

Q L T e r m

A Small and Simple Terminal Emulator for the QL

Program copyright (C) 1987, 88, 92, 93, 94, 2016 by Jan Bredenbeek

English documentation copyright (C) 1993, 94, 2016, 2019 by Jan Bredenbeek

LICENCE INFORMATION

As of 2019, QLTerm is distributed under the GNU Public Licence v3.

The released versions, source code and terms of licence can be found at:

<https://github.com/janbredenbeek/QLTerm>

Feedback to the author can also be given at this site, or via e-mail: jan@bredenbeek.net

CONTENTS

Introduction	page 3
Getting Started	page 4
Using QLTerm	page 7
Customising QLTerm	page 10
QLTerm Reference	page 11
Appendix 1: Keyboard and Display Control Codes	page 18
Appendix 2: XMODEM Protocol Overview	page 22
Appendix 3: QLTerm Revision History	page 23

INTRODUCTION

QLTerm is a small and simple program that enables you to use the Sinclair QL, or a QL-compatible computer, as a terminal for communication with other computer systems (e.g. a Bulletin Board System), using the RS232 serial interface and a modem.

QLTerm supports the plain ASCII, VT52 and ANSI terminal standard, including the PC extended character set and colour commands.

In addition, QLTerm provides file-transfer (up- and download) using the XMODEM protocol (with and without CRC error detection).

QLTerm can be used with the Modaptor and QL Modem by Miracle Systems or modern intelligent (e.g. Hayes-compatible) modems that connect directly with the QL's serial port. QLTerm supports four methods for generating 1200/75 bps split-speed, using the QL Modem, Modaptor(+), Hermes or direct serial I/O.

QLTerm is designed to be fast and compact and is written entirely in 68000 assembly language. It was (and is) **not** the goal of QLTerm to provide the most sophisticated all-singing-and-dancing communication program ever designed, so keep this in mind if you find QLTerm rather limited. There are many terminal emulator programs available for the QL nowadays; most of them are very complex and require a lot of knowledge (and time!) to install. In contrast, QLTerm is simple to install, provides easy access to the elementary functions and has extremely good display performance, even at 19200 bps.

Finally, with the advent of QL emulators which support TCP/IP access through the TCP device, such as QPC, Qemulator, SMSQmulator and uqlx, QLTerm can be used as a Telnet client to access systems using the Telnet protocol. This has not yet been fully implemented, but QLTerm has been tested successfully against a number of Telnet BBS systems. Please see the included TELNET_BAS file which can be used to establish a Telnet session from S*BASIC.

GETTING STARTED

Installation

Depending whether you downloaded QLTerm downloaded from a Bulletin Board or got it on microdrive or disk, you should either unpack the QLTRMnnn_ZIP archive file or copy the QLTERM, QLTERM_DOC and QLTCNFIG_BAS files to a clean cartridge, disk or directory. Then enter, from within SuperBASIC, the command:

EXEC_W <device>QLTerm

(For <device> you must of course substitute mdv1_, flp1_ etc. as appropriate).

QLTerm is now ready for operation, but you probably will need to change some things to suit your needs. Press the F3 key; you will now see a menu of available options at the bottom line of the screen.

Firstly, select the Interface command by pressing the **I** key. You will then get the following prompt:

(Note: An underlined option means that the option is highlighted on the display; this may be different from what is shown here if you have a customised copy of QLTerm).

Transmit mode: Echo Off Echo On Local Only

For our purpose, the **Echo Off** option should be selected. Use the left- and right-arrow keys to highlight this option, then press ENTER.

Next, you will get:

Modem type: Qmodem Modaptor Buffered

You need to make a choice here depending on your modem equipment.

If you have a Miracle Systems QL Modem, select **Qmodem**.

If you have another type of modem connected via the Miracle Systems Modaptor interface, select **Modaptor**.

If you have neither a QL Modem nor a Modaptor connected, select the **Buffered** option.

Port: SER1I SER1H SER2I SER2H

This selects the port to be used for modem communications. In most cases, you need to select **SER2I** here. If you have your modem connected to the SER1 port rather than SER2 (Note: this requires a different wiring between QL and modem!), you should select one of the **SER1** options.

If you are using a high-speed modem (capable of line speeds greater than 2400 bps) or a modem which uses a fixed speed between computer and modem (e.g. Astracom 1000), you need to use handshaking to prevent loss of outgoing characters. In that case, select the **SER1H** or **SER2H** option depending on whether your modem is connected to SER1 or SER2.

Baud: 75 300 600 1200 2400 4800 9600 19200 12M75 12P75 12S75 12H75

Here you need to select the baudrate to be used between QL and modem.

If you have selected **Qmodem** earlier (Miracle Systems QL Modem), you should choose the **12P75** option here.

If you have connected a "dumb" modem (without an internal buffer or interspeeder) through the Modaptor(+) and wish to make a call at 1200/75 bps, you should choose either the **12M75** option (for the "old-style" Modaptor with two switches on it) or the **12P75** option (for the "new-style" Modaptor+). For calls at any other rate using the Modaptor, select the appropriate rate. If you have an "old-style" Modaptor with two switches, you should also set them appropriately according to the baud rate.

