**Failure Modes and Effect Analysis:**

**The FEMA table below displays the potential failures that can disrupt the functionality of our project. It will help guide us to make sure our project is fool proof. The first column displays what part of the project is not functioning correctly. The second column displays the potential problem that can happen with respect to the input. The third column displays what negative effects the potential failure can cause. The fourth column displays what can cause the potential failures. The last column displays what actions to take in order to fix the issues.**

**Since our project is mainly based on software, it is important our code and configuration for our STM32 microcontroller is properly done. Faulty code/configuration can result in issues for the transmission and receiving of our TNC. Packet formatting is also important, because invalid formats can result in misinterpretation from the systems receiving from our TNC. Also, since the “push to talk” is the only circuit, besides our microcontroller, it can possibly fail if the components used are not sufficient.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Process Step/Input** | **Potential Failure Mode** | **Potential Failure Effects** | **Potential Failure Causes** | **Action Recommended** |
| Bits in packets | **1.)** Receiving TNC/Computer mistakes bits for flags | **1.)**   * misinterpretation of information from receiving end * disposal of packet due to invalid size | **1.)**   * Bits, anywhere in payload of KISS packet, are arranged as “11000000” * Bits, anywhere not in flags of HDLC packet, are arranged as “01111110” | **1.)**   * **Bit stuffing:** * In KISS mode, a “1” is added after every “00000” arrangement in payload. Receiving TNC removes added “1” after every “00000” * In a HDLC, a “0” is added after every “11111” arrangement. Receiving TNC removes added “0” after every “11111” |
| Packet format | **1.)** Invalid Packet  Format:   * Less than 136 bits in frame * Not bounded by opening and closing flags * Not octect aligned | **­1.)**   * Inaccurate information received | **1.)**   * Code failure * Excess noise on the received audio to digital conversion | **1.)**   * Receiving TNC disposes packet * Rewrite Code |
| Transmitter | **1.)** Transmitter is kept on for an extensive amount of time | **1.)**   * Receiver is polling for an extensive amount of time for frames to be sent | **1.)**   * Delay in frames being sent | **1.)**   * **Inter-Frame Time Fill:** when necessary for a TNC to keep transmitter on while not sending frames, flags should be sent to fill in timebetween frames being sent |
| Microcontroller | **1.)** Transmits audio signals with improper frequencies  **2.)** receives audio signal with noise | **1.)**   * bit errors * Receiving TNC misinterprets data   **2.)**   * bit errors in packets sent | **­1.)**   * Incorrect code/configuration   **2.)**   * noisy environment | **1.)**   * Reconfigure microcontroller or rewrite code   **2.)**   * check for good connections * Move to a less noisy environment |
| Push-to-talk (PTT) | **1.)** LED burns out  **2.)** MOSFET gets too hot | **1.)**   * User cannot tell if TNC is sending audio signal to radio.   **2.)**   * Can damage components near MOSFET * MOSFET can burn out and TNC cannot perform audio transmission | **1.)**   * LED used is old   **2.)**   * MOSFET is consuming too much power * insufficient MOSFET used to handle required Power * improper capacitors and resistors used in PTT circuit | **1.)**   * Replace old LED   **2.)**   * Add heat sink to MOSFET * Replace MOSFET with a better one * Reconsider using different resistors/capacitors in circuit |

