1. **Purpose**
   1. PLACEHOLDER
2. **Scope**
   1. PLACE HOLDER
3. **Description of software**
   1. PLACEHOLDER
4. **Software acceptance criteria and standards**
   1. The following acceptance criteria and standards shall be used to verify the software before use in clinical trials. All tests will be performed with the prospective software release installed on the Summit NIP and with three sets of three Micro2+Stim frontends attached to the Summit NIP. This configuration is described as the Summit bench test unit in the following.
   2. *Multiple/Simultaneous electrode stimulation at multiple electrodes:*
      1. To test the ability to stimulate at a specified amperage, each Mirco2+Stim frontend of the Summit Take-home system shall be connected to a Ripple resistor board with an 50 KΩ (ripple something something). Voltages across the resistor shall be measured during the intended stimulation. For this test, a single, fixed amplitude of current intensity and pulse duration will be used for both the cathodic and anodic phases during a charge balanced stimulation
   3. *Accuracy of stimulation amplitude.*
      1. To be considered an “accurate” stimulation, the amplitude mean during the plateau of each phase is within 10% of the expected value. The time of the window for mean absolute averaging is an **X** ms long slice experimentally found to consistently not contain any of the rise or drop of each phase. Beyond verifying the accuracy of stimulation, this test verifies that the combination of the specific resistor value and voltage recording system can adequately identify broken channels and provides an confirmation of stimulation command expectations.
   4. *Accuracy of stimulation train interpulse periods.*
      1. To be considered an “accurate” stimulation, the time duration between the start of the response voltage recorded for two sequential pulses on the intended channel must be within either 0.1 ms or 10% of the expected value, whichever is greater. The time of the start of the pulse will be at the point that the voltage reaches 90% of the extremum value of the first phase of the current pulse. Beyond verifying the accuracy of stimulation, this test verifies that the combination of the specific resistor value and voltage recording system can adequately identify the duration between sequential pulses on a channel.
   5. *Limitation of stimulation train interpulse periods.* 
      1. To be considered an “accurate” stimulation, the time duration between the start of the response voltage recorded for two sequential pulses on the intended channel must be within either 0.1 ms or 10% of the expected value, whichever is greater.
   6. *Acceptance of individual stimulation routine.*
      1. For the individual stimulation routine to be considered acceptable, the intended channel must be “accurate” at 3 of 3 intended train interpulse periods within the train interpulse period limit. Further, the intended channel must be “limited” at 3 of 3 intended train interpulse periods that fall below the train interpulse period limit.
5. **Software storage and version control**
   1. Once the software has been verified as acceptable, it shall be given a unique timestamped filename and a checksum of the code shall be saved in accordance with SOP CNI 03D-00 Data Archiving.
6. **Hardware(?) verification tests**
   1. Follow steps to set up Summit NIP (SNIP), and Ripple test board.
   2. Open Matlab and select “include folders and subfolders” for the following files:
      1. [INSERT FILEPATH]
   3. Open “StimVerifyScript.mat”
      1. Run the code and start impedance collection for all 32 channels.
   4. Once impedance collection is complete, one file will be saved named:
      1. “StimVerification\_#”
         1. # indicating which run of impedance testing.
7. **Data Collection**
   1. **[PLACEHOLDER]**
8. **Data Analysis**
   1. Open MATLAB, navigate to, and open the following file:
      1. [FILEPATH]/**StimAnalysisFull.m**
   2. Open the desired “StimVerification\_#” file from the appropriate directory
   3. Ensure there are two nonzero matrices inside named:
      1. “SC” (1x32 cell)
      2. “WF” (32x2500 cell)
   4. Run “StimAnalysisFull” and wait briefly. The script will run through analysis of data from all 32 channels tested and return four (4) separate heatmaps and a singular value:
      1. **Figure 1 (numberOverThresh)**
         1. Plots the total number of stimulations returning values above expected per channel. The number of above expected values is plotted within the bars of the heatmap. Red indicates a higher number of stimulations that were recorded as being higher than expected values.
      2. **Figure 2 (avgVOmap)**
         1. Plots the mean voltage above expected values for each channel. Red indicating an average further away from expected values.
      3. **Figure 3 (maxVolts)**
         1. Plots the maximum voltage recorded that occurred above expected values for each channel. Red indicates a higher maximum, blue indicates a lower maximum.
      4. **Figure 4 (dRecorded)**
         1. Plots the difference expected values and recorded values across all time points for each channel.
      5. **meanoverstimtotal**
         1. contains the average overstimulation of all channels on the board.
   5. Once data has been collected and figures produced, save to the desired directory and press control + C to clear the variables.
   6. Load a new “StimVerification” file and repeat above steps.
9. **Stimulation Verification Testing (Oscilloscope)**
   1. Ensure NOMAD Summit is connected properly
      1. Ethernet cable plugged into ethernet port on power cable
      2. Ensure proper one-up and three-up cables are plugged in the following order
         1. Three-up cables into ports-A-C
         2. One-up cable into Port D (not necessary for stimulation)
      3. Attach the three up cable’s front end (from port A) into break out board with the backside of the front end facing upwards so the number 1 (on the front end) is next to electrode 1 of the breakout board.
   2. Set up Oscilloscope
      1. Plug USB from oscilloscope into computer (PNILabview)
      2. Plug power cable into wall
      3. Turn on (button on top left of oscilloscope)
      4. Make sure the settings are the same as the photo below:
      5. Attach oscilloscope probe so the alligator clip is on the ground side (downstream) of the resistor and the pen-looking probe is attached to the source side (upstream) of the resistor
   3. Open MATLAB and use the following file path to find the Stimulation Verification Script for the Oscilloscope
      1. CNI/COB/XipppyServer/COB\_Python/StimVerification
      2. Run each section (other than the deletion group) in order until at stimulation group.
      3. Start with
         1. Front end = 1
         2. CableUpNumber = 1 (port A 2= Port B 3=Port C)
      4. Run this section while moving cable from electrode 1->3->5….31->2->4….32
      5. Swap Front end so 2nd front end is plugged into breakout board
      6. Add one to Front End so FrontEnd = 2
      7. Run this section while moving cable from electrode 1->3->5….31->2->4….32
      8. Swap Front end so 3rd front end is plugged into breakout board
      9. Add one to Front End so FrontEnd = 3
      10. Swap Front end so 3rd front end is plugged into breakout board
      11. Run this section while moving cable from electrode 1->3->5….31->2->4….32
      12. Unplug frontend 3 from breakout board and plug in front end 1 from Port B
      13. Add one to CableUpNumber so CableUpNumber = 2 (Port B or 3 for Port C)
      14. Repeats steps 9.3.l to 9.3.u for ports B and C
   4. **For each set of 32 channels being tested: Steps 9.3.l, 9.3.o, 9.3,s**
      1. Touch cable with resistor to the first channel
      2. Run script in MATLAB
      3. After each successful test, light on Oscilloscope will turn green. Once the light turns green:
      4. Move cable to next channel in sequence
      5. Wait for green light
      6. Repeat.
   5. Once all 32 channels on the board have been recorded, save results in the “StimVerification” folder (should be automatic)
      1. Make sure to label each test so you know which is which