Ultra Maritime Digital
Communications Centre

Software Guidelines

The following Guidelines apply to any party developing software for or in partnership with the Ultra Maritime Digital Communications Centre (UMDCC). These guidelines are focused on development in the Simulink environment but also apply to projects coded in Matlab, or any other modern programming language.

April 17, 2014

Verification Process

Whether you are developing a single custom Simulink block/function, a complete module or an entire system the design must go through a verification process before being included in the UMDCC software library. This software library will be maintained by one individual but will be available for use to all team members.

To have software verified and included in the library the developer must meet the following criteria:

- The software must meet the Guidelines outlined below
- The developer must write a test application and verify the software functions as intended
- The developer must participate in a design review where he/she walks the team through the software they have created
- The software must be tested and verified by a third party
- The developer must provide a written recommendation for inclusion into the library, including intent to keep or remove previous versions of the software

Development Guidelines

- Naming Conventions
  - Variables should have descriptive names in lower case
  - Functions should have descriptive names with the first letter capitalized
  - Constants should have descriptive names in all capitals
  - Modules and documentation should have consistent naming; differentiated only by their extension (name.mdl, name.pdf, name.c, name.h, etc.)
- Block/function descriptions

  - Each block or function must be fully described through commenting in the source code
  - Input/Output statements and a description of their relationship must be provided for each custom Simulink block or C/C++/Matlab function
    * These statements should provide a brief but complete description of each input the function and the resultant outputs
    * This description can be provided as a mathematical statement or a brief but complete verbal description

- General Guidelines
  - Follow ANSI standards for C/C++ development
  - No for loops in Matlab
  - No passing of hidden variables
  - Global variables must be passed through function calls
  - Avoid dynamic memory allocation
  - Avoid multiple indexing (Use 1D static variables)

- Testing/Verification
  - A test program must be developed solely for the purpose of testing new software
  - Software is not to be used in any system level simulations or submitted for review before being fully tested according to the Testing and Verification Guidelines outlined below

- Documentation
  - Owner, date, and revision number must be included in both source code and documentation
  - Input/Output statements must be included in both source code and documentation
  - Description of Input/Output relationship must be included in both source code and documentation

    - Each function/module/system must include an application note which describes how someone unfamiliar with the software could use it as part of their design
        - This should include a set of safe operating parameters for the inputs

Testing and Verification Guidelines

To pass testing and verification the developer must create test software which demonstrates the functionality of their function/module/system and prove that it performs as expected for a set of known inputs. The developer should put in time and effort to determine what parameters can cause the system to fail and outline them fully in the documentation. For verification and inclusion in the UMDCC library the documentation must be consistent with the source code when examined by a third party.

Version Numbering

Software produced for the UMDCC group must follow a two stage version numbering convention (v x.y). Changes in functionality, the addition of new features, or modification of the inputs and/or outputs will be indicated by incrementing the major version number(x). Minor revisions to the documentation, structure/organization of the code, or minor bug fixes will be indicated by incrementing the minor version number(y). All code still under development (not yet included in the UMDCC library) will be marked as beta by having a major version number (x) of zero.

The Software Library

The software library will be maintained by an individual to ensure that version control is properly handled and no unverified software is submitted to the library. The library will be stored in the cloud and accessible to all people working on UMDCC projects.