In other cases, the rate you should select depends on the type of modem used. Most modems which have as highest speed 2400 bps (both "intelligent" and "dumb") should have the baudrate set equal to the speed of the connection to be made. For instance, if you have a 2400 bps modem and you are going to make a call to a Bulletin Board which you know only supports 1200 bps connects, you should set the baudrate to 1200. If you know the BBS supports 2400 bps, you can set the baudrate to 2400.

If you have your 1200/75 bps modem connected directly to the QL's serial port, you should use the **12S75** option. If you have fitted the **Hermes** IPC chip (which is desirable at 1200/75!), select the **12H75** option which makes use of Hermes's split-baudrate facility.

If you have a modem which supports higher line speeds than 2400 bps (or uses internal error correction such as MNP or V.42), you can often use a fixed baud rate between computer and modem (which should be as high as possible to obtain maximum performance at high line speeds). The modem will then automatically convert the data rate on the line to that on the serial link in both directions. With these modems, you should select the highest speed possible for computer-to-modem communications (refer to the modem's manual). Usually, this will be 9600 bps (e.g. Astracom 1000 or 2400 bps modems using error correction) or 19200 bps (modems supporting V.32 or higher data rates).

Notes:

1. When using a high-speed modem, or a modem equipped with error correction, where the modem-to-computer baud rate may be greater than the data rate on the line, be sure to select the Handshaked option when choosing the serial port (SER1H or SER2H). Also, the CTS (Clear To Send) line from the modem must be properly wired to either the DTR line (on SER1) or the CTS line (on SER2). Failure to do this might result in loss of transmitted data!
2. Although QLTerm allows you to set the baud rate to 19200, this does not mean that this will always work. In particular, the QL's serial ports originally cannot receive data at 19200 bps (although they can send data at this rate). Data reception at 19200 is possible though if you use **Hermes**, a replacement chip for the 8049 second processor in the QL.

Parity: None **Odd Even Mark Space**

You should select **None** here in the vast majority of cases.

Next, press the F3 key again and select **O** for Options. You will then see:

Terminal Emulation: ASCII VT52 ANSI

The option you should select here depends on what type of terminal QLTerm should emulate. **ASCII** is the simplest type, offering only basic cursor control. Select this type if you do not need advanced terminal emulation, e.g. when calling a BBS that doesn't support ANSI (e.g. QBOX). **VT52** offers you more advanced cursor control, including colour. It is rarely used in amateur Bulletin Board Systems, but you may find it useful when communicating with mainframes etc. **ANSI** is a most popular protocol in the world of Bulletin Boards. It offers advanced cursor- and colour control. Select this if you are going to call a PC-based Bulletin Board.

Colour: NO 4-Colour 8-Colour

You can turn colour display on or off, independent of the terminal type setting. Turning colour off may be useful if you have a monochrome monitor. When using colour, you can choose between 4-colour (when using native QL 512x256x4 resolution) or 8-colour (when using high-resolution graphics modes, e.g. on QPC2 or the Q68). In 4-colour mode, QLTerm will convert blue to red to prevent blue text from being displayed in black and hence invisible on a 4-colour screen. Other colours will still be displayed as usual so don't be surprised if you still see more than four colours on your high-resolution screen!

In the configuration program QLTCNFIG_BAS, you can also choose option AUTO for this setting. When set, QLTerm will automatically determine the appropriate colour mode at startup, depending on your system's capabilities.

With colour display turned off, you still have the advantage of ANSI cursor control (e.g. animation, full-screen online editors) while retaining maximum legibility.

Logging: RAW COOKED

This option controls whether or not VT52- or ANSI control sequences are being logged when using the log-to-file option. It is described in more detail in the next section, under "Logging to a file". For the moment, you don't have to worry about it.

Convert CR/LF for LOG and TRANSMIT: NO YES

If you are intending to use QLTerm's log-to-file option, or transmit textfiles using the Transmit command, you will probably want to strip the Carriage Return characters at the end of each line from the incoming data and convert the line feed characters in QDOS textfiles to CR when transmitting the file. If you select **YES**, QLTerm will do so.

USING QLTERM

Dialling a Number

Now that we have the Interface and Options settings set up properly, we can start dialling an online service such as a Bulletin Board System.

The dialling procedure will be different depending on the modem you have.

- * If you have a Miracle Systems QL Modem, use the Call command (F3 followed by C) within QLTerm. Enter the number as you would dial it normally, e.g. 012-345678. Note that any non-numeric character, such as a dash sign, will cause QLTerm to pause two seconds before dialling the next digit. This may be useful when dialling from a PABX exchange for instance.
- * If you don't have a QL Modem, you should dial either using the modem's own command or by means of phone set connected to the same line, if your modem cannot dial a number by itself. For instance, when you have a Hayes-compatible modem, you can dial a number by issuing the command **ATDP** or **ATDT**, followed by the phone number (and terminated by an ENTER keypress). Note that ATDP uses pulse-dialling, while ATDT uses tone-dialling. You should check which method is supported on your exchange (pulse-dialling is supported on all public exchanges but not on some PABX exchanges, while tone dialling is only supported on newer public exchanges).

