

4 Testing

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, power system, or software.

The testing plan should connect the requirements and the design to the adopting test strategy and instruments. In this overarching introduction, given an overview of the testing strategy. Emphasize any unique challenges to testing for your system/design.

4.1 UNIT TESTING

What units are being tested? How? Tools?

The algorithms that are implemented on the radios are what need to be tested. These can be tested by creating a test environment in which modifications can be made and performance differences can be measured. The computer and components are the only tools we need to test modifications to software. The software we are using, srsRAN, also has end-to-end LTE tests that can be used to measure bandwidth.

In regards to the enclosure, this can be tested by iterating through multiple design prototypes and 3D printing them to make sure they work as intended.

4.2 INTERFACE TESTING

The interfaces in our design include the enclosure for the B210, Skylark mMIMO, X410 with all devices inside and hooked up as well as the software that is run on them. The enclosure can be tested through ensuring that the dimensions are correct after it is printed and then making sure that the power supply can support all the devices. At the same time we have to make sure the software that we modify is actually making a difference in latency between devices for the software side of things. This can be seen when running the srsRAN program and we will also visibly be able to see if all the devices are receiving the necessary power supply to be turned on.

4.3 INTEGRATION TESTING

Integrating software algorithms onto the hardware enclosures is key to the success of our design. Testing the wireless speeds will provide us with an assessment of the validity of our design on both the hardware and software ends. If either of these parts is not designed correctly could result in not meeting the speed and latency requirements. Github tests can provide these metrics.

4.4 SYSTEM TESTING

Describe system level testing strategy. What set of unit tests, interface tests, and integration tests suffice for system level testing? This should be closely tied to the requirements. Tools?

System testing is a level of testing that data completeness and fully integrated product. Testing the full application including external peripherals in order to check how the components interact with the whole system.

4.5 REGRESSION TESTING

The only thing that would probably be applicable to regression testing would be the software. In order to ensure we do not take steps backwards when editing the software we will always keep the originals so we have something to look back on if we somehow decrease performance after certain modifications.

4.6 ACCEPTANCE TESTING

For our project, acceptance testing would involve showing our advisor that the software enclosed in the UE system can successfully communicate with the base station. This communication will have to be reliable and reach the specified upload and download speed. Our team will also have to show our advisor that the enclosure we created is sturdy and can enclose all of the required hardware.

4.7 SECURITY TESTING (IF APPLICABLE)

N/A

4.8 RESULTS

What are the results of your testing? How do they ensure compliance with the requirements? Include figures and tables to explain your testing process better. A summary narrative concluding that your design is as intended is useful.

This will be graphs comparing the performance and measurements of different algorithms on different SDRs.