CAT Controlled Fract-N Synthesizer

Overview

The unit is a PIC based interface that takes in serial Computer Aided Transceiver (CAT) commands and translates them into the register values to programme an LMX2541 Fractional-N synthesizer. An IF offset can be applied to the commanded frequency so the synthesizer can function as a local oscillator in a converter or receiver. An LCD shows the frequency and the IF selected. The module was designed to emulate an Icom IC746 transceiver, and uses Icom's CI-V protocol. However, subsequent tests when used in full transceive mode suggested that emulating an IC756 was a better solution when interfacing to the WSJT-X software Sufficient CI-V / CAT commands have been included so that the WSJT-X software will recognise the module (as an Icom transceiver) and can set it for Doppler tracking. Dummy, 'bland' responses are provided for the additional requests WSJT-X sends to the transceiver. No Tx/Rx switching is implemented, so if this is needed, the frequency control will have to be "faked".

The simple CI-V set used means it will also emulate most Icom transceivers, although this has not been tested and the CI-V address will have to be changed. A couple of CI-V commands have been "hijacked" for purposes of storing controller information.

The controller works in 1Hz steps and allows the LMX2541 synthesizer [1] to be set to any frequency a particular chip variant can generate. Above a few hundred MHz, permitted frequency limits are determined by the variant used. For frequencies lower than about 450MHz, any frequency can be generated whatever variant chip is used. Illegal commanded frequencies (those the chip variant can't manage) are detected and a 'fail' response echoed back on the CI-V interface as well as showing on the LCD.

Circuit Details

The circuit diagram appears in Figure 1 which shows that little more is needed other than the PIC and LCD module. To ensure accuracy of serial interface timing, the PIC uses a crystal controlled clock. The serial interface shown is Icom's single wire bidirectional CI-V port which connects via a 3.5mm mono jack connector, but there is no reason why Tx and Rx signals should not be separated. For connection to a standard RS232 port an inverter and (possibly) a level shifter like a MAX232 will be required. 9600 baud is used, but this can be changed in the PIC firmware to other rates supported by the on-chip USART.

A link, on pin 6 of the PIC, allows echo-back of received characters. If the link is installed, any character received from the controller is immediately sent back. If the link is left out and pin 6 open circuit (it is pulled high internally) no echo-back is provided.

This option is provided for circumstances where direct connection of the TXD and RXD lines to the controller's port is used, eg via a MAX232 interface. A 'true' CI-V interconnection with its single wire plus ground automatically provides loop-back via hardware. The CI-V protocol demands that this loop-back be present, as it forms part of the collision detection procedure for multi-users on the bus. So if a real CI-V interface is in place, the link should not be installed. For non-CI-V wiring with direct three wire interfacing, the link should be in place

A four line x 16 character LCD module with standard parallel interface [2] offers maximum flexibility. For normal operation only the top two lines are used, so a 2 line display will suffice.

The contrast of the display can be adjusted by varying the value of the resistor from pin 3 of the LCD to ground (shown as 1k5 in Figure 1)

PIC Code

This is contained in file *CAT_Ctrl_LO.asm* [3]

After switching on, or resetting the interface a start up delay of 5 seconds has been included so that ovenned oscillators and TCXOs used as the synthesizer reference can stabilise before and data is sent to the synthesizer chip. If the chip is programmed before the reference appears, the internal VCO calibration process will fail and phase lock will not occur. After this startup delay, the stored frequency is loaded into the synthesizer; either the DEFAULTFREQ defined at compile time, or any value that has been subsequently stored using the '09' CI-V command (see below) .

The PIC's internal USART is used for serial communication. The default baud rate is 9600Hz with a 4MHz crystal. This can be changed, if desired, by altering the value programmed into the SPBRG register within the routine labelled *StartUp* . See the 16F626 data sheet for details of how to calculate values for other baud rates.

Several CAT commands are supported. Command '05' (hex) which sets the frequency and command '03' which reads it are the most important for correct operation. Commands '04', '07' '1A', '1C' are also decoded and send 'bland' responses since the WSJT-X software sends these and expects replies.

Three additional commands have been added for convenience in setting up and controlling the unit. Command '09' (write current memory) saves the current frequency to EEPROM and is recalled next time the processor is reset.

