DIY Constructors Manual

*The **QLAN to USB Bridge** (QLUB) **Adapter** is an open-source solution to allow the connection and communication between real Sinclair QL hardware and a QL-Emulator running on a host PC using Sinclair's QLAN/QL-Net protocol and cabling, with support for the TK2 QLAN extensions, including FSERVE.*

Several options exist today to allow transfer of programs and data between emulated and QL hardware, including SERNET and removeable SD-card media, however these require either additional hardware or software at the QL end and thus limit their accessibility for 'unexpanded QL' users. QL-Net on the other hand is built-in to all QLs and is thus compatible with even a minimal QL configuration.

The QLUB Adapter is composed of both hardware and software components – the hardware based on a microcontroller development board (Teensy ++2.0) with custom firmware and some additional components, as well as software written to run within a QL emulator on the host PC, interfacing between QDOS/SMSQE client applications or device-drivers and the microcontroller which is connected to the host PC via an available USB 2.0 port.

The adapter requires a few additional components, readily available and simple enough for most enthusiasts to assemble on a breadboard with minimal soldering required - instructions and a schematic appear later.

A SBASIC program has also been devised to test the basic sending and receiving of files between the QL emulator and a peer QL (or compatible) and is already effective for transferring files between these platforms.

This 'proof of concept' software is being made available to the QL Community at this pre-release stage (December 2020), along with the latest microcontroller firmware and these draft instructions so as to allow user testing and feedback whilst work continues on the software release-candidate that will ultimately be fully integrated in to QDOS/SMSQE running in the emulator.

The QL emulator QPC2 has been used throughout the development cycle, but the final solution should be compatible with any QL emulator that exposes the host PC's COM/serial ports and a platform for which a suitable USB Virtual COM port driver exists for the Atmel microcontroller.

As the ZX Spectrum with the Interface-1 also provides compatible network hardware and software, the QLUB will work equally well connected to a Spectrum (or indeed, a Spectrum Next with Int-1). The same applies to any QL-like solution that includes the QLAN NET port hardware such as the Q68, QXL and Aurora, among others.

As of December 2020, the project remains a work in progress, with the microcontroller firmware tested successfully, most of the software API interface and data-structure design complete and the realisation of a simple 'message-queue' server Job well under-way in assembler.

MARTYN HILL
29 Dec 2020

DIY Constructors Manual

Goals and Motivation

The original goal of this project was to facilitate development of QL software on a PC within a QL emulator - to then be transferred to a target QL without recourse to swapping removeable SD media back and forth which, at the time of inception at least, was particularly cumbersome.

As well as removable media, SERNET was also tried but proved unsuccessful on the author's hardware and was not pursued further; in any case, I was not enamored with the bulky serial-cables needed to connect hosts, versus the simplicity of the original QL-Net cabling.

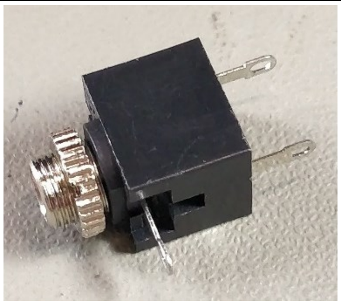
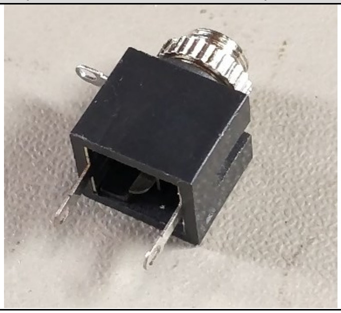
Another significant motivation was a fascination with networking in general and the Sinclair Networking capability in particular. In fact, it is this fascination with Sinclair's design that has been the *real* motivation to continue with the project over the lengthy time since its conception.

It has always been the intention to make this solution accessible and available to all QL users under an open-source arrangement. I would be equally happy to work with any of the hardware developers in the community to bring a ready-made solution to market - as long as the source code and designs remains open to all.

