Thread Safety Issues

Threads have their own stack but static variables within the program are on kept on the stack. The threads simultaneously share access to these variables. This may cause a logic error called a race condition where multiple threads may overlap while compiling and running the same function and the variable may be changed by these threads while another thread is in the middle of checking it. In order to avoid this issue is to use the mutex lock.

This fix may work if the function is not re-entrant as it could interrupt the function in the middle of execution which in turn can disable other interrupts, making the whole program suffer. By using atomic variables however, this may be avoided.