When the dialling process has been completed, there are obviously two possibilities. The first possibility is that all goes well and you see the login screen from the remote system after a few seconds. But if you don't get anything, or just garbage on the screen, something is obviously wrong.

The easiest case is of course an engaged phone line, or the other end simply doesn't answer your call. When using an intelligent Hayes-compatible modem, this can be easily verified as the modem will report "BUSY" when it detects an engaged line. When dialling manually using a phone, this shouldn't also be hard to recognise.

If your modem doesn't seem to recognise anything, and it doesn't have a provision to monitor the line via a built-in speaker, it may be useful to check the number by dialling it using a phone set and listening to the line. It might be that the line is just busy but your modem doesn't recognise the signal (the Miracle Systems QL Modem doesn't recognise any phone signals at all), or perhaps you have dialled the wrong number and actually got someone on voice!

If you can hear the other end's data carrier, it could be that your modem is set to the wrong speed, or simply isn't compatible with the other end's modem at all.

If you are using the Miracle Systems QL Modem and you do not get a connection, you should disconnect the modem using the **Hangup** command (press F3 followed by H) before trying again. Do not persist in re-dialling over and over again but check the line manually using a phone!

Using QLTerm online

Once a connection has been established, your QL will act as a (rather intelligent) terminal for the remote system. That means, any output from the remote will be displayed on your QL's screen, and any characters you type will be sent to the remote (with the exception of function keys). In terminal mode, a status line is displayed at the bottom of the display showing the various parameters you have set using the **Interface** and **Options** commands.

Pressing the **F3** key will bring up a **Command Menu** at the bottom line. You can select a command by pressing the key of the first letter of it's name. Pressing **ESC** in the command menu will return you to terminal mode.
The individual commands will be discussed below and in the Reference section further on.

Logging to a file

In some cases, it may be useful to keep the data you receive on the screen in a file for later review. This also allows you to keep the connection time down to a minimum, thus saving phone costs.

The **Log** command, invoked by pressing F3 followed by L, can be used for this. When invoked, it asks for the name of the file to be used to store incoming data. You should enter the complete QDOS filename, including device name etc (e.g. ram1_bbs_log). It is recommended that you use a "fast" device such as RAMdisk for logging, as "slow" devices like microdrive or even floppydisks tend to disable interrupts at certain times, causing loss of incoming serial data.

There are two settings in the **Options** command which affect the format of the logged data:

The **RAW/COOKED** option controls whether VT52- or ANSI escape sequences are being logged (RAW) or not (COOKED). **Raw** logging is useful if you want to playback the log file later using QLTerm's **Transmit** command, as colour and animation codes within the log file are preserved. On the other hand, if you want to import the log file into a text editor, you may want to filter out the escape sequences from the actual data as they will probably be quite annoying. This can be accomplished by selecting **Cooked** logging.

Secondly, the **Convert CR/LF** option affects the end-of-line format in the log file. When set to **ON**, carriage-return characters are stripped from the log file so that lines end with only a LF character, as is the QDOS convention for text files. When set to **OFF**, CR characters are logged as normal (which is required if you want to playback the log file using QLTerm later).

When you find you have logged enough data, you can close the log file by issuing the Log command again.

File transfer

Having established a connection, you may want to transfer some files to or from the remote system. This is achieved using the **Upload** and **Download** commands.

* **Uploading** means that you are **sending** a file to the remote system

* **Downloading** means that you are **receiving** a file from the remote system.

To upload or download a file, you should first instruct the remote system to go into up- or download mode. Usually, it will ask for a protocol to be chosen. You should select the **Xmodem** protocol then.

When the remote system is waiting for you to begin the up- or download session, press F3 followed by U (for Upload) or D (for Download). QLTerm will then ask for the name of the file to be sent or received. You must enter the full QDOS filename, including the device name. When Uploading, QLTerm will also ask if you want to send the header along with the file. This is only relevant if your file is of EXECutable type; refer to the Command Reference section for further information. In most cases however, you can respond with N(o) to this question.

The transfer will now proceed. On the bottom of the screen, QLTerm will display the block number of the 128-byte block being transmitted or received, and the number of attempts having been made (when an error occurs during block transfer, the receiving end will automatically ask for a re-send up to 10 times).

Transmitting a Text file

In some cases, e.g. when writing a large message, you can save a large amount of online time (and money!) by creating the message text offline using a text editor, and then sending it all at once during the online session. The **Transmit** command (F3 followed by T) can do this for you. Transmit asks for the (full) name of the file to be sent, which must be a textfile. The file is then sent line by line, with a 1-second pause between lines to give the remote system time to "digest" the data.

When the **Convert CR/LF** option in the **Options** command menu has been set to **ON**, the LF character at the end of each line in QDOS textfiles will be replaced by a CR character which is required by most online services.

