# Design

TestCase

* Setup Function
* Teardown Function
* Test Methods
* RunTest Function (Calls all Test Method Functions)

TestSuite

* RunTest Function (Calls multiple TestCase RunTest Functions)

TestRunner

* Main Function
* Runs Tests
* Reports Results
* Top Level Setup and Teardown Functions ????

Standard set of Asserts

* AssertTrue – Boolean expression
* AssertDoubleCmpAbs – Compare two doubles with a absolute tolerance
* AssertDoubleCmpRel – Compare two doubles with a relative tolerance
* AssertStrCmp – Compare two strings
* AssertStrnCmp – Compare two strings limited to at most N characters
* AssertMemCmp – Compare two memory regions
* AssertMemCmpValue – Compare memory region to fixed value
* AssertMemCmpCount – Compare memory region to byte count pattern
* AssertMem2BinFileCmp – Compare memory to binary file
* AssertFailed – Explicitly fail a test
* What Else????

Is there a easy way to not require that setup and teardown functions are manually called inside each test method?

Do we want to have different levels of setup and teardown functions that are chained, i.e. TestCase setup and teardown functions vs TestRunner setup and teardown functions where the TestCase functions call the TestRunner functions? Somewhere we need to initialize everything to a default state.

Higher level test runner. Need to avoid global namespace problems. Need to consider how to initialize the test environment where each tasks set of test suites would be independent of another tasks test suites, so they could be added to any test runners set of tests to execute. Also need some way to build the unit test i.e. some standard way of doing it so that the test suite could be built and run stand alone or as a part of a bigger set of tests.

# Deliverables

Code

Examples

Documentation

Also requires updates to UTF

* Reset functions to reset the test environment to a default state.
* Additional functions to support asserts for events and software bus packets