CROSS-CONNECTING MMTTY with DX4WIN - by K8UT v4 (13 JAN 2004)

The following document describes the steps required to interconnect the DX4WIN logging program to the MMTTY RTTY program inside one computer, so that DW4WIN now has a full-featured RTTY program lashed onto it. All program control is through DX4WIN, so all you have to do is start MMTTY and run it in "headless" mode. This means that the RTTY scope and waterfall come from MMTTY, but RTTY text display, logging and keyboard macros all work through DX4WIN, giving you the ability to use DX4WIN's %keys for callsign, signal report, name, etc.

Cross-connecting MMTTY and DX4WIN requires a sound card and at least one serial port, depending on whether you interconnect the two software programs using software interconnect of hardware interconnect.

HARDWARE INTERCONNECT: This method requires three serial ports. Two of the serial ports are used with a null modem cable to allow DX4WIM to communicate with MMTTY. All you need from the third serial card is DTR/RTS - which is how MMTTY keys the rig into transmit/receive mode. (I also have a keyed line from my LPT port - I use this to key the transmitter in CW and PSK modes from within DX4WIN)

SOFTWARE INTERCONNECT: This method only requires one serial port, from which you connect the DTR/RTS line to key the rig into transmit/receive mode. Rather than install two hardware serial lines for the interconnect, you install a free software program to emulate the two lines. You can download the program from http://www.mixw.net/related.htm . Unfortunately, the program only works on Windows 2000 and XP, so if you are still on Windows 9x or ME, you must use the hardware interconnect described above.

Both the hardware and the software interconnects have been demonstrated to run on a homebrew 866 MHz Pentium III PC. Since running these two programs simultaneously requires some significant processing power (graphics, I/O, sound card, timed interrupts) slower PCs may be incapable of keeping up with the real-time processing demands of RTTY signals.

I created new DX4WIN keyboard macros to talk to the TNC emulator in MMTTY. I have another document that describes my DX4WIN keyboard macros.

-larry
K8UT

VERSION INFORMATION:
= = = = = = = = = = = = =
DX4WIN 5.02, and MMTTY 1.62

HARDWARE ARRANGEMENTS:
= = = = = = = = = = = = =
To keep my explanation simple, I will refer to COM1 as the MMTTY keyed line to the rig, COM2 as the MMTTY RTTY serial line, and COM3 as the DX4WIN RTTY serial line. If you are a DX4WIN user, you probably have another connection for CW or PSK keying your rig from DX4WIN - that connection has no bearing on this configuration (I happen to use the LPT port method).

COM1: Build a DTR/RTS interface cable (using the single transistor circuit in the documentation of either DX4WIN or MMTTY) that will connect your computer's COM1 port to your rig's MIC PPT plug or an ACC socket. This cable will shift your rig to transmit/receive based on signals from MMTTY (remember, MMTTY is the back-end part of this equation)

COM2 and COM3: Build or buy a "null modem" cable. (there are many references on the Internet for building such a cable, depending on whether your serial ports are 9 pin or 25 pin) Plug one end into your COM2 port, the other into your COM3 port. This cable is the mechanism whereby DX4WIN and MMTTY send information to each other.

SOUND CARD IN/OUT:
= = = = = = = = = = = = =
Connect the "line out" or "speaker out" from your sound card to "mic in" on your rig; and the "mic in" on your sound card to "audio out" on your rig. This connection is well documented in both the DX4WIN and MMTTY references. Some folks think you need transformers and dropping resistors for impedance matching. I use a straight connection between the rig and computer, and adjust the audio levels with the Windows volume controls. You will have to do some experimenting to get this the way you like it. If you are already running PSK from DX4WIN, you are all set. If you are NOT already running PSK, you'll be able to after you get this running. HOWEVER, keep in mind that it is MMTTY, not DX4WIN, that "talks" to the sound card for RTTY. MMTTY is the back-end of this mode.

