Testing

Unit Tests

The primary goal of the unit tests are to expand the code coverage for subsystems, typically by mocking the variables necessary to test a single function.

Functional Tests

Functional tests are used to perform a more in-depth analysis of the degree of functionality for a specific subsystem. Many of the functional tests written for the submarine modules will focus on accuracy of measurements and readings. This will be extremely important in verifying the reliability of the SLAM system implemented on the submarine as new modules are transitioned from simulated testing to real-world.

In-water Tests

Real-world testing is often performed at Gibb pool, which allows for full systems testing with an in-water pinger.

Compiling and Running

To compile and run all tests, execute the following command from the ~/ros directory:

```
$ catkin_make run_tests
```

To build just one specific test, execute the following command from the ~/ros directory:

```
$ catkin_make <test_name>
```

The naming convention for tests is "test_" followed by the subsystem name, like "test_localization_system"

To run a specific test, navigate to the directory of that test (i.e. ~/ros/src/robosub/test/functional_tests/localization) and execute:

```
$ rostest <test_name>
```

By default, rostest creates a new master for all publishers and subscribers. If you wish to view messages being passed between nodes, simply add —reuse-master to the above command.

From:

https://robosub.eecs.wsu.edu/wiki/ - **Palouse RoboSub Technical Documentation**

Permanent link:

https://robosub.eecs.wsu.edu/wiki/cs/testing/start?rev=1555306799

Last update: **2019/04/14 22:39**