Tynemouth Software

SOFTWARE SERIAL CARD FOR RC2014 AND MINSTREL 4TH

OVERVIEW

This is a serial interface for the Minstrel 4th using the RC2014 bus. There is no UART or ACIA chip involved, the serial data is “bit bashed” using an IO port to read and write to the serial port.

The output is 5V RS232, designed for an FTDI serial to USB adapter cable or similar.

The baud rate is dependent on the CPU clock and the following speeds are used in the supplied code:

Z80 Clock	Cycles Per Bit	Baud Rate
3.25MHz	56	57600
6.5MHz	56	115200
7.3728MHz	64	115200

The software required to drive this is explained in detail on the github

- https://github.com/tynemouthsoftware

PARTS LIST

CAPACITORS – CERAMIC RATED 6.3V OR HIGHER

6 x 100nF axial *(usually marked 100n or 104)*

RESISTORS – ALL ¼W 5% OR BETTER (4 BAND RESISTOR COLOUR CODES SHOWN)

2 x 1KΩ
4 x 2.2KΩ
2 x 1MΩ

SEMICONDUCTORS

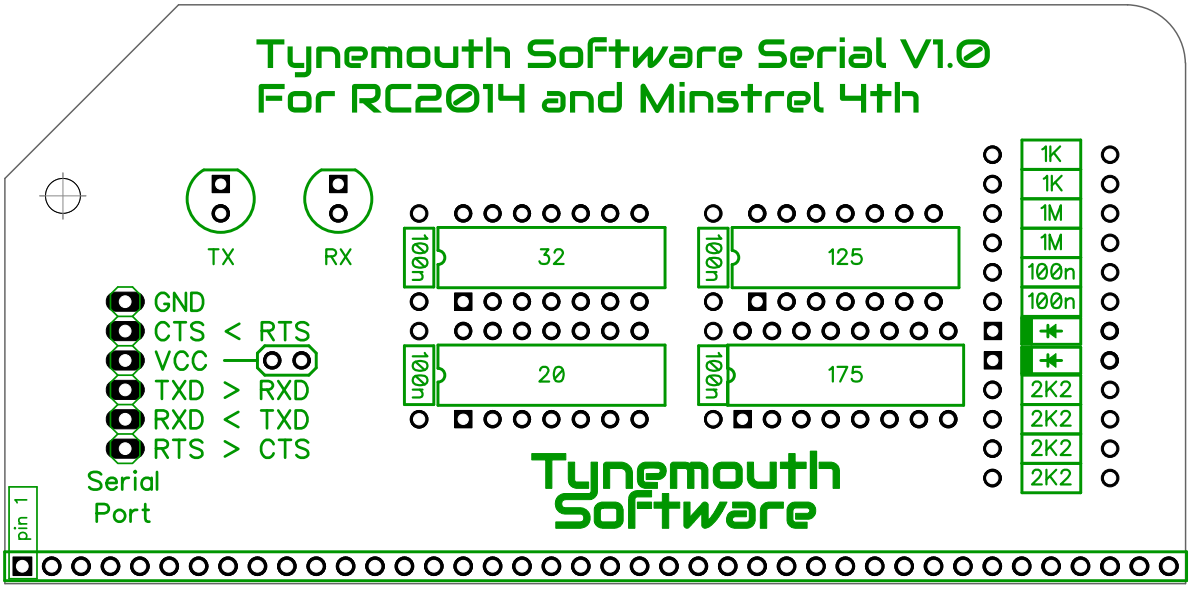
1 x 74HC20
1 x 74HC32
1 x 74HC125
1 x 74HC175
2 x 1N4148 diode
1 x Red 5mm LED
1 x Green 5mm LED
1 x Replacement Minstrel 4th ROM with support for software serial (optional or write your own code)

CONNECTORS / SWITCHES

1 x 40 way 0.1” right angled header
1 x 6 way 0.1” right angled header
1 x 0.1” 2 way header and jumper (optional)
3 x 14 way IC sockets (optional, turned pin recommended)
1 x 16 way IC socket (optional, turned pin recommended)

Tynemouth Software

COMPONENT PLACEMENT



ASSEMBLY

Start with the resistors and capacitors, then the ICs, LEDs and connectors.

