

Wireless Cat connection for TS-50s

After played a while with ESP12e WiFi microcontroller I got idea to use it for connecting TS-50s cat to computer. TS-50s does not have proper connector, just TTL- level at digital unit pcb inside the rig.

There are several way to connect TS-50s to PC. At the time RS-232 com ports were used connecting was made with Maxim 232 chip, or similar.

Now, at USB bus time, cheap Chinese usb2ttl converter can be used.

Both of them have one disadvantage. They do not have galvanic isolation between PC and rig.

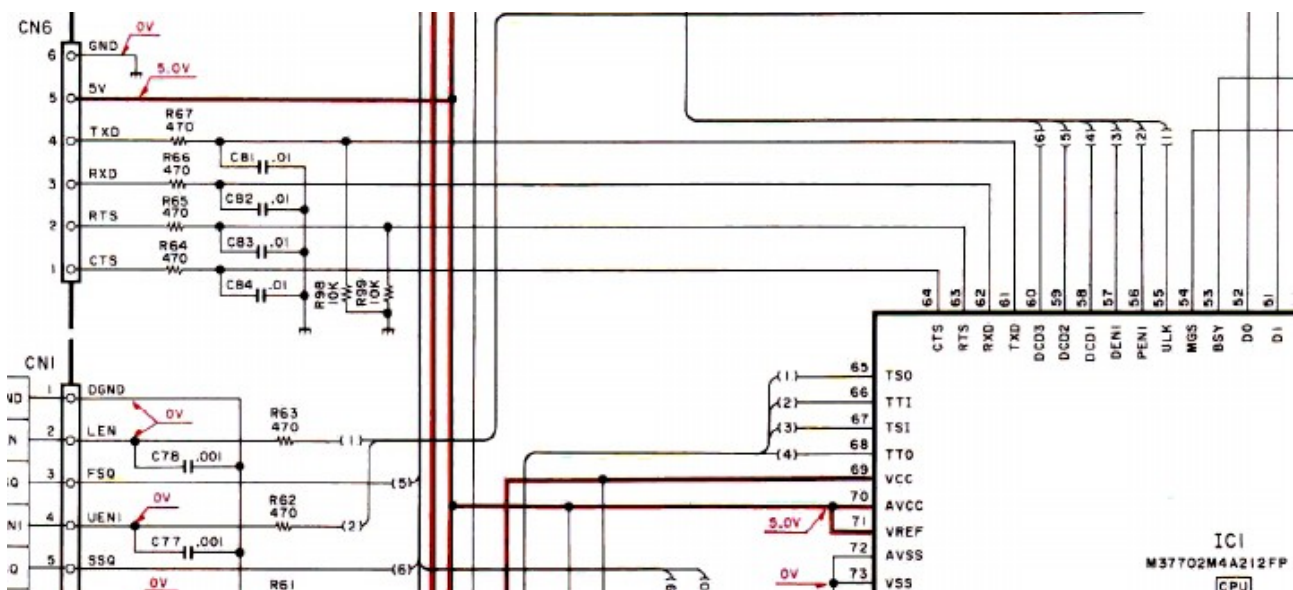
By using WiFi network isolation is guaranteed.

TS-50s socket CN6 does carry also 5V supply that can feed ESP12e. However there is one problem: Rig uses 5V and ESP12e is 3.3V device.

Small regulator LM1117, or similar, can do the job to drop Vcc to right level.

However data from rig comes still at 5V level. Simplest way to fix this is to use 3/2 resistor divider. ESP12e is not very sensitive for low voltage 1-levels, but 0-levels should be very near of GND.

On the other hand 3.3V levels that ESP12e TXD pushes to rig is directly acceptable level for TS-50s.



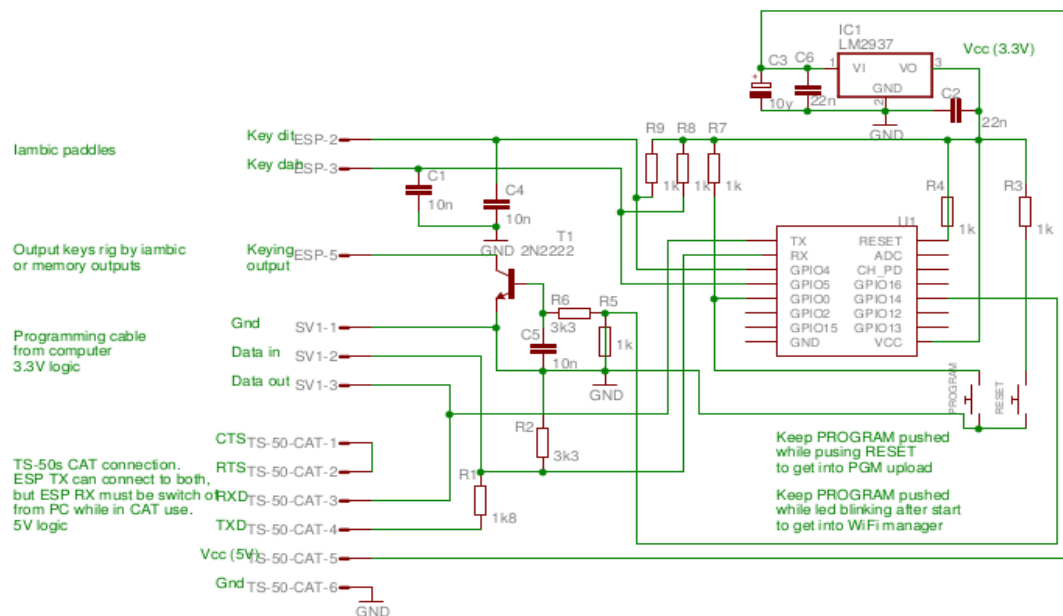
Data direction can be found from service manual. Lines that have pull down resistors of 10k are transmitted from rig to pc.

