Framework Guidelines

- **No dynamic memory allocation**
    - Avoids an entire class of bugs (heap fragmentation, allocation failures, etc.)
    - Stronger guarantees of long uptimes
    - Sometimes harder to design and write code
    - All memory must be allocated at compile time
    - Can limit choice of 3rd party libraries such as printf
    - Benefit is some types of bugs will get caught earlier, or avoided entirely, by this “front loaded” approach

- **Careful use of interrupts**
    - Spend shortest amount of time possible in interrupt context
    - Use efficient signalling mechanisms to notify threads of events

- **Star pattern**
    - Avoid crisscrossed / spaghetti relationships
    - Central entity manages relationships between components and threads

- **Threads / tasks**
    - Prefer to use message queues & message passing
    - Avoid explicit use of mutexes, semaphores, and other synchronization primitives, unless required
    - Single thread to interact with a peripheral avoids inter-thread synchronizartion issues or raise conditions

Possible Improvements

- Implement Kalman Filter
- Better comments. Use of Doxygen
- Look into trigonometric function implementations for embedded
- Could implement a HPF for gyro data
- Could consider HW vs SW timer tradeoffs