RTTY Keyboard Adapter

Andy Talbot G4JNT Sept 2010

50 Baud Version added 1 Nov 2011

This adapter module decodes the output from a PS2 keyboard and generates RTTY Data at 5V logic level. It was designed to allow simple QRP RTTY transmitters to be built that work, for example, by shifting the frequency of a crystal with a varicap diode without having to have a PC on hand running software like MMTTY to generate the data.

The module has been designed so that it can run in parallel with the NUE-PSK Modem (which, although it works with RTTY mode, does not generate a suitable single data output line – only audio tones). By connecting the four wires of the keyboard interface in parallel on both units, the same keyboard can be used for transmitting using this adapter, and also for controlling the NUE-PSK unit. The 5V supply can even be extracted, thus making use of the NUE-PSK's efficient switch mode regulator. A subset of its control functions applicable to RTTY transmission have been incorporated into the adapter so, for example, function key F10 toggles Tx / Rx on both the NUE-PSK and RTTY adapter simultaneously.

No macros or callsign memories, just a simple, straightforward keyboard RTTY interface although an integral type ahead buffer with buffer fill warning and embedded Tx/Rx control does allow for fast typists.

Details

A PIC 16F627 decodes the characters typed on the keyboard and converts the scan codes generated into RTTY characters which are generated at 45.45 (or 50 Baud) baud with 5 Volt logic level (idle is logic '0'). The 50 baud version is for operation at VHF/UHF, Another logic level output is available for Tx/Rx switching on Port B3, PIN 9 – **this is not shown on the circuit diagram** although the pad is shown on the PCB layout. This is controlled using the same key sequence as the NUE-PSK, ie F10 for immediate toggling, or [ctl]-S and [ctl]-Q for embedded Tx-On and Tx-Off respectively.

An internal 64 deep type-ahead buffer has a LED illuminate when the buffer is half full and extinguishes when sufficient characters have been sent to reduce its contents below 50%. Letters / Figures shift is handled automatically and is fully transparent to the operator; another LED is illuminated while the signal is in *Figs Shift*. The Letters Shift character is sent when idling.

The circuit diagram is shown in Figure 1 with a small single sided PCB using surface mount construction in Figure 2. This includes a 6 pin mini-DIN connector for the keyboard mounted through holes along with the two LEDs and PINs for DC supply and the Data out and Tx control connections. The rest of the components, including the integral three-terminal ceramic resonator for the processor clock are all surface mount on the copper. There are two grounding links to be made. They are not absolutely essential, but do close-up grounding paths that would otherwise meander a bit too much, so may assist EMC / spurious issues. A four pin header is included for in-circuit PIC reprogramming.

The two photographs show my original PCB version – *but note that the position of the two output connections are reversed from that in the final PCB layout.*

The PCB copper layout is in the file ***KBDRTTY_Cu.pdf*** at 1:1 scale for PCB manufacture at home. The PIC firmware is in the file ***KBDRTTY.HEX*** Both files are included with this archive. ***KBDRTTY.GIF*** is a higher resolution of the circuit diagram of Figure 1