Note: Some message editors used by Bulletin Board Systems treat an empty line as end-of-message. If your file contains empty lines, the rest of the file would then probably be interpreted by the remote system as commands, which would undoubtedly greatly upset the system (and it's operator!). You should then avoid having empty lines in your file (placing a full stop on the empty line will do).

Terminating a Call

If you have finished with the online session, you should disconnect (put offline) your modem. Most modern modems will do this automatically when they sense that the remote system has terminated the connection, so instructing the other end to disconnect will often be sufficient (e.g. the Goodbye command on Bulletin Board Systems).

If you have a Miracle Systems QL Modem, your modem has to be put offline manually. This is done using QLTerm's **Hangup** command (press F3 followed by H).

Quitting QLTerm

The **Quit** command (F3 followed by Q) will terminate QLTerm and return to SuperBasic, or the program from where you started QLTerm.

If you use a Qmodem and the modem is still online, QLTerm will first ask you to disconnect it. When using other modems, be sure that you have disconnected the modem otherwise you might end up with excessive phone charges!

CUSTOMISING QLTERM

When QLTerm is started up, the parameters set by the **Interface-** and **Options** commands have default values. These default settings can be customised so that they already have the values you desire when QLTerm is started up.

To customise your QLTerm copy, LRUN the SuperBASIC program **QLTCONFIG_BAS** supplied on the QLTerm disk or distribution _ZIP file. The program will ask for the name of the QLTerm file you want to customise (e.g. **flp1_QLTERM** when your copy is on the disk in floppydrive 1).

Next, the currently installed options will be shown. These are:

- Download buffer size in kilobytes;
- Modem type;
- Port;
- Baud rate;
- Parity;
- Echo on/off or local-only;
- Terminal mode (ASCII/VT52/ANSI);
- Colour (AUTO, B&W, 4-colour, 8-colour);
- CR/LF conversion (yes/no);
- Log mode (Raw/Cooked).

Using the up and down cursor keys you can select the option to be changed, and with the left and right arrow keys you can change the selected option. When changing the Download buffer size, you have to enter a number greater than zero which will be taken as the new buffer size in kilobytes.

When you have everything set to your personal preferences, press the ESC key to exit from the menu, then press Y to write the new values to the QLTerm copy or N to abandon the configuration program without changing the old values.

QLTERM REFERENCE

This section describes the functions of QLTerm in detail. It is assumed that the reader has familiarised him/herself with the basics of data communication, terminal- and file transfer protocols in general - if not, reading the "Getting Started" section of this manual, or perhaps a Beginner's Guide to data communication, is recommended.

Starting up

QLTerm may be invoked using the SuperBasic **EXEC** or **EXEC_W** command. When using **EXEC**, QLTerm will run concurrently with SuperBasic and can be "switched away" using the CTRL-C key.

It should be noted that, due to the fact that QLTerm uses the same console window for screen output and keyboard input, QLTerm has to "poll" this window with zero timeout for input (otherwise simultaneous output wouldn't be possible). On a QL without the Pointer Environment installed, this will have the undesired effect that QLTerm tends to constantly "grab" the cursor when it is not in use by another program.

It is therefore recommended that QLTerm should only be multitasked with other programs when the Pointer Environment is present.

QLTerm may also be started up with channels and/or a command string as parameters. Either one or two channels may be specified; when one channel is specified QLTerm will use this as serial channel for both input and output, when two channels are specified QLTerm will use the first one for serial input and the second one for serial output, effectively treating them as pipe channels. The latter case may be useful when connecting QLTerm internally to another program or the TCP device to access systems over TCP/IP (see also the included **TELNET_BAS** program which can be used to initiate a Telnet session).

When a command string is present and serial I/O is not redirected through channels as described above, QLTerm will take the command string as a device name and attempt to open a serial channel to the device as specified. This allows QLTerm to be used with devices other than the standard SER1 and SER2 ports.

Terminal Mode

When QLTerm is started up, it enters Terminal Mode. In this mode, incoming characters are displayed on the screen and keys pressed on the keyboard will cause one or more characters to be transmitted (except in Local Only mode - see the **Interface** command further on).

When in Terminal Mode, QLTerm will display a **status line** at the bottom of its display. The items shown on the status line are, from left to right:

- * Echo off/on or Local Only mode. This controls echoing of typed characters - see the **Interface** command;
- * ASCII, VT52 or ANSI terminal emulation mode. Set by the **Options** command;
- * Colour or Black & White mode. Set by the **Options** command;

- * The serial port used. Usually SER1 or SER2 with I or H parameter, but may be specified on the command string when starting QLTerm. If QLTerm's in- and output channel has been redirected, this field will show the string "ReDir".
- * The baud rate used. Either one of the standard rates 75, 300, 600, 1200, 2400, 4800, 9600 and 19200 bps or "split-rates" 12M75, 12P75, 12S75 and 12H75. The latter are used for sending at 75 bps while receiving at 1200 bps, as required by the CCITT V.23 protocol. The QL's serial ports normally can't handle this split-speed and it is generated by either special hardware or a software-trick - see the **Interface** command for a complete description.
- * The parity, either None, Odd, Even, Mark or Space. Note that, when using a parity other than None, QLTerm will strip the topmost bit of incoming bytes. Parity errors in received data will not be detected.
- * The modem type, one of "Qmodem" (for Miracle Systems QL Modem), "Modaptor" (when using the Miracle Systems Modaptor or Modaptor+) or "Buffered" (for other type modems). With the "Qmodem" or "Modaptor" types, 1200/75 bps split-speed is achieved by setting the baud rate to either "12P75" for Qmodem and Modaptor+, or "12M75" for Modaptor. Also, Qmodem and Modaptor+ allow for auto-dial using the Call command.

