A. Receiving TNC Flowchart

The following flowchart displays the process for receiving an audio tone from the radio. The TNC will first convert the received packet to digital, then check for any bit errors by comparing FCS fields between transmitter and receiver. If it is a valid packet, then it will check if digipeat is enabled. If digipeat is enabled, then it point to the “valid AX.25 Packet (checkpoint)” bubble in the transmitting flowchart. If digipeat is disabled, it will then check if the TNC should be receiving the packet and convert it to a KISS packet after removing the bit stuffed zeros if any.

A close up of text on a white background

Description automatically generated

B. Transmitting TNC Flowchart

The following flowchart displays the process for transmitting an audio tone to the radio. The PC first sends the KISS formatted packet to the TNC, through USB/UART, with the specified subfields in the packet. Then, it puts the packet in AX.25/HDLC format as shown in “AX.25/HDLC Formatting Flowchart.” If the packet needs to be transmitted to any digipeaters it will then require a repeater subfield in the address field. When the packet is valid, it will then start the transmitting process. It first turns on the push to talk button, then converts the HDLC packet to an analog signal using ASFK. Once the radio receives the signal, the push to talk button then turns off. This flowchart also includes the case where the TNC is receiving from a radio or if digipeat is enabled.

A close up of a map

Description automatically generated

C. Packet Formatting Flowchart

The following flowchart displays process for formatting a KISS packet into an HDLC packet. It first changes the KISS flags from “11000000” to “01111110.” Then an address field is created, using address bits from payload, which includes the packet’s destination, source and repeater(if any). Then the packet includes what type of frame it is and if it is an I frame, then it will generate a Protocol Identification field. Then if there are any five consecutive “1’s” in any field other than the flags, it will add bit stuffed “0’s.” Then a FCS field will be generated to check for any errors between transmitter and receiver.

A picture containing game

Description automatically generated