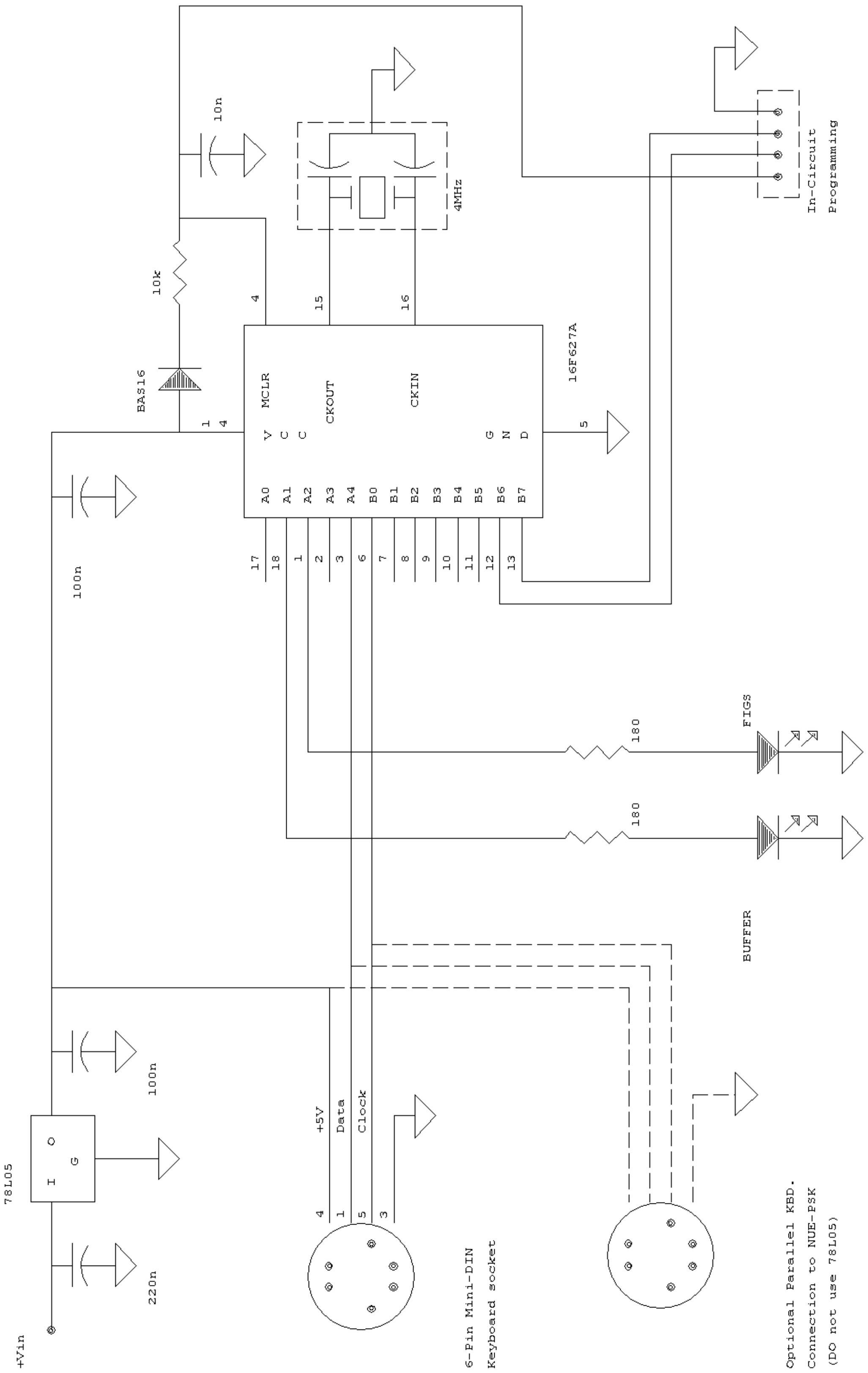

Figure 1 Circuit Diagram of the RTTY PC Keyboard Adapter

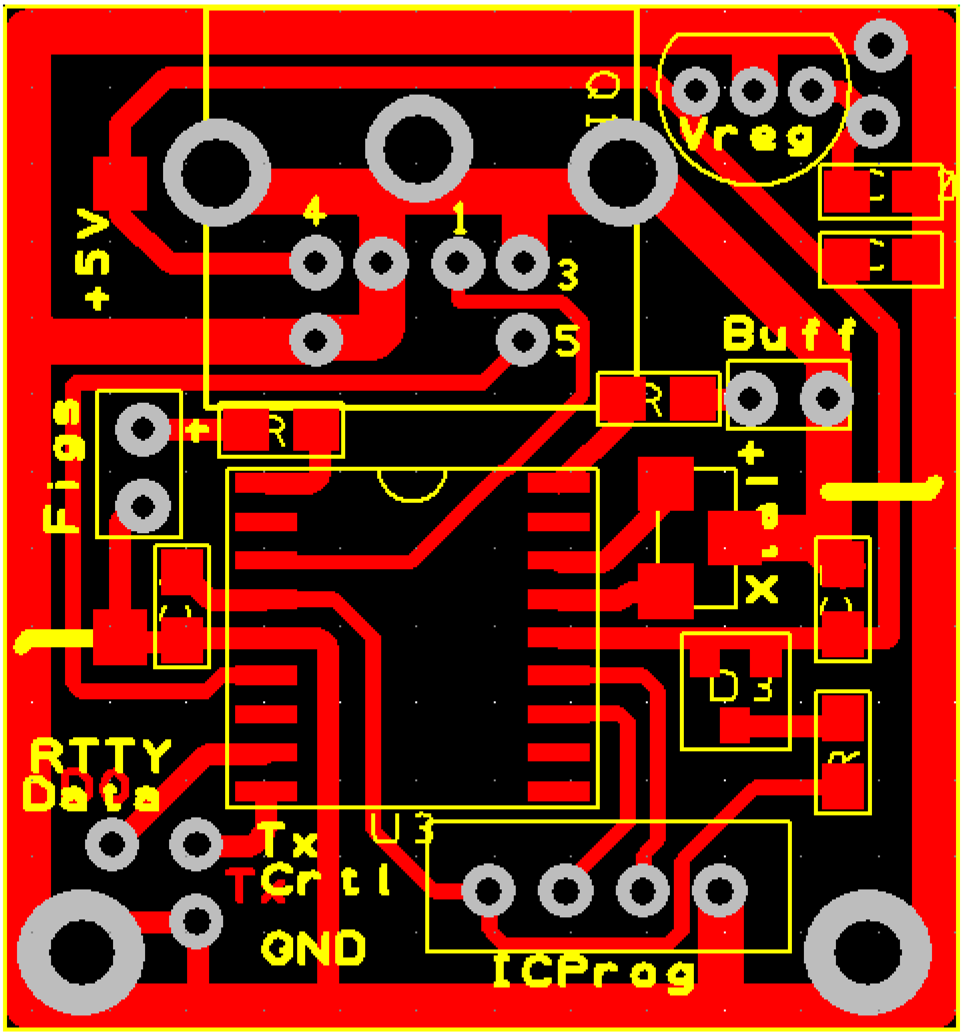

Figure 2 PCB Layout