Function Keys in Terminal Mode

The function keys F3 to F5 have a special meaning, other function keys will be ignored by QLTerm. The function keys recognised are:

F3: Command Mode

The F3 key enters Command Mode, and will display a menu of available commands on the bottom line. To select a command, press the first letter of the command name. To return to terminal mode, press the ESC key.

F4: Transmit XOFF code

Pressing this key causes a XOFF character to be sent (CHR\$(19), CTRL-S). On most online services, this causes the display to be paused until a XON (CHR\$(17), CTRL-Q) or another character is received.

Note that QLTerm automatically sends a XOFF character when its receive buffer (8K bytes) has been filled by about 75%, and sends a XON character when the buffer has been emptied so that less than 32 characters are left. This avoids loss of data when using a fast modem on a slow machine.

F5: Transmit ETX (CTRL-C) code

Pressing F5 causes an ETX character (CHR\$(3), CTRL-C) to be sent. As CTRL-C is normally used on the QL to switch between jobs, this has been provided to still allow a CTRL-C to be sent. It may be used, for instance, to cancel transmission of text on BB systems.

Command Reference

(B)uffer

This allows the size of the buffer used in Xmodem file transfers to be set.

When Downloading, QLTerm will buffer incoming data blocks to improve efficiency with "slow" media like Microdrives and floppydisks. When the buffer is full, QLTerm will write out the received data, continuing with the transfer when all data has been written.

The default buffer size is 32K bytes, which will be sufficient in most cases. It may be changed either by this command, or by customising QLTerm (see relevant section). A smaller buffer size has the advantage of using little memory, but degrades file transfer efficiency. A larger buffer is also not recommended because of large memory requirements and the risk of the remote "timing out" while QLTerm is busy emptying the buffer.

When there is insufficient free memory for the buffer to be allocated, the Download command will be aborted.

(C)all

This command causes the Miracle Systems QL Modem to dial the specified number. The command is also suitable when using the Modaptor+, and the modem has the ability for the line relays to be controlled via the DTR line. In other cases (e.g. Hayes-compatible modems), the Call command cannot be used.

The number given may contain digits (0 to 9) and non-numeric characters (e.g. '-'). Any non-numeric character causes a pause of 2 seconds to be inserted before dialling the next digit.

The Call command may also be used to put the Miracle Systems QL Modem online without dialling. In this case, simply press ENTER when QLTerm asks for the number.

(D)ownload

This command initiates an XMODEM receive. It asks for a filename (including device name); the file to be downloaded will be stored under this name.

Immediately after entering the filename, the receive session will start. Transfer progress is shown on the bottom line (each block = 128 bytes). QLTerm supports Xmodem with both one-byte checksum and two-byte CRC error detection; it will use CRC mode if the remote supports it, otherwise checksum mode.

The incoming data will be buffered before writing out to the selected device to improve performance (see the Buffer command). If there is insufficient free memory available for the buffer to be allocated, the command will be aborted with an error message. In that case, a smaller buffer size has to be specified using the Buffer command.

(H)angup

When using a Miracle Systems QL Modem, or Modaptor+ with modem control through the DTR line, this command can be issued to put the modem offline. For QL Modem, this is always required after the session has been terminated, since the modem doesn't disconnect the line automatically.

When using other modems, the Hangup command cannot be used. These modems must be put offline using their own appropriate method if they do not automatically disconnect.

(I)nterface

With this command, various communication options such as baud rate, parity etc. are controlled. All options are set using menu's, where the arrow-left and arrow-right keys can be used to select the desired setting, and the ENTER key stores the selected value.

The first option set is the **transmit mode**. This may be set to "Echo Off", "Echo On" or "Local Only".

"Echo Off" is used in normal, full-duplex terminal operation. Characters typed are sent through the line, with no local echo (the remote is responsible for any echo).

"Echo On" is usually used in half-duplex communications, or when QLTerm is used in "user-to-user" terminal communication. In this mode, characters typed are echoed to the local terminal screen **and** sent through the line. This way, the user can see his/her own typing without the remote having to provide the echo.

"Local Only" doesn't send any characters to the remote but does echo them on the local terminal. This may be useful when using the Transmit command to display logged files (notably when they contain ANSI- or VT52 control sequences).