Command '0D' is specified as "Duplex Offset" in the normal CI-V suite and is used here, (in conjunction with command '0F') as a convenient short-cut to avoid using a PIC programmer to change the IF offset. It accepts a six digit value in kHz as the IF offset which is stored in EEPROM. Any new value saved using this command will always be regarded as positive until it is changed using command '0F'. *It is not possible to send any frequency resolution better than 1kHz. If 1Hz resolution offsets are needed, the full four-byte value needs to be inserted into the .ASM file at compile-time.*

Command '0F' is a multi-function one with sub-commands, normally used for setting duplex and split operation. Here, only subcommands '11' and '12' are recognised and are interpreted as setting the IF offset negative (high-side LO) or positive (low LO) respectively.

Customisable Constants

Several constants first need to be defined; these appear at the beginning of the PIC source code and are shown here.

CIVADDR is the CI-V address. 0x56 is the default for the IC746 and WSJT-X assumes this when the IC746 is specified. Use 0x50 for the IC-756.

VCOMIN and *VCOMAX* are the lower and upper limits of the VCO for the particular variant of LMX2541 chip used, specified in Hz. For the examples below they are 2810000000 Hz and 3230000000 Hz which are the datasheet values for the LMX2541–Q3030 variant.

FREF is the comparator frequency to the phase detector. As the R, or reference, divider is set to a value of unity, this number also becomes the reference input frequency .
After dividing by 1000000 (as a constant, at compile time) the value rounded to the nearest MHz is sent to Reg4 for the VCO auto-calibration.

FIF is the IF offset and can be a positive or negative value. A negative value (high side LO injection) is stored as, for example, -d'10701500' It can be modified over a limited range of values with the "set duplex offset" CI-V command. The sign can be changed using the CI-V "Set offset direction" command

DEFAULTFREQ is the startup frequency. It can be modified in EEPROM by the CI-V "write current memory" command

The two Debug flags (only one may be used at a time) should not be set for normal operation as they slow things down and make the resulting PIC code larger. However, they can be useful to see the four values calculated for any demanded frequency and set of constants, or the resulting Reg4/2/1/0 values sent to the chip. In each case, the values appear on lines 3 and four of the LCD.

```
        CIVADDR             =       0x56            ;Zero for auto set

        VCOMIN      =       d'2810000000'   ;Values for Q3030 variant
        VCOMAX      =       d'3230000000'   ;
        FREF        =       d'10000000'     ;Reference

;These constants placed into EE or RAM
        FIF         =       d'28000000'     ;IF Frequency   Hz
        DEFAULTFREQ =       d'144000000'

;Compile time debug flags
        DebugFractN =       0               ;Display N, F, D, Odiv on LCD lines 3/4
        DebugReg    =       0               ;Display Synth registers 0,1,2 ,4 on LCD lines 3/4
```

The demanded frequency arrives on the CAT interface as a BCD value in units of 1Hz. Once the frequency is decoded it is converted to a 32 bit binary value and kept as the *Freq* value stored in PIC registers *Freq3 / 0* and used for the calculations described below. The demanded frequency is shown on the LCD line 1, and the IF appears on line 2 (to 1kHz resolution)

CI-V command 03 reads back the value stored in these *Freq* registers after converting to the correct BCD format needed for the CI-V protocol.

Fract-N Synthesizer Calculations

The calculations are performed in subroutine `CalcRegisters` . The IF offset stored in EE memory as a signed 32 bit binary value is first subtracted from the *Freq* value (a positive offset therefore means low-side LO injection). The result gives the actual output frequency, F_{OUT}, from the synthesizer chip. A number of calculations are then performed to get each of the four values needed for the Fractional-N Synthesis.

First, it must determine the value of RF output divider that allows the VCO to operate within its correct band. The following algorithm is used in subroutine `CalculateRange` :

A = INT(*VCOMIN* / F_{OUT}) + 1 B = INT(*VCOMAX* / F_{OUT})

If A > B then an F_{OUT} value has been requested that cannot be generated by that chip variant. A flag is set to allow the firmware response to illegal frequencies and no more calculations are performed..

If A = B, this is the only value of output divider that will permit the requested frequency to be generated. The result is stored in *OpDiv* and subsequently used to set the output divider bits in Reg4 of the LMX2541

If B > A then more than just one value of output divider will allow the VCO to stay within its allowed range.
Here, the average of the two is used, generated from *OpDiv* = (A + B) / 2. It may be possible in later versions of the firmware to select an optimum value that gives best spuri levels.

