

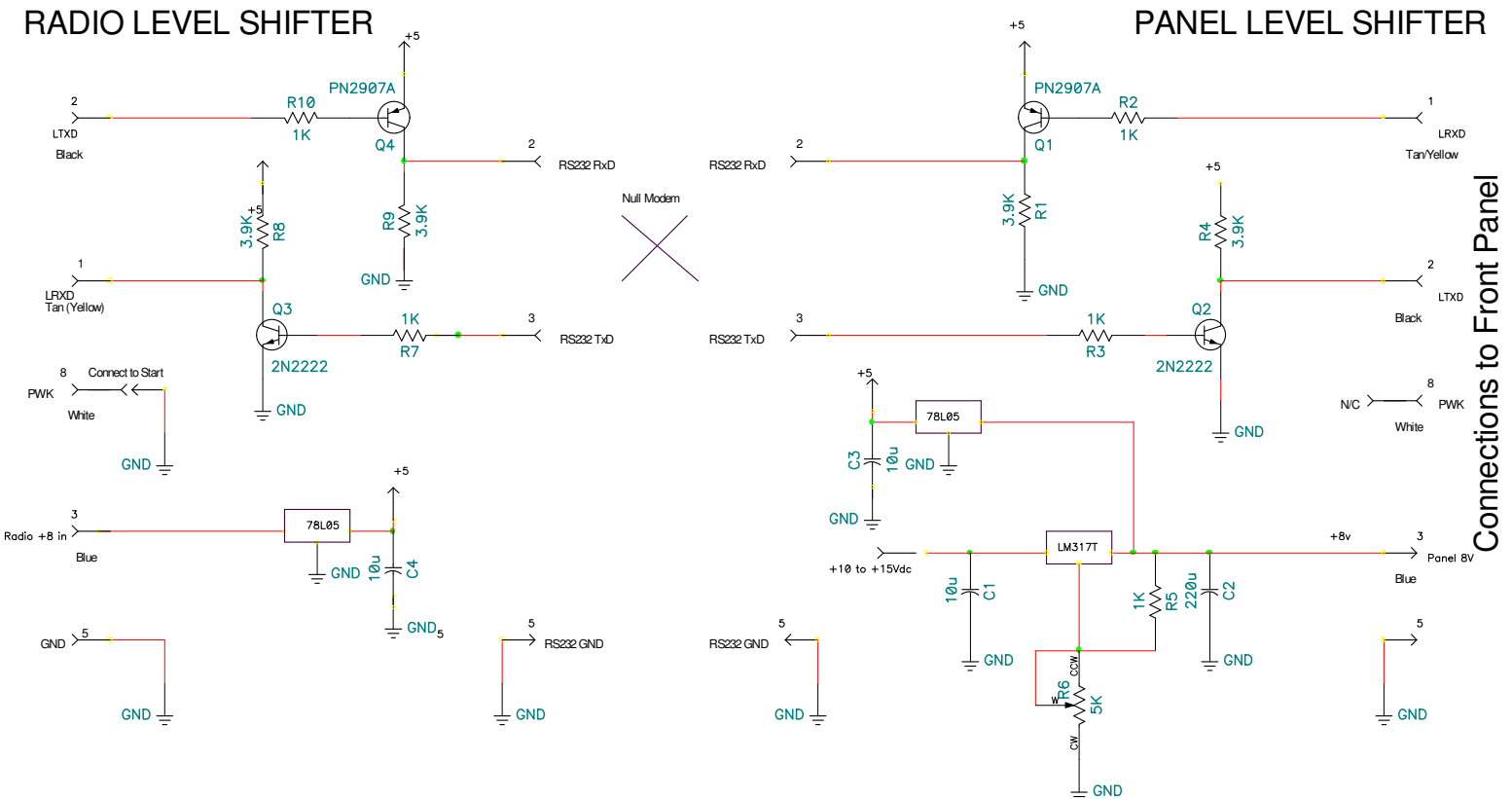
Intercepting the ICOM IC-706MKIIG Control Head Codes