Next, the **modem type** is set. This may be "Qmodem", "Modaptor" or "Buffered". "Qmodem" should be selected when using the Miracle Systems QL Modem, "Modaptor" when using the Miracle Systems Modaptor or Modaptor+ in conjunction with non-buffered modems, and "Buffered" when using neither QL Modem nor Modaptor.

Note: the term "buffered modem", in this context, means that the modem is equipped with internal hardware that "interfaces" the data stream coming from the signal decoder to that on the serial connector. As these modems often contain their own microprocessor to control their operation, these may also be called "intelligent modems".

Unbuffered modems (manually controlled) usually give poor results at 300 or 1200/75 bps, due to an error in the QL's 8049 second processor which causes data corruption. The Modaptor(+) should be used with these modems to reduce this problem. Another solution is replacement of the 8049 by the Hermes chip.

Next, the **port** is set. This may be SER1I, SER1H, SER2I or SER2H. With the "I" flavours, the output handshake lines (DTR on SER1 or CTS on SER2) will be ignored, which is suitable for the QL Modem and other non-buffered modems which don't require (and perhaps even don't have) a handshake for their transmitted data. The "H" flavours do use the DTR or CTS line, and are required with most buffered modems (especially when they use a terminal-to-modem data rate which is higher than the line speed).

Note: The "I" and "H" parameters only affect the handshake that controls the flow of **outgoing** data as seen from the QL. The handshake of **incoming** data, via CTS on SER1 or DTR on SER2, is entirely controlled by the QL's second processor. Although the QL manual says that input handshaking should always be used, this is not always required. Non-buffered modems can be used without any handshaking at all, if only one of the serial ports at a time is used. Buffered modems using a fixed terminal-to-modem data rate of course require a handshake in both directions, but this shouldn't give much problems when the connection with the QL is properly wired.

Next, the **baudrate** is set. This may be one of the standard QL rates of 75, 300, 600, 1200, 2400, 4800, 9600 or 19200 bps, or one of four 1200/75 bps split-speed rates. In the latter case, some explanation may be required.

The QL's serial ports, in their native form, do not support split-speed rates (i.e. transmit speed different from receive speed) as required by unbuffered modems supporting the CCITT V.23 standard.

An early solution to this problem was the **Modaptor** by Miracle Systems, which was later superseded by the **Modaptor+**. Using simple electronics, these devices could generate the 75 bps transmit baud rate out of 1200 bps (which however requires a specially generated output signal, which is different for the "native" and "plus" version!). In the **12M75** and **12P75** modes, QLTerm will generate this special signal for the native Modaptor (12M75) or the Modaptor+ and QL Modem (12P75).

A more recent solution is the replacement IPC processor called **Hermes**. Apart from solving the serial input problems, it also allows you to have a different input- and output baudrate. When selecting **12H75**, QLTerm will place the currently used port in 1200 bps receive mode while setting the transmit speed at 75 bps. On QL's which don't have Hermes fitted, this option will not work properly (the QL will still receive at 75 bps!).

If you neither have a Modaptor nor Hermes and still wish to connect a 1200/75 bps modem directly to the QL, you can use the **12S75** option. This will generate the 75 bps transmit signal using a software algorithm that directly operates the TxD line. Using this has several disadvantages though - it is timing-critical and works only on native QL's without a Gold Card attached.

Finally, the **parity** can be set. This may be **None**, **Odd**, **Even**, **Mark** or **Space**, according to the requirements of the remote system. Most amateur Bulletin Boards use no parity.

Log

This command allows logging of received data to a file. When the Log command is issued, it asks for the (full) filename. It is recommended that RAMdisk is used, as other devices will usually not be fast enough or tend to disable interrupts at certain intervals, possibly upsetting or slowing down serial I/O.

There are two possible log modes: **Raw** and **Cooked** (set by the **Options** command). In Raw mode, VT52- and ANSI control sequences will be logged as well, in Cooked mode they won't. The latter is useful if the logged data is to be used as a normal textfile, but Raw mode allows you to preserve colour and animation within the log.

If CR/LF conversion has been activated (see the **Options** command), then incoming carriage return characters will be suppressed. This is useful when the logged data is to be read in as a QDOS textfile, which do not use CR as line termination. If the file is to be "played back" later using the Transmit command, LF conversion should be turned off.

To close the log file, the Log command should be issued again while the logging is active.

Quit

This command quits QLTerm and returns you to the process from which it was started up. If a call has been originated using the Call command and the modem is still online, QLTerm will ask you if you want to disconnect it before leaving.

Options

This command allows the terminal emulation mode, colour use, Log mode and LF conversion to be set.

The **terminal emulation** can be either **ASCII**, **VT52** or **ANSI**. The first mode is the simplest of the three, offering only basic cursor control. VT52 and ANSI modes allow for better cursor control and colours. ANSI is very popular amongst amateur Bulletin Board systems.

Colour can be set to NO or YES. When switched off, the default display colours are used and any VT52- or ANSI colour control sequences received will have no effect. This may be useful when using a monochrome monitor. Also, since the QL's display will be in 4-colour mode and ANSI supports sixteen colours, there is a possibility that certain colour combinations will lead to "invisible" text when different fore- and background colours translate into the same paper- and ink colours. Turning colour off avoids this problem.