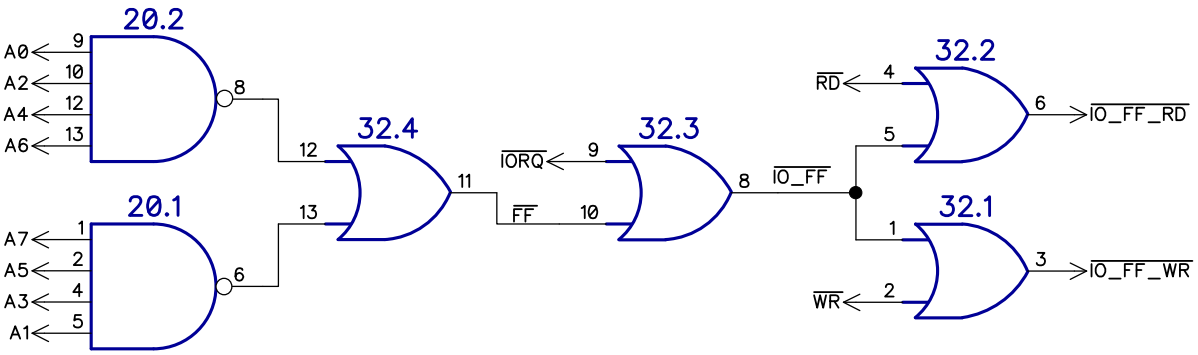
IC sockets are optional, they should not be required for simple logic ICs like these, but you can fit them if you prefer.

The LEDs should be installed matching the footprint on the PCB. The cathode (negative) lead is the shorter one, and corresponds to the flat on the case.

SCHEMATIC

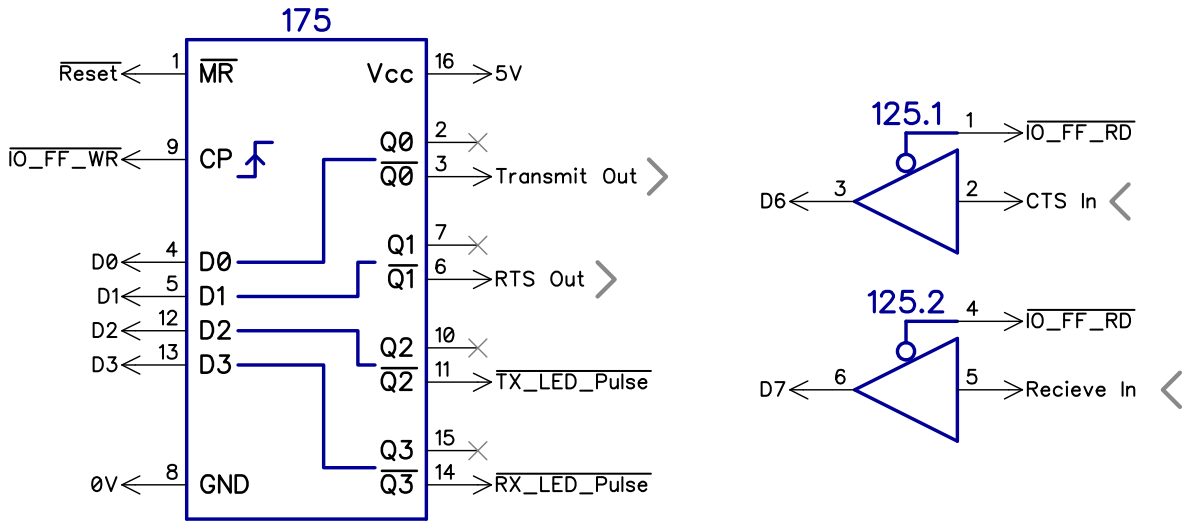
ADDRESS DECODING

The IO port responds to address $FF, which is decoded using the following logic to generate IO_FE_RD and IO_FE_WR signals for IO reads and writes respectively to the address $FF.