The denominator , or D register, of the Fractional-N process is then calculated from :

D = *FREF* / OpDiv , The resulting value is stored in register D2/0 and sent to Reg2 of the LMX2541. This results in 1Hz steps at the output frequency

The VCO frequency is then given by F_{VCO} = F_{OUT} * *OpDiv*

The value of the integer divider, N, is given by the integer part (the quotient) of F_{VCO} / *FREF* . The result is stored in registers N1/0, later used for setting the bits in Reg1 and Reg0 of the LMX2541

The *Remainder* of the division is then converted to the Fractional, or F value using a convenient shortcut. Since we know the output step size is 1Hz as this is was defined via the D register calculated earlier, the VCO grid must therefore be numerically equal to *OpDiv* Hz. So the Fractional, or F, value is conveniently given by *Remainder* / OpDiv. The result is sent to registers F2 / 0 for subsequent transfer to Reg1 and Reg0 of the LMX2541.

Register Calculations Worked Example.

FREF = 10000000 (10MHz)
VCOMIN = 2810000000 (LMX2541 - Q3030)
VCOMAX = 3230000000 " " "
FIF = 28000000 (28MHz)

FREQ = 144000000
F_{OUT} = *FREQ* – *FIF* = 116000000 (116MHz)

A = INT(2810 / 116) + 1 = 25 (MHz used here for clarity)
B = INT(3230 / 116) = 27

A is less than B, so this frequency is permitted.
Take the mean of A and B : *OpDiv* = 26

Denominator D = *FREF* / OpDiv = 10000000 / 26 = 384615

F_{VCO} = 116000000 * 26 = 3016000000

N = F_{VCO} / F_{REF} = 301 with a remainder of 6000000

F = 6000000 / 26 = 230769

So we now have values for the four registers essential to the Fract-N frequency generation :

OpDiv = 26 N = 301 D = 384615 F = 230769

Back-calculate to double check :

F_{OUT} = 10MHz * (N + F / D) / OpDiv = 116MHz (+ 28 = 144MHz)

Transceive Operation

The synthesizer can be used as the LO for a transceiver, for example an IC202. (In this instance, an IF offset in the region of 10.7MHz would need to be specified).

For transceive operation to work, the frequency has to be properly reprogrammed during Tx and Rx cycles allowing full pseudo-split operation. However, the controller has no knowledge of Tx/Rx, or PTT state – it responds only to frequencies sent to it via the CAT interface.

Practical testing has shown that WSJT-X can get confused when some transceiver emulations, are used in the software, such as the IC746 originally chosen. This comes-about because the 0x1C command to control Tx/Rx does not actually report the proper Tx state of the transceiver, but replies as if always in Rx mode. This can cause problems when in QSO using full Doppler control of Tx and Rx frequencies.

The solution is to emulate a transceiver that does not implement CAT controlled PTT operation. The IC756 (CiV address 50) is one such and if full Tx/Rx operation is needed, this is a suitable emulation option.

One option that could be could be considered would be to use the spare port B3 on the PIC to monitor a transceiver's PTT line, and report back properly after a 1C command. Future firmware may have this option added, optionally activated depending upon a compile-time flag.

CAT / CI-V Command Format

Below is a description of the CI-V codes used for this controller. All are shown first as the command from master controller to PIC interface, followed by the reply. All values are hexadecimal. The target address in the examples is 0x56 and the controller's address is the default 0xE0.

Request Frequency
FE FE 56 E0 **03** FD reply FE FE E0 56 03 56 34 12 44 01 FD (144.123456 MHz)

Set frequency (144.123456 MHz) :
FE FE 56 E0 **05** 56 34 12 44 01 FD reply FE FE E0 56 FB FD or FE FE E0 56 FA FD
FB in the response is replaced by FA when an illegal frequency has been requested

Save current frequency
FE FE 56 E0 **09** FD reply FE FE E0 56 FB FD

Set and store IF offset , kHz (28MHz) :
FE FE 56 E0 **0D** 00 80 20 FD reply FE FE E0 56 FB FD

Set Negative IF Offset
FE FE 56 F0 **11** FD reply FE FE E0 56 FB FD

The following are included to allow WSJT-X to recognise the transceiver, but have no action on the Fract-N synth controller. Returned values were gleaned from experiment. Other commands may be added later to support different transceivers.