Logging (Raw/Cooked) controls whether or not VT52- and ANSI control sequences will be logged. See the discussion under the **Log** command for more detail.

Convert LF's for LOG & TRANSMIT will, when enabled, convert the end-of-line sequence used in terminal communications (i.e. CR+LF) convert to QDOS convention (Log) and vice versa (Transmit).

Transmit

This transmits a file as if the contents were typed from the keyboard. It is in fact a simple ASCII upload, but when the Transmit Mode (see **Interface** command) has been set to Local Only, it can be used to playback an earlier logged file (this allows for any VT52- or ANSI control sequences when enabled).

If the "Convert CR/LF" option in the **Options** command menu has been set to YES, any LF's will be converted to a CR to enable QDOS text files to be transmitted.

During transmission, QLTerm will pause 1 second after each line sent to give the remote enough time to digest the data. The transmit process can be aborted by pressing ESC.

Upload

This command sends a file to the remote using the XMODEM protocol. It should be activated once the remote is ready to receive the file.

After entry of the file name (which has to be specified in full), QLTerm will ask if you want to send the file header to the remote in front of the file contents. This header is specific to QLTerm and contains information stored in the QDOS file header of the file to be sent. When sending certain files (notably EXECutable files), it may be necessary to send the header along, otherwise it won't be EXECutable once received at the other end! Also, sending the header will preserve the exact file length, otherwise some "junk bytes" will be appended to the transferred file since the XMODEM protocol uses fixed-length 128-byte blocks and does not record the file length in any way.

Headers added this way will only be recognised by QLTerm, and a few other QL terminal programs. It is therefore recommended that headers should not be sent unless strictly necessary (EXECutable files).

The file transfer will now be initiated, and block- and retry numbers shown on the status line as with Download.

APPENDIX 1: KEYBOARD- AND DISPLAY CONTROL CODES

Cursor keys

When certain cursor key combinations are being pressed, QLTerm will send a code, the function of which will generally match the usage of the corresponding QL cursor key. Note that, in many cases, not all codes shown here will actually be recognised by the remote system.

The code sent when a cursor key combination is being depressed depends strongly on the terminal mode used. The following combinations will however always send the same code:

<u>Key(s)</u>	<u>Code</u>	<u>Hex</u>
ENTER	CR	0D
CTRL LEFT	DEL	7F
F4	XOFF	13
SHIFT F4	XON	11
F5	CTRL-C	03

In ASCII mode, the following codes are also used:

<u>Key(s)</u>	<u>Code</u>	<u>Hex</u>
LEFT	BS	08
RIGHT	HT	09
DOWN	LF	0A
UP	VT	0B

In VT52 mode, the following codes are used:

<u>Key(s)</u>	<u>Code</u>	<u>Function</u>
LEFT	ESC D	Cursor left
ALT LEFT	BS	Backspace
RIGHT	ESC C	Cursor right
TAB	TAB	Tabulate
CTRL RIGHT	ESC K	Erase line from cursor
UP	ESC A	Cursor up
ALT UP	VT	Vertical tab
SHIFT UP	ESC I	Reverse line feed
DOWN	ESC B	Cursor down
ALT DOWN	LF	Line feed

In ANSI mode, the following codes are used:

<u>Key(s)</u>	<u>Code</u>	<u>Function</u>
LEFT	ESC [D	Cursor left
RIGHT	ESC [C	Cursor right
UP	ESC [A	Cursor up
DOWN	ESC [B	Cursor down
TAB	ESC [I	Tab
SHIFT TAB	ESC [Z	Backward Tab
SHIFT UP	ESC [V	Previous page
SHIFT DOWN	ESC [U	Next page
CTRL RIGHT	ESC [@	Insert character
CTRL DOWN	ESC [L	Insert line
ALT UP	ESC [S	Scroll up
ALT DOWN	ESC [T	Scroll down
SHIFT ALT LEFT	ESC [Space @	Scroll left
SHIFT ALT RIGHT	ESC [Space A	Scroll right
CTRL ALT LEFT	ESC [M	Delete line
CTRL ALT RIGHT	ESC [K	Erase from cursor to end

Display control codes

QLTerm recognises various control codes (depending on the terminal mode) to control screen functions. The full summary follows below.

ASCII mode:

<u>Code</u>	<u>Hex</u>	<u>Function</u>
BELL	07	Alarm
BS	08	Backspace
HT	09	Horizontal tab
LF	0A	Line feed
VT	0B	Vertical tab (cursor up)
FF	0C	Clear screen
CR	0D	Carriage return
DEL	7F	Delete character left of cursor

VT52 mode (in addition to ASCII):

<u>Code</u>	<u>Function</u>
ESC A	Cursor up
ESC B	Cursor down
ESC C	Cursor right
ESC D	Cursor left
ESC H	Cursor home
ESC I	Reverse line feed
ESC J	Erase to end of screen

ESC K	Erase to end of line
ESC T c	Set ink colour to "c" (1)
ESC U c	Set strip colour to "c" (1)
ESC V c	Set paper colour to "c" (1)
ESC Y y x	Set cursor to position x,y (2)

Notes:

1. Character values for "c" are: "@" black, "A" blue, "B" red, "C" magenta, "D" green, "E" yellow, "F" cyan and "G" for white.
2. Character values for "x" and "y" are the column value plus 20 hex and the row value plus 20 hex, respectively.

