

Ham 37 – CI-V MOD Computer Control & Audio Hub, Python

Dr. Marc & Rosemary © 220312

1. The CI-V manual or data (MOD) computer control & audio hub is for radios without a sound card and data ports.
 - a. The code gives the hooks for control from the keyboard or from I/O pins.
 - b. The design is for the Raspberry Pi on Linux.
 - c. The setup for many applications will be on a Windows machine.
 - d. Consequently, the code is in Python.
 - e. The code outline includes both input and output from a radio to a computer.
2. Output/input control data stream arrives through a USB adapter with FTDI 232 chip.
 - a. The chip has a standard Windows or Linux driver.
3. Open a separate small screen for ‘CI-V MOD’ radio control.
4. On the screen, have an entry block to change the radio command address for the data stream.
 - a. Each type radio is different. IC2730 = 90h.
 - b. The data command will need the address.
 - c. Default to last address.
5. Send a data command to read the radio for present frequency.
 - b. Display on screen.
6. Send a data stream to the radio.
 - a. When spacebar pressed, send a PTT command to the radio.
 - b. Show an indicator on the screen.
 - c. When spacebar is pressed again, release the PTT.
 - d. Change indicator on the screen.
 - e. On release, send a tone, ping, or other very short sound to the speaker.
7. Other commands will be used later, but this is proof of concept testing to show our interconnect wiring is working.
8. On the screen, create a block which will display the command stream for trouble shooting purposes.
 - a. Display the last two exchanges of send and response.
9. Provide an exit.
10. When integrated to the Pi, another program will be running that will set one of the GPIO pins.
 - a. This program should detect the pin state.
 - b. When that pin is active, send PTT command, just like our spacebar capability above.
 - c. Keep the spacebar input, but also detect the pin, to send the same commands via the FTDI.
 - d. If unfamiliar with Pi GPIO, maybe I can dredge that up from archival memory.
11. The control commands are for Icom CI-V data.
 - a. Although these are Icom commands, other vendors, including Yaesu, adopted the same architecture.
 - b. A good resource, including intro drawings, is in back of pdf file which lives at <https://www.icom-france.com/uploads/files/produit/not-IC-2730-Web-en.pdf>
12. The protocol is a critical item to make a widespread, stable network, using diverse rigs for our communications group.
 - a. The process has extensive application for ham projects of many stirpes.
 - b. Is that ever cool?
13. Python screen interface:
 - a. Inputs set: radio address (90h), computer address (e0h), baud (19,200), 232 serial port (1-6), audio serial port (1-6)
 - b. Configure: frequency (146.047), node system (A, E, M), node number (188087), et al
 - c. Input control: spacebar or pin1, a/A, z/Z, pin 2-3-4
 - d. Display control: spacebar or pin1, a/A, z/Z, pin 2-3-4
 - e. Display message: send & receive command stream.
14. Distribution:
 - a. Michael: Screen displays and operator i/o.
 - b. MOD: Data stream for radio operation.
15. The illustration is the internal connections of the RS-232 cable. The USB interface is a FTDI 232 chip. The diode is a 1N4148 to connect TXd and RXd on one conductor.
16. Life is good. Enjoy