Read Display Mode
FE FE 56 E0 **04** FD reply FE FE E0 56 04 91 91 FD

Read IF Filter
FE FE 56 E0 **1A** FD reply FE FE E0 56 1A 03 02 04 FD

Tx/Rx control
FE FE 56 E0 **1C** xx FD reply FE FE E0 56 1C 00 00 FD

Set VFO B
FE FE 56 E0 **07** xx FD reply FE FE E0 56 07 00 00 FD

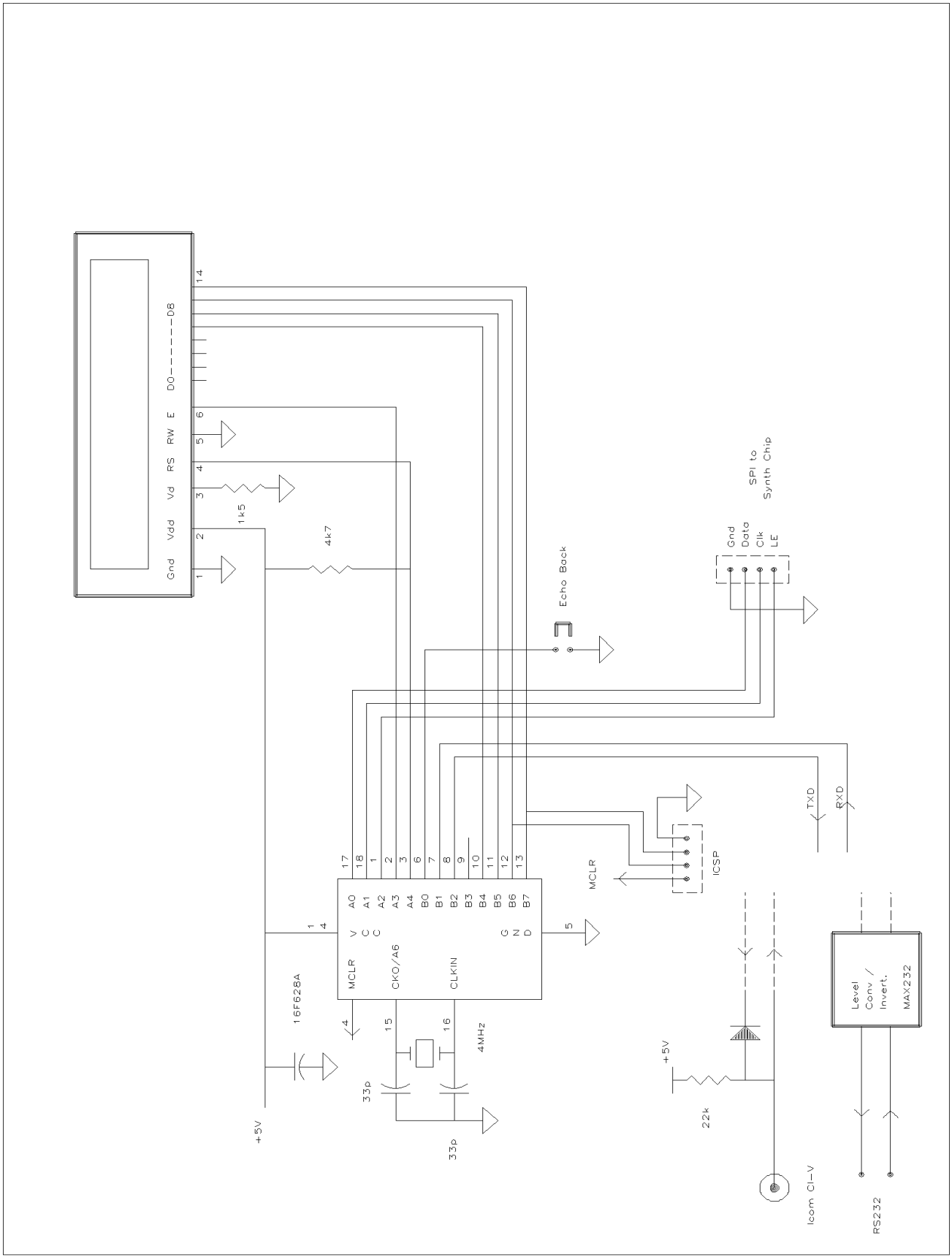
References

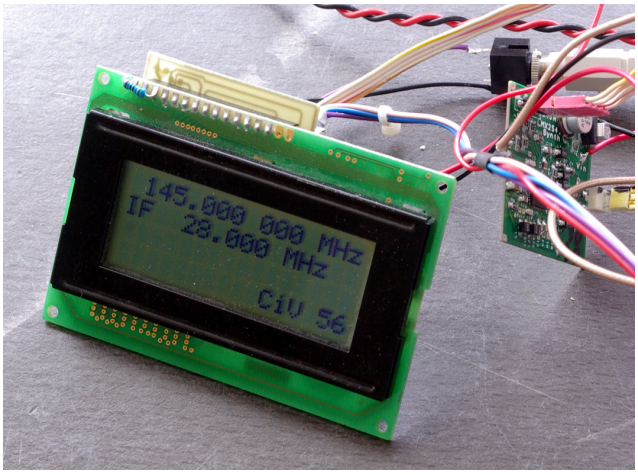
[1] LMX2541 Fractional-N Synthesizer module
http://www.g4jnt.com/LMX2541_Synth_Module.pdf