Frederick Glenn, K9SO

The purpose of this article is to describe the communications between the control head (HEAD) and the radio part (RADIO) of the IC-706, presumably all versions. The purpose of this project has been four-fold: First (1), to be able to control the RADIO part by using the control HEAD connected over a LAN (or Internet). Second (2), to be able to augment the limited CI-V command set of the IC-706 to allow full-featured computer control (for example CI-V codes for this radio don't include a volume adjustment). Third (3), to be able to continue using the RADIO part in the event of a lost or stolen HEAD. Finally (4), to be able to use the control head of the IC-706 as a CI-V (or other) interpreter to control virtually any radio via the Internet by assigning various functions to the HEAD button presses. So far, work has progressed on parts (1) and (2). More effort must be made to accomplish the goals of (3) and (4).

First, I'd like to acknowledge the work of SM7LCB who was the first person that I know of who had successfully remoted the HEAD of the IC-706 via the Internet. Likewise, I'd like to acknowledge the work of K5ATM who first envisioned the computer control concept for the IC-706. This article builds on their work with this radio and attempts to bring the pieces together with greater detail. This is an incomplete work. More will be posted as time permits.

In order to interface the HEAD and RADIO separately to a computer, one needs to build a level shifter for both ends of the separation cable as shown below:



For testing purposes, a null modem (represented by the big “X” in the middle) can be used to connect the HEAD to the RADIO. The purpose of the level shifting is to buffer the 5v serial logic of the IC-706 to the RS-232 voltages of the computer. While these interfaces shown don’t truly meet the RS-232 specifications for voltages, they usually work quite well. Alternatively, MAX232 chips can be used in a similar manner.

General Information

First, some background information. Communication between the HEAD and the RADIO is via TTL level (5V) standard serial 8-bit 19,200 bps rate. In general this is written as 19200, N, 8, 1. No handshaking is used. The general start-up process is this: The radio begins the startup sequence when the operator presses the POWER button on the control HEAD. This simply places a ground on the lead called PWK (pin 8) that is hardwired to the RADIO part. The RADIO responds by placing +8Vdc on pin 3, which is likewise hardwired back to the HEAD. This +8Vdc provides all of the operating power for the HEAD.

In the circuit above, the PWK lead is manually grounded at the RADIO end and +8Vdc is supplied by a regulator (LM317T) from a local +12V power supply. 78L05 regulators supply the +5Vdc for the interface back to the RADIO and HEAD. After grounding PWK for at least 100mS, the ground may be removed and startup will continue. Once removed and grounded again, the RADIO will begin a shut down sequence. This is simply standard operation of the POWER button on the HEAD.

Once initiated, serial data is exchanged over the two leads marked LTXD (Pin 2) and LRXD (Pin 1). While the data is being exchanged, the RADIO will beep 5 times during a successful start. In the steady state, the HEAD will send only a “Keep Alive” signal every 100mS and nothing else. Consideration of this signal is essential if operation over the Internet is desired. More details on the data protocols and codes will be given later.

Interface over LAN

Once the level shifter interface is built, interface over a LAN is straightforward and understanding of the actual protocols used is not necessary. Connecting the HEAD to a CLIENT computer’s serial port is via the 9 pin DB-9 connector in the schematic. This COM PORT must be pointed to an IP address in a separate HOST computer that is similarly connected to the RADIO part. This is not the usual “virtual com port” program since an actual hard serial port must be interfaced. One program that can do this is called REMAP PRO. A trial version of this program may be downloaded from <http://www.remappro.com>. N8VB’s virtual COM port program will not work in this manner. I used another coupled COM port to put the ground on the PWK at the HOST end (this time using a virtual com port program similar to N8VB’s).

This setup works well over a LAN where the IP packets are not significantly delayed, but over the Internet, packets can often be delayed by more than 100mS. When this happens, the “Keep Alive” signal sent out by the HEAD is delayed and the RADIO will shut off if even one “Keep Alive” packet is missing or late. Keep in mind that the ICOM designers never intended for the head to be remotely operated over the Internet and 100mS is a long and conservative time delay for a hard-wired serial circuit. So don’t blame them. We need to work around this constraint.