NOTE: There are many interface "kits" and products that range from $25 - $150 that will connect your keyed lines and sound cards. Although I built my original interface from Radio Shack parts, it was a pain. Later, when I decided that I wanted to run multiple modes, and RTTY FSK, and clean up the mess of wires behind the bench, I decided to purchase one of the more expensive commercial interfaces.

DX4WIN SETTINGS:
= = = = = = = = = = = =
You must configure DX4WIN to talk to the COM3 port, and the com settings must match those of COM2 that MMTTY cross-connects. These settings are found in DX4WIN's <F>ile, <P>references, RTTY.

RTTY interface:
Type : regular
COM Port: COM3
Baud rate: 4800
Flow control: RTS/CTS
Data/Parity: 8 bits/no parity
Scrollback capacity: 500
Options:
X check for Dxspots
_ announce DX to other window
X Always in immediate mode (IMPORTANT!)
X Use bell character'
X Show function keys

MMTTY SETTINGS:
= = = = = = = = = = =
You must configure MMTTY to talk to the COM2 port, and the comm settings must match those of COM3 that DX4WIN cross-connects. These settings are found under MMTTY's <O>ptions tab. There are lots of settings on these tabs - I will only talk about the ones that I changed, or that make a difference in setup.

<O>Ptions, <S>etup MMTTY, TX
PTT: COM1

<O>Ptions, <S>etup MMTTY, Misc

Sound Card Buff: 512 (IMPORTANT - setting this higher REALLY bogged down my computer)

For AFSK operation: Tx Port: Sound

For FSK operation: COM-TxD

<O>Ptions, <S>etup TNC Emulation(T)
Port definition
Port: COM2
Baud: 4800
Data length: 8 bits
Stop: 1 bit
Parity: None
Flow control: CTS
TNC type: TNC241
Local echo: After Sending
X Disable window (IMPORTANT - see text)
_ Stay on top

OPERATING:
= = = = = = = = =
For AFSK transmission, your rig must be in lower sideband mode. For FSK operation, you will need an FSK keying line to your rig, and the rig must be in RTTY mode.

Start DX4WIN first, then launch MMTTY. If you have the two serial ports configured properly, MMTTY will send a copyright notice and "Receive/Transmit" instructions to DX4WIN's RTTY window. You must go to the DX4WIN RTTY send window and press the <ctrl>R key to begin receiving RTTY text.

See my other document on function keys and macros.

To go to transmit mode, press <ctrl>T. To return to receive mode, type <ctrl>R or a "\".

LESSONS LEARNED:
= = = = = = = = = = = = =
1. The DX4WIN PSK window cannot be open when running MMTTY, because the two programs will fight for control of the sound card in your computer. When switching from RTTY to PSK, you must first shut down MMTTY before opening the DX4WIN PSK window.
2. DX4WIN RTTY settings MUST be in "always in immediate mode" for the macros to work properly
3. The MMTTY "Disable window" setting will minimize the screen space required by MMTTY, yet still allow you to see the tuning window. It also eliminates the confusion created when seeing two sets of receive and transmit text (one set from MMTTY, one set from DX4WIN). HOWEVER, during testing and debugging it is very helpful to see what text is flowing back & forth between MMTTY and DX4WIN - so you may want to leave this setting unchecked until you get the programs working properly.
4. In transmit mode, DX4WIN has a very limited transmit buffer with no <backspace> or <delete> key. So if you make any errors while typing you must either: send the mis-typed characters and hope the other operator can understand you; send XX to indicate an erasure and then re-type the text; or Press the <ctrl-R> key to shift to Receive mode (which empties the transmit buffer) then press <ctrl-T> to resume transmitting. None of these alternatives are very attractive, IMHO – which is why I operate MMTTY in stand-alone mode for everyday rag-chewing, and only use MMTTY/DX4WIN in cross-connect mode for contests.