Deliverable 3 - Preliminary Comprehensive Modular Build and Testing Plan (P-CMBTP

EECE 460

Team 2: MCU TNC Design

David Cain – C00043561

Kobe Keopraseuth – C00092349

Kaleb Leon – C00094357

Monday, Sep. 21, 2020

We designed our testing tree by back tracking our way through our current development process to decide which subsystems and components we need to be testing. We plan to have our Preliminary Comprehensive Modular Build and Testing Plan (PCMBTP) thoroughly assess our design's functionality. The design is statistically feasible as stated in previous appendices, so now we must test that physical and digital feasibility. This is done surgically by looking at every component and testing them then moving up to the subsystem made of these components and testing them together. Then, we finally reach the point where all the subsystem come together to test the main system as a whole. If all subsystems and components work separately and then as a subassembly to combine to a full system, the design is then proven to pass, and work as expected.

The first subassembly that needs to be tested are the components of the physical hardware system and circuits. This includes the micro controller itself and the two external circuits. According to our FMEA, if our micro controller does not function then the code has nothing to run on making the project fail before it has even begun. In terms of the micro controller we will need to test anything, and everything being used to accomplish our design. The main portions we use in terms of signal transmission and reception are mostly the cable connections between the radio to TNC and PC to TNC. These transmission lines include: the USB cable which handles serial communication between TNC to PC, the 2.5 mm Audio jack that handles analog signals between the radio to TNC and vice versa, and RS-232 connecter that is a backup digital communication line. Testing the effects of the micro controller signal processing is also very important due to the design needing to meet a certain amount of specifications. We have to test to make sure all of our power consumption, latency, and voltages are under spec for them to work with our design and have it function properly. In addition to that, we need the I/O pins on the micro controller to be fully functioning so that they can communicate with the external circuits. These external circuits are also a main portion of the hardware that we need to test to assure our input signals are how we need to process them and to change modes. The Audio input circuit needs to be tested down to the component level of the amplifier and the filter to assure we are getting the correct audio signal in and out of our design. The Push To Talk circuit is used mainly to switch modes so that our TNC knows when we are transmitting or receiving. This also needs to be tested down to the component level so that we know if will function as needed to allow our design to perform its tasks.

The second subassembly that needs to be tested are the components of the digital software that controls the hardware and data processing. This subassembly is broken down into three major subsystems which all need to be tested down to the component level. They are as follows: Kiss Packet Handling, AX.25 Protocol Formatting, and Frequency Shifted Audio Tone Handling. The Kiss Packet Handling controls how we handle inputs and outputs at the Data Link layer. This is critical for PC to TNC communication. We must take into account and test multiple components of this subsystem. The serial communication line needs to be tested and functional to make sure we can receive and transmit data across the serial bus between PC and TNC. This component needs to also be assessed on the latency of that communication line to meet our specs. In addition, the packet construction following the KISS protocol is a very important process to KISS transmission to PC. Lastly, we need to test the data extraction from this packet to make sure we are able to extract the correct data. Similarly, for the next subsystem of AX.25 Handling we also need to check whether we are extracting the correct data and formatting correctly following the protocol or when we send it off to the DAC the signal will be incorrect when being received by other radios. Lastly, that is where the last subsystem comes into testing. The Frequency Shifted Audio Tone Handling is a critical and complex section of our software, so it needs to be extensively tested. It needs to be able to handle ADC and DAC control which each entail of their own components. If these signals come in incorrectly or out incorrectly then our design is failed and not useful as a product.

It is through this methodology that our team has designed a testing tree with the described components, subsystems, and subassemblies for testing. As testing progresses, we will add more components we overlooked such that we make sure everything is functional in our design. When navigating the tree we will also write up test reports so that all our tests are well documented and noted fixes to our faults. This will ensure confidence in our design.

