensor soldered at 300C worked correctly.