Interface Over the Internet

With the possibility of lost or delayed packets causing premature shutdown of the RADIO, it is clear that the Keep Alive packets must be artificially inserted in the data stream at the HOST end (RADIO end). A program called Ncom HOST is currently being written that will do this. When completed, I will post a copy on this site. For those who don’t want to wait, the Keep Alive packet is Fe 0b 00 Fe (HEX representation of the 8-bit binary packets). This must be sent every 100mS if any of them sent out by the HEAD are missing.

Reading the Codes

A simple “bridge circuit” can be placed between the two level shifters when operated with a Null Modem. This would be a high impedance interface that would in turn interface to another RS232 port. The data can then be directly read using any number of serial data reading programs available. I was fortunate to have access to a Benedict Computer serial port reader so my connections were simplified and I needed no additional interface. So I simply started pressing buttons and viewed the results (a bit tedious, but straightforward).

Structure of the Codes

Overall, the data packets in both directions are of the form:

Fe <*data*> Fd

where Fe and Fd are the HEX representations of the 8 bit binary word. This is similar to the CI-V codes except that only one ‘Fe’ is sent. That’s where the similarity between the serial HEAD codes and the CI-V codes ends.

Essentially, codes sent from the HEAD towards the RADIO simply contain button press information. Also, when an analog control is adjusted (e.g., VOLUME) a data byte corresponding to that control’s new setting is sent (0 to 255).

The display on the IC-706 consists of many “hard icons” and 3 text areas. The hard icons are display parameters that are hardwired into the display itself. Examples of the hard icons are “USB”, “CW”, “DUP”, etc. These can be turned on and off via the setting of a single bit and do not need to be “drawn” on the display. See the IC-706 manual for a display showing all of the hard icons.

Communication from the RADIO back towards the HEAD consists simply of the data states for each of the hard icons. Also included are single bits controlling the status of the front panel LEDs (e.g., the LED color behind the P.AMP/ATT button). Finally, there is ASCII data for the character portions of the display (e.g., Frequency, Channel #, etc.). I will refer to the 3 text areas as FREQUENCY, CHANNEL, and MENU. The first two are self-explanatory. The MENU area is the text directly above the software keys (e.g. F-1, F-2, etc.).

Codes sent by the HEAD towards the RADIO

The button press data is sent out whenever a button is pressed. As an example, when the P.AMP/ATT button is pressed, the HEAD sends out

Fe 01 08 Fd

followed by a “close” command

Fe 01 00 Fd.

The RADIO does not return any ACKs but it will set the LED bit to correspond to the current state of the RADIO unit in the next response packet. When the VOLUME control is adjusted, the HEAD sends out

Fe 05 < *position data* > Fd

where *position data* is the new setting byte from 0 to 255. A summary of the command set is given in the appendix.

Codes sent by the RADIO towards the HEAD

Remembering that the RADIO only sends data to update the panel LEDs, text, and the display icons, the RADIO unit sends a 51-byte packet back to the HEAD (including the Fe ... Fd headers). All of the display information is included in these 49 data bytes. The exception to this is when the display goes into the graphics mode (for example to display SWR). I made no attempt to document the graphics data, but care

must be used when the RADIO enters the graphics mode, as the data packet length will change. The interface program must be able to handle this. Parsing details of this data packet are given in the appendix.

Radio system startup

If operation without a control head is desired, startup is a bit tricky because of the timing. If the HEAD is powered up without connection to the RADIO, it eventually settles into a series of Keep Alive codes (Fe 0b 00 Fd). I have discovered that the following sequence sent by the HEAD emulation software results in a successful start:

< Continuous string of Keep Alive codes > < assert PWK ground > Fe F1 Fd Fe <13 data bytes> Fd

Where the 13 data bytes are initial HEAD conditions (including the current settings of the VOLUME and SQUELCH controls). Fe F1 Fd seems to be a LOHIC code (Look Out, Here It Comes). After the start, the RADIO simply periodically sends the “51-byte packet” with the display information and the HEAD sends the Keep Alive packets.

