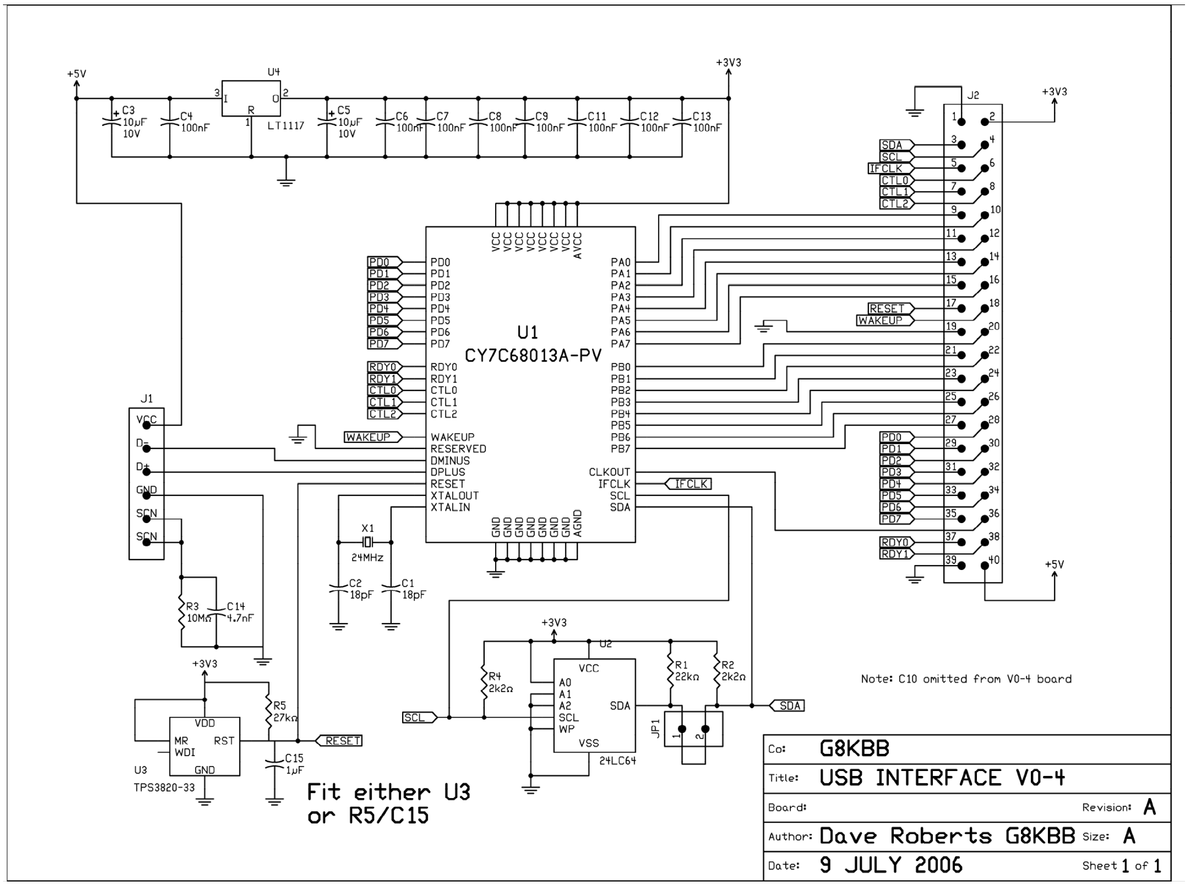
Controller hardware

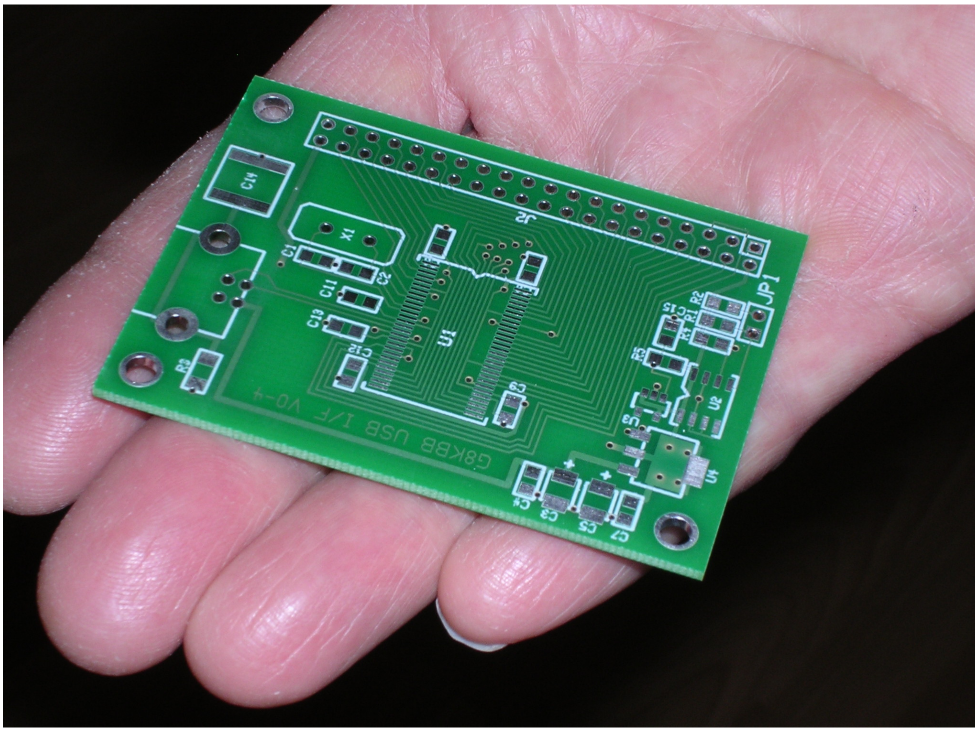
The controller hardware I am using is a simple CY68013A USB controller board as is used for the N2PK VNA

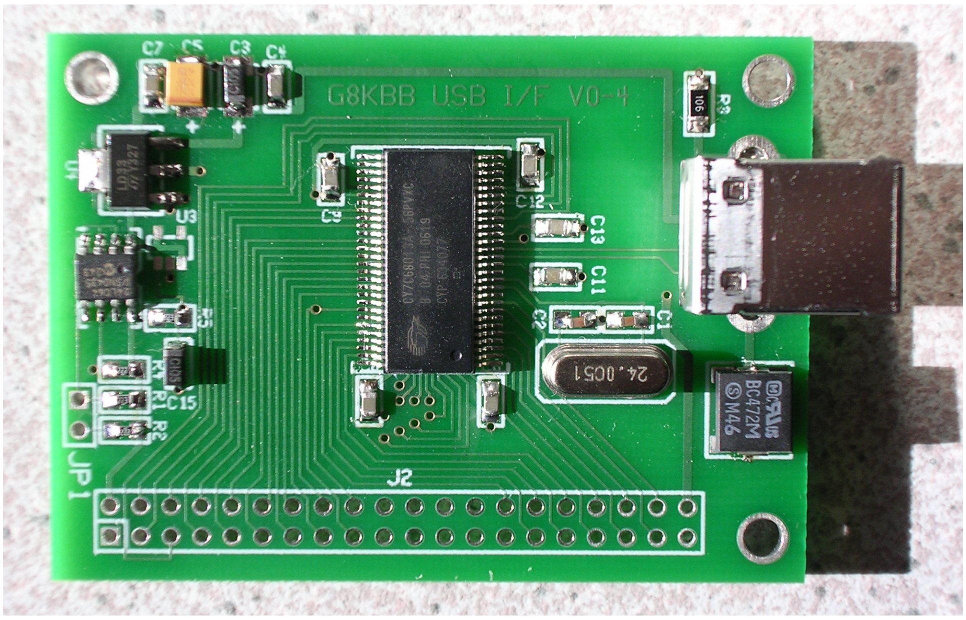
The schematic is thus



You can either use one of these modules (data is freely available on http://g8kbb.roberts-family-home.co.uk) or use the WB6DHW module, the elrasoft or any one of a number of similar boards. At the moment we only need 3 ports so the 56 pin version is fine, or maybe someone will do a SLIM version (but I would caution against it until it is all finished just in case). In any event, if you have the choice and the price is not silly, it might be better to get one of the larger ones (100 or 133 pin). For driving the parallel port as opposed to a replacement for the SLIM control board, 56 pins is fine.