Building the QLUB Adapter

To build and program the adapter you will need:

1. The **Teensy ++2.0 development board**, ideally with header-pins already fitted (c £20, local distributors are listed on the PJRC website: https://www.pjrc.com/store/teensypp.html)
2. A **mini-USB to USB-A cable** to connect the adapter to the host PC (during both programming **and** 'in action.')
3. The **Arduino IDE** (free to download: https://www.arduino.cc/en/software)
4. The **Teensy add-on libraries** (automatically downloadable from within the IDE)
5. The latest **microcontroller firmware** - currently, **QLAN-USB_v21e.ino** (available from the author via the Sinclair QL Forum: http://www.qlforum.co.uk/)
6. A **breadboard** with at least 30 rows of contacts (generally available)
7. Various **electronics components** and some wiring/soldering equipment (all generally available):
    - 2x 330ohm, 1x 47ohm, 1x 1Kohm, 1x 3K9ohm, 1x 10Kohm resistors
    - 1x PNP bi-polar high-speed switching transistor, ZTX-510 or equivalent
    - 2x mono-switched chassis Jack sockets (e.g. RS Stock# 106-874)

Switched chassis Jack socket example *(note: **3** connections)*	
	

NB – Stereo Jack sockets also have 3 connections, ***but are not equivalent, nor suitable*** for this project. Make sure you purchase the **mono-switched-type** explicitly.

8. Ideally, a suitable **project-box** (c 110mm x 60mm, 20-25mm deep)

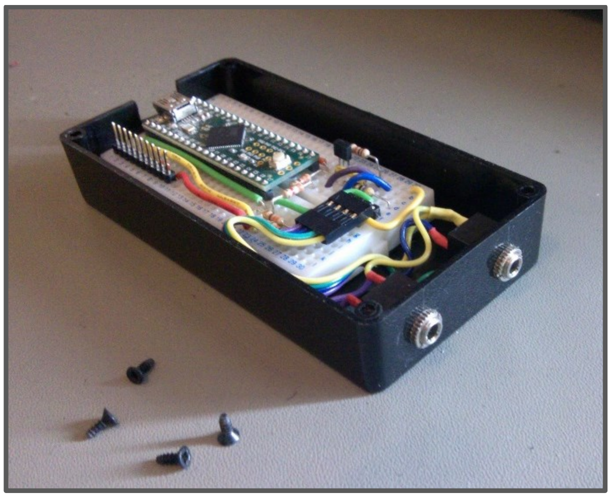


Figure 1: QLUB Adapter installed in project box

9. Don't forget one or more **mono (or stereo) cables**, terminated with 3.5mm TS/TRS Jack plugs at each end to connect the QLUB Adapter to your QL/Spectrum.

NET Port Connection Plugs	
3.5mm 'TS' Jack-plug (mono)	**3.5mm 'TRS' Jack-plug (stereo)**

The schematic for the additional hardware components will be familiar to anyone who has studied the QL or Spectrum Interface-1 Technical Service Manuals, as it is a like-for-like match, thus:

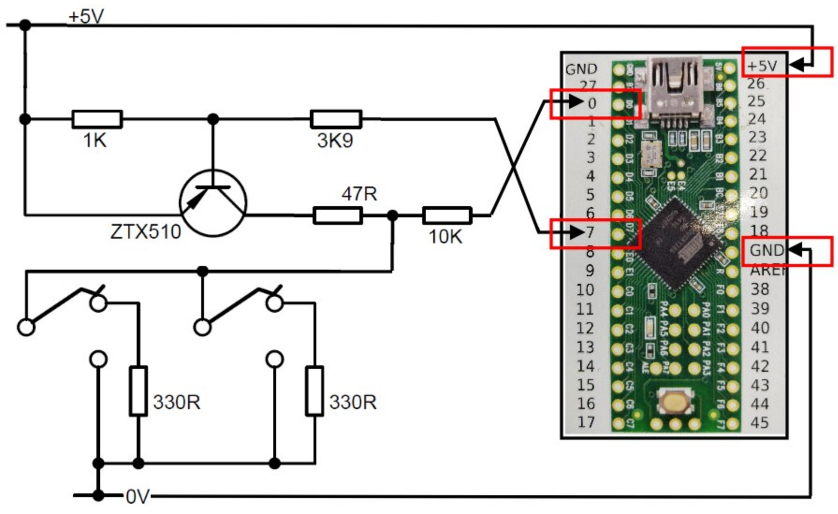
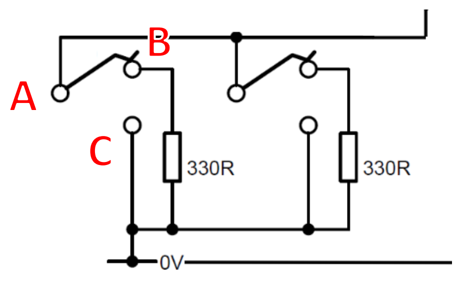
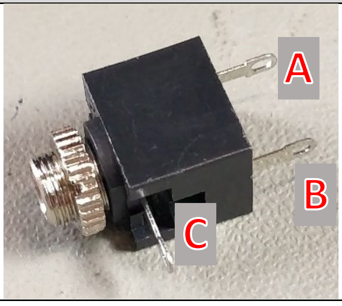


