Peer Review Checklist: Embedded C Code

Before Review:
0 _____ Code compiles clean with extensive warning checks (e.g. MISRA C rules)

Reviewer #1:
1 _____ Commenting: top of file, start of function, code that needs an explanation
2 _____ Style is consistent and follows style guidelines
3 _____ Proper modularity, module size, use of .h files and #includes
4 _____ No orphans (redundant, dead, commented out, unused code & variables)
5 _____ Conditional expressions evaluate to a boolean value; no assignments
6 _____ Parentheses used to avoid operator precedence confusion
7 _____ All switch statements have a default clause; preferably an error trap

Reviewer #2:
8 _____ Single point of exit from each function
9 _____ Loop entry and exit conditions correct; minimum continue/break complexity
10 _____ Conditionals should be minimally nested (generally only one or two deep)
11 _____ All functions can be unit tested; SCC or SF complexity less than 10 to 15
12 _____ Use const and inline instead of #define; minimize conditional compilation
13 _____ Avoid use of magic numbers (constant values embedded in code)
14 _____ Use strong typing (includes: sized types, structs for coupled data, const)
15 _____ Variables have well chosen names and are initialized at definition

Reviewer #3:
16 _____ Minimum scope for all functions and variables; essentially no globals
17 _____ Concurrency issues? (locking, volatile keyword, minimize blocking time)
18 _____ Input parameter checking is done (style, completeness)
19 _____ Error handling for function returns is appropriate
20 _____ Null pointers, division by zero, null strings, boundary conditions handled
21 _____ Floating point use is OK (equality, NaN, INF, roundoff); use of fixed point
22 _____ Buffer overflow safety (bound checking, avoid unsafe string operations)

All Reviewers
23 _____ Does the code match the detailed design (correct functionality)?
24 _____ Is the code as simple, obvious, and easy to review as possible?

For TWO Reviewers assign items:  Reviewer#1: 1-11; 23-24  Reviewer#2: 12-24
Items that are covered with static analysis can be removed from checklist
Template 1/27/2018:  Copyright 2018, CC BY 4.0, Philip Koopman