Rig is receiving commands with lines without pulldown resistors. Sounds funny, but so it is.

RTS and CTS should do some handshaking, but in tests I found out that best response can be found if they are just connected together and nowhere else.

ESP12e transmit data goes to rig's receive data and vice versa. 5V feed regulator and ground is common for supply voltage and data.

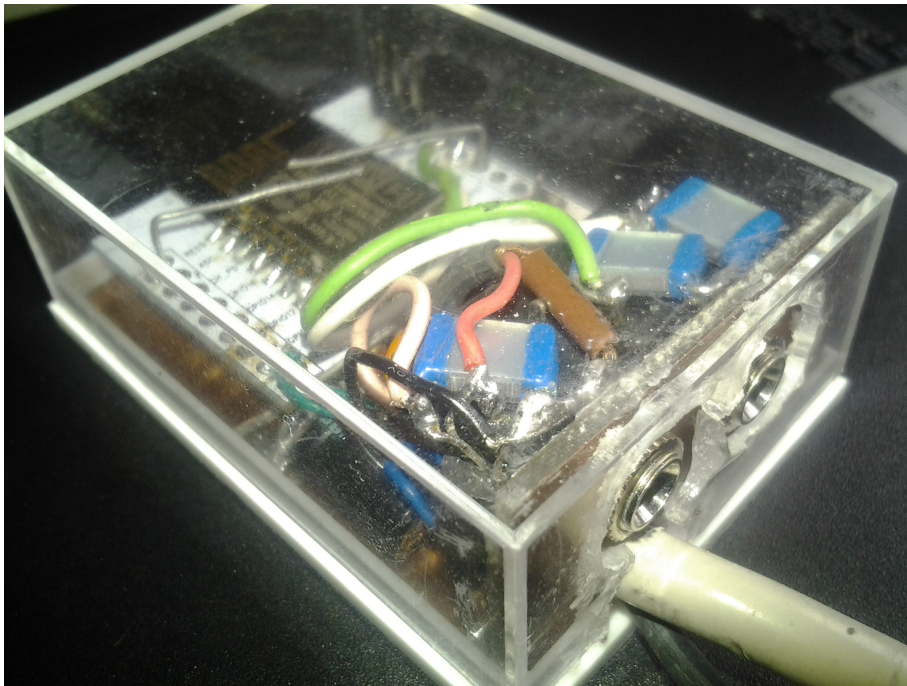
Schematic diagram



From diagram all connections can be seen.

It also shows that iambic paddes are connected. YES: ESP12e has also cw-keyer !

Not very excellent, but usable.



Prototype was built in ring box. (Consult your YL/XYL!)

3.5mm jacks are for
iambic paddles and stright
key.

Black wire goes to 3.5mm
plug that connects to
TS-50s CW key socket.

Grey wire goes to TS-50s digital unit board CN6 socket.

I think this all could fit also inside the rig. Just the CW key input should then to be changed to stereo type.

Note the program and reset buttons that are made from resistor legs. Works well !



Program

At this phase program is at Beta state, but shown to work with wsjt-x and cqrllog (linux versions).

The ts50v2 emulates rigctld commands from TS-50s. In addition it has iambic keyer and supports also rigctld commands "b", "\send_morse" and "L KEYSBD", set key speed level". Default speed is 22WPM.

This tested with cqrllog by selecting "hamlib keyer" at same TCP port as rigctld itself.

Supported commands can be found from source file command_parser.ino.

Consult also hamlib/rigctld man page found with Google using words: rigctld commands

Program supports some "get" commands grouped together, but mostly only one command per line that differs from "real rigctl". Cqrllog uses command "fmv" to get frequency, mode and vfo at same go. This works fine.

At program start blue led is blinking and if "program"-button is pressed then stays on for 3 minutes. During that time you will find a new WiFi network named "TS50s". Connect to that (no passwords) and type any address to your web browser. You are directed to WiFi setup where you can browse and connect local WiFi networks.

If connect is successful blue led sends CW "R" if not "N" and restarts ESP12e. After restart TS50s is a part of successfully connected WiFi network.

If you do not have WiFi network you can use "reset" at setup and ES12e forgets previously defined Wifi (if any) and works as standalone AccesPoint with IP address 192.168.4.1

You can then telnet 192.168.4.1 4532 to get control to rig, or use that address/port at your program (cqrlog, wsjt-x, etc.) settings.

As said this is BETA and has no guarantees to work.

Im also planning to write TS50v1 program that is simpler.

It is just TCP2serial server that could be used by creating virtual serial port with program "socat" in Linux. Windows has also similar com2tcp programs, but I do not know anything about them as there is no windoze PCs in this house.

Additonally iambic keyer will be added and also implementing of KS and KY kenwood commands for setting morse speed and send string.

That was the first goal (v1), but rigctl emulation was easier (??) to implement so v2 is ready first.

Programs are GPL2 open source so feel free to share them and make improvements.

—

Saku

OH1KH