Figure 2: QLUB Adapter Wiring Schematic
*(copied in part from QL Schematic by **D Westbury** – see http://www.dilwyn.me.uk/docs/hardware/QLissue5.pdf)*

More on the Jack sockets…

Knowing how to connect the three solder-tags of the switched-Jack socket can prove perplexing – the following diagram should help, but bear in mind that there are slightly different designs of switched sockets on the market, so take care to validate your work against the model of socket you purchase…

Switched Chassis Jack Socket – Identifying the connections	
	

Connection points **A and C** are common to both sockets in a two-socket design – whereas connection **B** on each socket attaches to its own pull-down/termination resistor.

Note 1*: It is perfectly possible to build* ***only one socket*** *in your design as long as the QLUB Adapter will always act as the terminating device* ***at either end*** *of your network. In this case you can use a single* ***non-switched*** *type socket (only 2 connections) and will need to connect the pull-down/termination resistor* ***in parallel across*** *the two connection points. The termination is still required, but at only half the resistance value (i.e. 165ohm.) Simply solder 2x 330ohm resistors in parallel across the socket connections to achieve the desired termination.*

Note 2: *Even more simply, if the QLUB Adapter is going to sit* ***between*** *two other stations on the network, then you can use 2x* ***non-switched*** *type Jack sockets and* ***dispense*** *with the termination resistors altogether – the two other end stations then take-on the role of terminating the network.*

Programming the Teensy Dev Board

There is ample description and instruction on use of the Arduino IDE for programming Atmel microcontroller boards like the Teensy, plus additional information on the PJRC website specific to the Teensy range of development boards to get you going.

Please refer to these resources before posting questions to the author via the Sinclair QL Forum.

Once programmed and after every subsequent restart/reconnection to the host PC, the QLUB adapter will automatically start listening for suitably formatted commands from the host PC via the USB port. Thus, the USB connection remains an active part of the QLUB design and **not** just used during programming.

When new firmware is released, it is simply a matter of downloading and opening the latest `.ino` source code file back in to the Arduino IDE and re-flashing the device.

To flash successfully, **no other application** can be holding the virtual COM Port open, so you must close the QL Emulator to release the SERial ports when ready to re-flash the microcontroller.

File-Transfer Software

The QDOS software *currently* available (**SendFileMQv14_bas**) is written in SBASIC and allows for simple file-transfer back/forth between emulator and the QL or Spectrum. Like the microcontroller firmware, you can request this from the author via the Sinclair QL Forum.

The final QDOS software will support the full TK2 NET extensions, including FSERVE and will be fully integrated into SMSQE as a revised NET driver and MQ Server task.

Once you have validated which SERial port the USB Virtual COM port appears-as within the emulator when plugged-in, simply edit the variable `virtualCOMPort%` near the top of the listing to match and then RUN the program.

See the initial `REMarks` in the listing for ideas on what to enter at the QL/Spectrum end of the link.

QDOS and ZX File headers are both managed correctly and automatically by the provided transfer routines.

For a full explanation of the use and workings of the QL LAN in general, please refer to the **Networking the QL** article, also by this author and available from the Sinclair QL Forum.

Considerations/Caveats

- The initial installation of the microcontroller development board itself can prove problematic and sometime takes a few attempts to install the Virtual COM Port driver (in Windows) before you can successfully program the device – just persevere.
- The dev board can also appear to hang from time to time when running the provided QLUB firmware and may occasionally need to be unplugged/re-connected to get correctly re-enumerated by the host OS USB stack before the QL emulator can access the SERial/COM port successfully.
- Use of the QL network has seen mixed success in practice, due to a number of software-timing and hardware issues over time. The QLUB Adapter can't fix those directly, but otherwise follows the Sinclair defined timing even more closely than native QL hardware. Modified timing-constants in the TK2 code have been seen to address most issues and details can be provided on request.
- Likewise, interconnecting QLs and Spectrums is often problematic and the Shadow ROM code in the Interface-1 has a couple of previously undiscovered bugs that make use of the network unreliable even with the QLUB Adapter. The author can make modified ROM software available for the Interface-1 that overcomes these issues.
- Otherwise, most issues you are likely to face using the QLUB Adapter are common to native QL/Spectrum networking and lots of advice is provided in the **Networking the QL** article previously mentioned. *Have you read it yet?*