Here is a piccy of a typical board:




G8KBB USB I/F V0-4
J2
JP1
24.0C51

There are two ways to connect the hardware

1. In place of the SLIM control board
2. Driving the SLIM control board

The first is designed for new builds where you can avoid the need for the control board, the second for people with a current parallel control board.

1. In place of the SLIM control board

The signals from the controller board connect directly to the SLIM modules such as the PLL, DDS and ADC modules. Therefore the 40 pin header shown above connects to the hardware. The current pinout is modelled on the SLIM controller thus – but this is SUBJECT TO CHANGE !!!

```
//------------------------------------------------------------------------------
// bit definitions for IO ports
//------------------------------------------------------------------------------
//
// PortA ADC interface
//
#define bmAdcConv      bmBIT7
#define bmAdcSerClk    bmBIT6
#define bmMagData      bmBIT5        // Input bit
#define bmPhaseData    bmBIT4        // Input bit
#define bmSpareA3      bmBIT3
#define bmSpareA2      bmBIT2
#define bmSpareA1      bmBIT1
#define bmSpareA0      bmBIT0

// PortB control port 1
//
#define bmRbwfilA2     bmBIT7
#define bmRbwfilA1     bmBIT6
#define bmRbwfilA0     bmBIT5
#define bmDataDds3Pll2 bmBIT4
#define bmDataPll3     bmBIT3
#define bmDataDds1     bmBIT2
#define bmDataPll1     bmBIT1
#define bmDataClk      bmBIT0

// PortD second VNA control port
//
#define bmSpareD7      bmBIT7
#define bmInvertPDM    bmBIT6
#define bmSpareD5      bmBIT5
#define bmLePll2       bmBIT4
#define bmFqUdDds3     bmBIT3
#define bmLePll3       bmBIT2
#define bmFqUdDds1     bmBIT1
#define bmLePll1       bmBIT0
```

I would suggest anyone building a controller, whilst they can use the 56 pin version as shown above, buys the 100 or 133 pin version if the price is similar for the extra flexibility it gives.

2. Parallel port driving version

In this version the board drives the current parallel port. The connections are as follows

Parallel port pin on MSA		USB connection on controller	
db-25 pin	Name	Signal	Comments
1	STROBE	PD3	Latches data into P4
2	D0	PB0	Data bus for latches Bit 0
3	D1	PB1	Data bus for latches Bit 1
4	D2	PB2	Data bus for latches Bit 2
5	D3	PB3	Data bus for latches Bit 3
6	D4	PB4	Data bus for latches Bit 4
7	D5	PB5	Data bus for latches Bit 5
8	D6	PB6	Data bus for latches Bit 6
9	D7	PB7	Data bus for latches Bit 7
10	ACK	PA4	Phase data from slim ADC
11	WAIT	PA5	Magnitude data from slim ADC
12	PE	PA3	Spare input line
13	SELECT	PA2	Spare input line
14	AUTO	PD2	Latches data into P3
15	Error		Unused pin
16	INIT	PD1	Latches data into P2
17	SELT	PD0	Latches data into P1
18-25	ground	GND	Signal ground

USB port D bits 4-7 are spare
USB port A bits 0,1,6,7 are spare
USB RDY / CTL pins are unused

Connect the pins directly to the 25 pin connector on the MSA via short leads – the shorter the better as these signals are fast logic and not high power bus drivers. Avoid filtered D connectors.