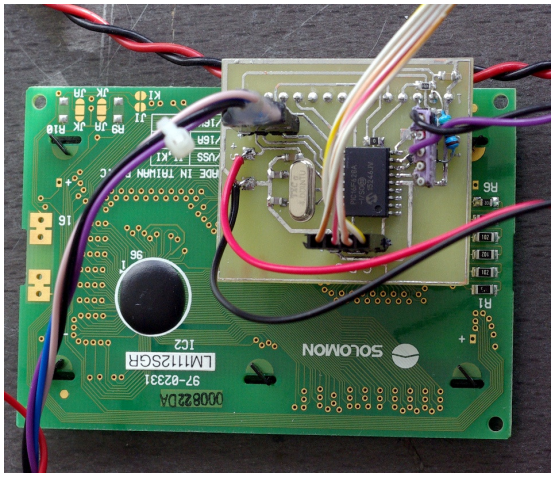
[2] Suitable low cost displays are available from Kevin Avery, G3AAF
kevin.avery@tunstall.com

[3] PIC Firmware, assembler format only http://www.g4jnt.com/CAT_Ctrl_LO.asm

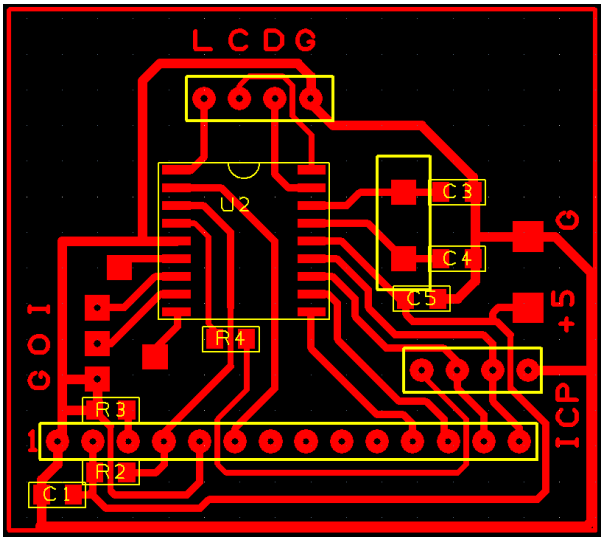
Figure 1 CAT controlled Synthesizer Circuit Diagram



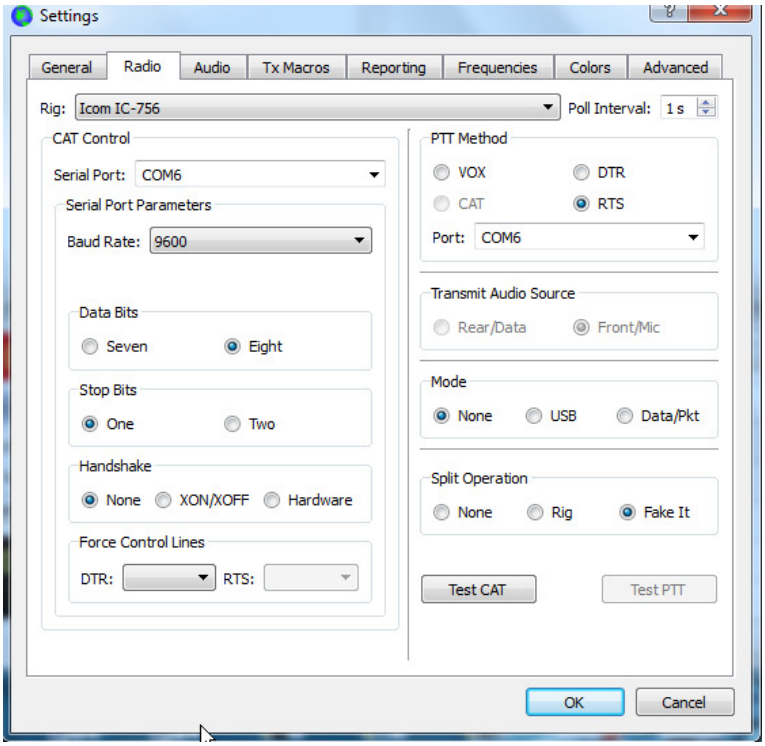




Two views of the PCB AND LCD



PCB Component Placement



WSJT-X Radio Settings Window for use with the controller