IO PORT

The IO port is implemented with a 74HC175 and half of a 74HC125, triggered using the read and write signals generated above.



The IO port is arranged as follows:

Port $FF	Read	Write
D0	-	Transmit Data Out (inverted)
D1	-	RTS Out (inverted)
D2	-	TX LED (active low)
D3	-	RX LED (active low)
D4	-	-
D5	-	-
D6	CTS In	-
D7	Receive Data In	-

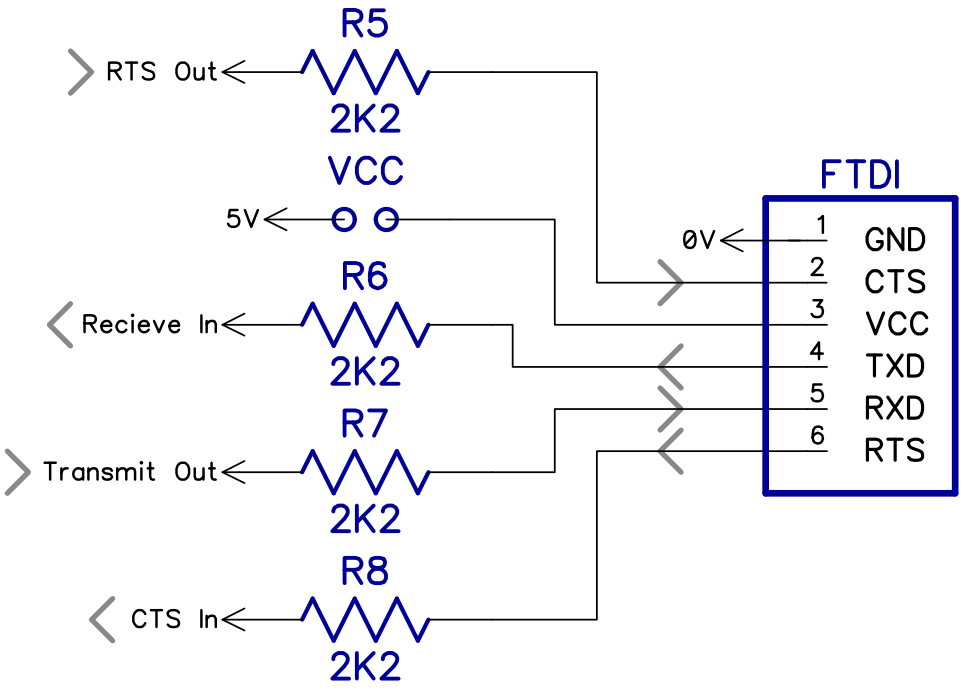
Bits 0 and 7 have been chosen for the data signals to allow use of rotate instructions to minimise the number of cycles required.

The two output signals are inverted, writing a 0 will set the RS232 line high (inactive, logic 1) and writing a 1 will set the line low (active, logic 0). These bits are set to 0 (inactive) at reset.

The input signals are no inverted and represent the line level (0 = logic 0, 1 = logic 1).

SERIAL PORT

Communication with the outside world is via a 6 pin header suitable for a 5V FTDI USB serial cable.