B. *Diagram and Explanation of the Tree*

As shown in our testing tree, our design system is split into two Subsystems: software and hardware. The Software Subsystem is more complex than the Hardware Subsystem, since most of our project is mainly software based and the hardware's purpose is only to support receiving and transmitting signals from the TNC. In the Software Subassembly, it is broken into three branches that have many supporting leaves that represent different software processes as opposed to the Hardware Subassembly leaves that represent physical component testing. For each subassembly are different tests that we plan to run for each process and physical components. For example, under the hardware subsystem, under each circuit subassembly we will test and ensure each component's nominal value and desired signal output. Components such as resistors will have to be measured to ensure our circuits' outputs. The signal outputted from the physical amplifier circuit will have to be compared to the signal output from spice software. As for the software subsystem, we will validate that each process or subassembly outputs the correct bitstream and frequencies. The bitstreams outputted from the microcontroller have to be outputted in specific sequence. The microcontroller also must output specific frequencies and must be measured to ensure that there are not many errors from the output. Latency will also be tested for serial communication to ensure optimal performance.
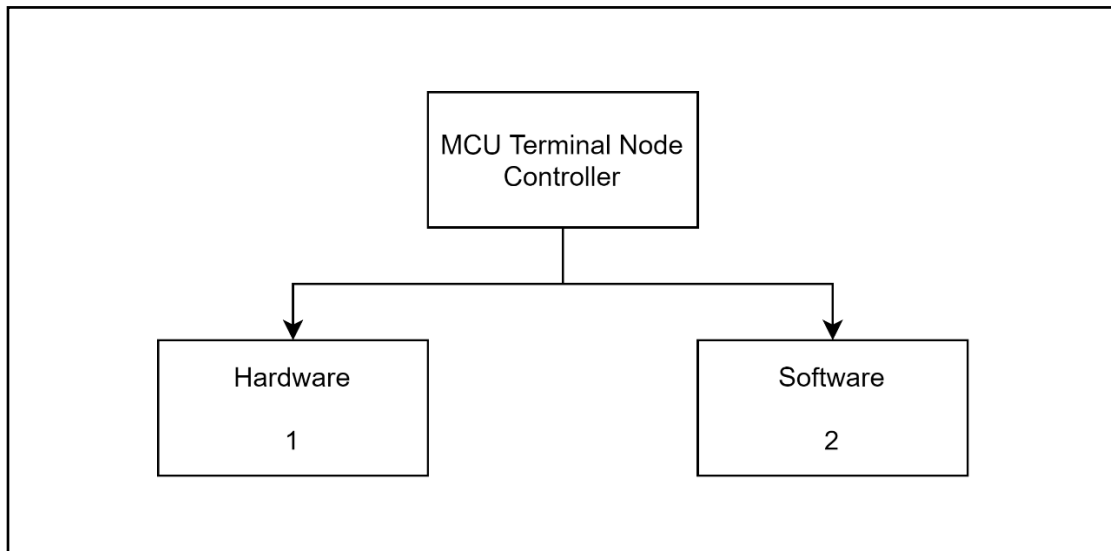
Top of Comprehensive Testing Tree
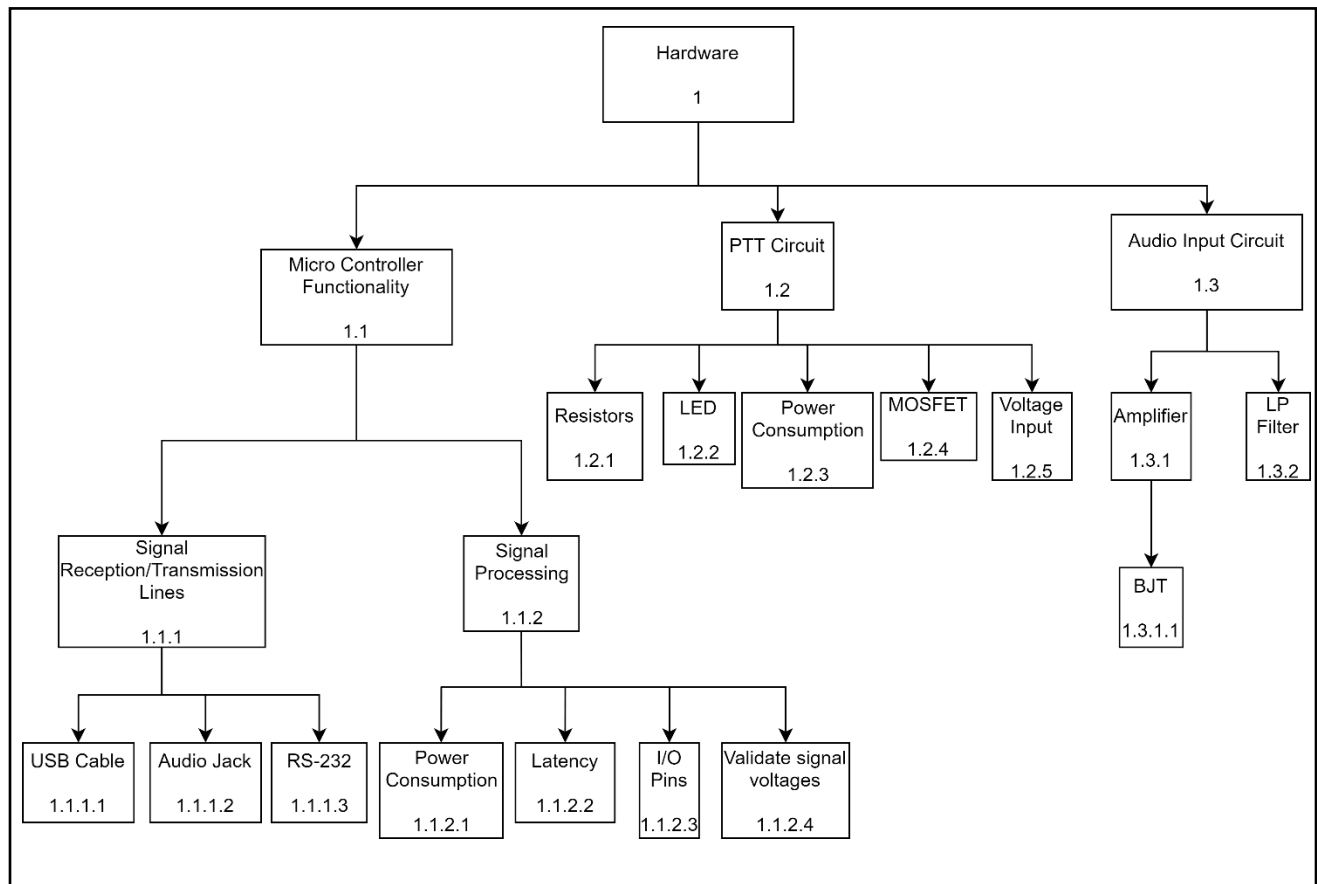
Hardware Comprehensive Testing Tree
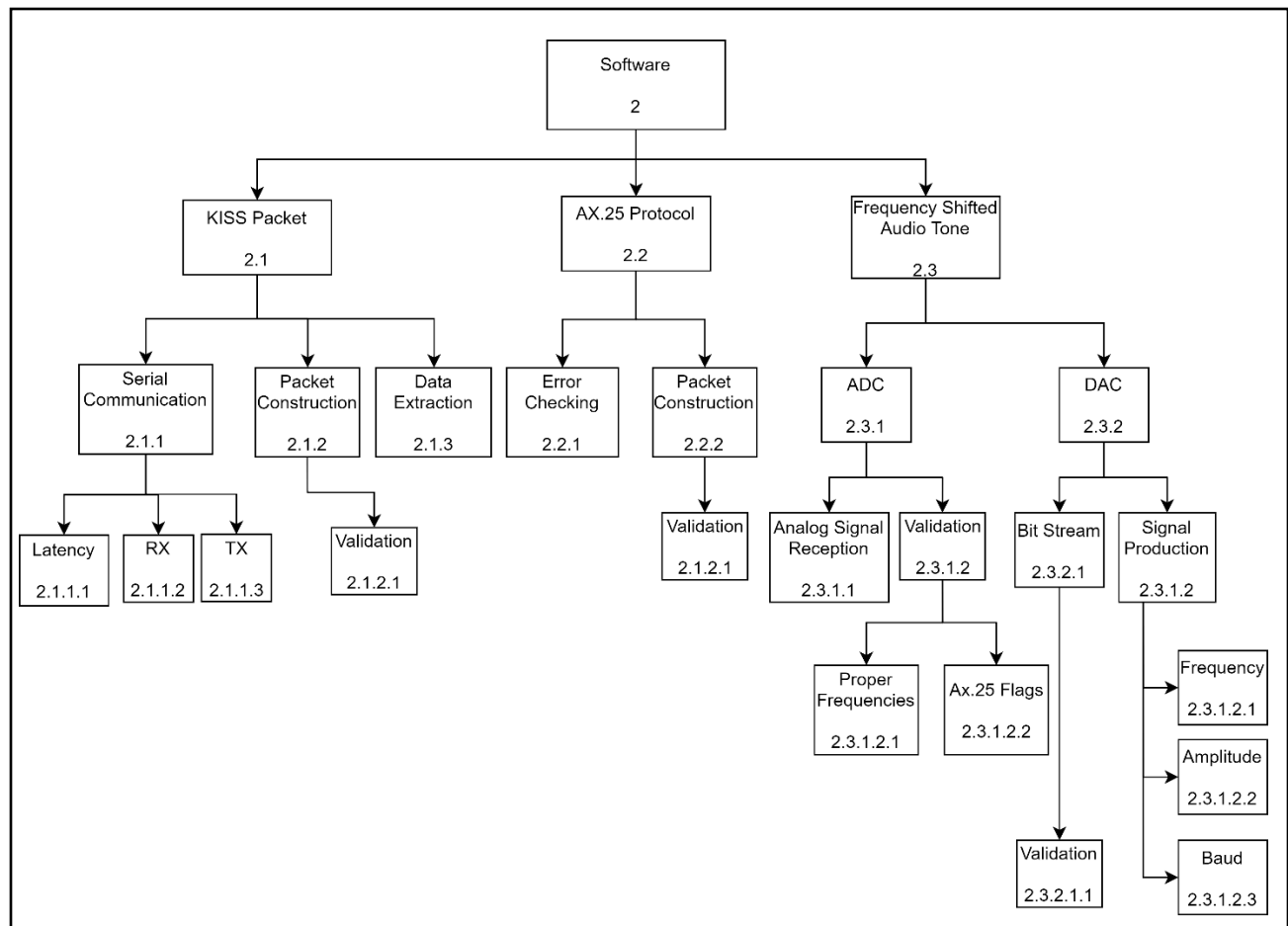
Software Comprehensive Testing Plan

C. Test Report Template

TNC TESTING FORM(REV1)	
Leaf on the Tree:	
Device Under Test (Testing Tree Number):	
Date:	
Person(s) Conducting Experiment:	
Signature:	
Experiment Purpose:	
Experiment Procedure:	
Equipment Settings/Software Settings (w Revision):	
Testing Diagram/Picture:	
Data Points:	
Pass/Fail:	
Interpreted Notes:	
Recommendations for Modification:	