Yaesu FT-991(a) external meter display and Tune button (By PA0LUX)

**See also You Tube: https://www.youtube.com/watch?v=XeM1ff3EJcw**

Note that the Menu/Settings #28 has to be set to RS232C

Note that Menu/Settings #29 has to be set to 19.200 bps

Note that you may want to change the rotation of the display (upside down) as a result of your mounting method. If so replace the “3” with “1” in this line of code in the .ino file : tft.setRotation(3);

This is a design of an Arduino (Nano) with a 2.8“ TFT with ILI9341 chip acting as a simultaneous (bargraph-)display of all 7 meters available in the FT-991(a). Also, a 3.2 “TFT with the same resolution (240 x 320) and ILI9341 chip will work fine. Touch is not used.

It reads all meter settings from the radio via CAT, through the Rs232 connector.

The display will also show what power has been set in the radio. Besides that, it has a (momentary) pushbutton that can be used to tune the radio with 20 Watts (adjustable in .ino file).

The following data are displayed:

- SWR

- Comp

- IDD

- VDD

- ALC

- S Meter - is substituted by Power out when in TX.

- Power out - instead of S Meter, when in TX

- Present power setting

- Connection status

At tune button press, it stores the present power setting and the present mode. It will then engage an FM-N transmission. Note that this is a CAT command, the indication on the transceiver will be FM.

When released, the tuning signal will stop. The previous power and mode setting is restored after the button has been released. The tune power can be adapted in the sketch with the variable; **set\_tune\_pwr "PC020;"** The 020 means 20W. You can change it to anything between 005 and 100.

The display shows a green “LED” to indicate that there is correct communication with the transceiver. The green led turns on when there is a correct answer received from the radio. It does check for correct baud rate. When not connected or when the transceiver is powered off (and 13,8 V still present), or a wrong baud rate, the green led will turn into a red one.

The circuit is to be connected to the RS232 connector of the transceiver (thru a null modem cable), thus leaving the USB connector of the transceiver available for other logging and/or CAT software. The Arduino cannot directly be connected to the transceiver. The FT-991(a) talks RS232 and the Arduino talks TTL levels. Therefore, a simple Rs232 – TTL level converter is used, widely available for about 2 euros. It comes complete with a 9 pin Rs232 D-connector.

It is advised to use a connector at least at one end for the connections between Arduino and Rs232 level converter.

Note; Do NOT connect USB and RS232 (of the Arduino) at the same time, it will not work (will do no harm either). Also, better not to connect 13,8V and USB at the same time.

To program the Arduino is very simple. Do not program while the Rs232 level converter is connected.

Make sure you have the Arduino IDE (downloaded and-) installed on your PC. This is the editor for sketches. Connect your Arduino via USB to the PC and check which new port has just shown up in your computer. Now in the Arduino IDE choose via “Tools” menu this port and choose the correct Arduino board (in my case Nano).

The program uses 3 libraries: SPI, Adafruit\_GFX\_Library-1.11.5 and Adafruit\_ILI9341-1.5.12. Make sure you add these to your library (if not already present) using the Arduino IDE.

Copy the correct sketch (. ino file) to your PC and create a folder with exactly the same name as the copied file. Now move the copied file into the new folder.

Now in the Arduino IDE go to “Files” menu, choose open and navigate to the .ino file and open it.

Press the “right arrow” in the IDE. The program will be uploaded to the Arduino. It will indicate when ready.

Set the Rs232C rate in the FT-991(a) to 19200 bps (see notes above this article). Connect the cables and everything should work now.