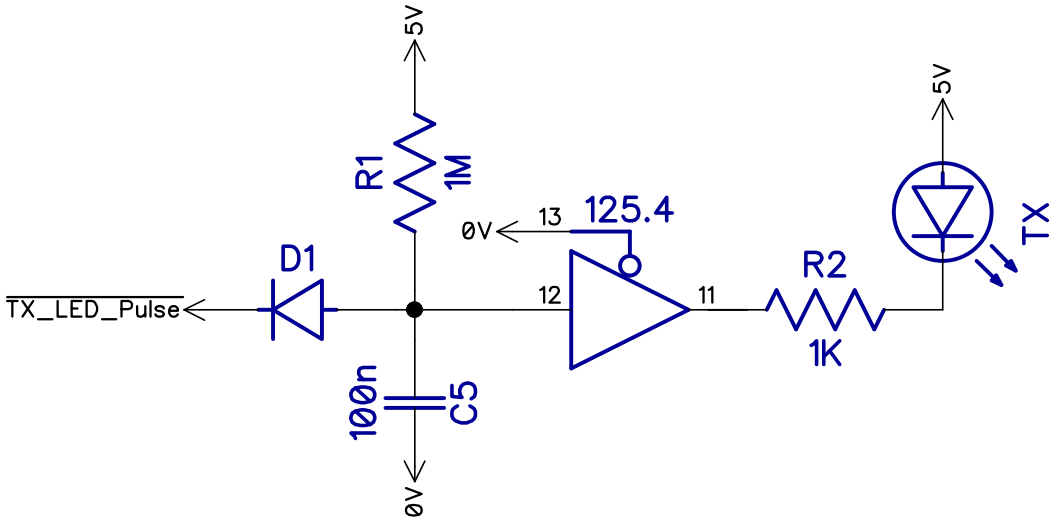
The connections are as follows:

Pin	PC ←→ 4th	Function
1	GND	0V (usually the black wire)
2	CTS ← RTS	The RTS output, low to indicate the board is ready to receive a character
3	VCC	5V supply from the PC.
4	TXD → RXD	This is the serial data input to the board from the PC
5	RXD ← TXD	The is the serial data output from the board to the PC
6	RTS → CTS	The CTS input, low to indicate the PC side is ready to receive a character

If the 5V jumper is fitted, this can be used to power the Minstrel 4th via the USB cable. This is an alternative to the normal barrel jack 9V DC in, and should not be fitted at the same time. Be wary of current limits of the USB power source.

LEDS

In order to make the LEDs flash long enough to be visible even on single characters, buffered pulse extenders are made using the two spare 74HC125 gates (the RX LED circuit is identical).



Writing a 1 to bit 2 of port $FF causes the TX_LED_Pulse signal to go low. This should be held low for at least 100µs, one character width works well, and then returned to high.

This will discharge the 100nF capacitor C5 via D1 so the input to the buffer goes low and the LED turns on.

Approximately 50ms later, C5 will have charged up enough via the 1MΩ resistor R1 to count as a logic 1, so the LED will go off again. This is retriggerable as every transmitted byte will discharge C5 again, so the LED will remain on until 50ms after the last byte was transmitted.

Tynemouth Software

ENHANCED FORTH ROM

The optional replacement ROM for the Minstrel 4th includes the many additional commands, a selection of which is listed here.

TX – Send a byte over the serial connection (non-blocking, CTS is not currently checked)

RX – Try to receive a byte over the serial connection, will return -1 if no byte available (non-blocking)

RXB – Wait to receive a byte over serial, will block until a byte is received of return -1 after timeout.

TEE – Anything printed to the screen will be echoed over the serial connection

UNTEE – Stop echoing the screen

TTY – Convert characters sent over the serial port as keypress to allow remote control, this also activates TEE. Press SHIFT + BREAK on the Minstrel 4th keyboard to exit

TAPIN *filename* – Open a tap file image on the accompanying serial server running on the PC end (see github)

LOAD *(name)* – Load a file from the tape image, the name is optional or the dictionary name will be used

TAPOUT *filename* – Open a tap file for writing

SAVE *name* – Save to a previously opened tape image

LS – Display directory of the file server folder

LOAT *name* – The original tape load routine

SAVT *name* – The original tape save routine

More commands and functionality is being added, updates will be available from the github

- https://github.com/tynemouthsoftware

Look out for announcements on

- http://blog.tynemouthsoftware.co.uk