ANSI Mode (in addition to ASCII):

<u>Name</u>	<u>Code</u>	<u>Function</u>
CHT	ESC [n I	Cursor Horizontal Tab
CNL	ESC [n E	Cursor Next Line
CPL	ESC [n F	Cursor Preceding Line
CUB	ESC [n D	Cursor Backward
CUD	ESC [n B	Cursor Down
CUF	ESC [n C	Cursor Forward
CUP	ESC [n;n H	Cursor Position
CUU	ESC [n A	Cursor Up
DSR	ESC [6 n	Device Status Report
ED	ESC [s J	Erase in Display
EL	ESC [s K	Erase in Line
HPA	ESC [n `	Horizontal Position Absolute
HPR	ESC [n a	Horizontal Position Relative
HVP	ESC [n;n f	Horizontal & Vertical Position
IND	ESC D	Index
NEL	ESC E	Next Line
RCP	ESC [u	Restore Cursor Position
RI	ESC M	Reverse Index
SCP	ESC [s	Save Cursor Position
SD	ESC [n T	Scroll Down
SGR	ESC [s m	Set Graphic Rendition
SL	ESC [n SPC @	Scroll Left
SR	ESC [n SPC A	Scroll Right
SU	ESC [n S	Scroll Up
VPA	ESC [n d	Vertical Position Absolute
VPR	ESC [n e	Vertical Position Relative

Notes:

1. Most control codes have an optional numeric parameter **n**, represented in ASCII. When absent, a default value of 1 is taken for most control codes. For codes ED, EL and SGR a string consisting of numeric parameters, separated by a semicolon, may be given.

2. The codes ED and EL erase parts of the display, as follows:

ESC [0 J (or ESC [J)	Erase from cursor to end of screen
ESC [1 J	Erase from beginning of screen to cursor
ESC [2 J	Erase entire screen
ESC [0 K (or ESC [K)	Erase from cursor to end of line
ESC [1 K	Erase from beginning of line to cursor
ESC [2 K	Erase line containing cursor

3. The code SGR sets the colour and other attributes of characters subsequently received:

Parameter value: 0	Revert to normal, white on black
4	Underline
7	Reverse paper and ink colours
30	Ink black
31	red
32	green
33	yellow
34	blue
35	magenta
36	cyan
37	white
40	Paper black
41	red
42	green
43	yellow
44	blue
45	magenta
46	cyan
47	white

4. The code DSR will generate a reply in the form ESC [y;x R, where y and x are the current vertical and horizontal cursor position, in decimal (1-based).

APPENDIX 2: XMODEM PROTOCOL OVERVIEW

The XMODEM protocol, designed by Ward Christensen, transfers a single file in blocks of each 128 bytes. After receipt of a block, the receiver sends a response code, either ACK (06 hex) when the block has been received without errors, or NAK (15 hex) when the block has been received with errors. A NAK will also be sent at the beginning of the transfer to initiate sending of the first block, and when the receiver times out during block receive.

The protocol has two versions, the difference being the way in which transmission errors in data blocks are detected. The original protocol specification uses a one-byte modulo-256 checksum at the end of each data block to detect errors. Since there is a significant chance that this method fails to detect certain errors (for instance an error in bit 7 of two bytes occurring within the same block), a new version has been designed which uses a Cyclic Redundancy Check (CRC). This uses two bytes, calculated by feeding the 128 data bytes through a polynomial function using modulo-2 arithmetic. This way, nearly all sorts of errors will be detected.

The new CRC-based protocol is designed in such a way that it is backward compatible with the old checksum-based protocol. When the transfer starts, the receiver initiates by sending the character "C" rather than NAK. If the sender supports the CRC method, it will respond by sending the first block. If not, it should ignore the "C", after which the receiver will revert to checksum mode and send a NAK. QLTerm supports both the CRC and checksum method.

As sent, the 128 data bytes of a block are preceded by a SOH character (01 hex), the block number (one byte) and the block number's complement. The block number starts at 01H (for the first block), increases by one for each subsequent block, and wraps to 00H for block 256. The data bytes are followed by either the one-byte checksum or the two-byte CRC (which is sent high-order byte first).

When the last block has been sent, the sender responds to the receiver's ACK with an EOT (04H) character, which should be ACKed by the receiver as a normal data block. Note that the last block is always 128 bytes long like other data blocks, so transferred files will always be a multiple of 128 bytes long.

Header-protocol used by QLTerm

When Uploading using QLTerm, a header may be sent along with the file being transmitted. This is useful especially with EXECutable files, since the dataspace of these files is kept in the QDOS file header. Also, sending a header will preserve the exact file length, preventing XMODEM appending up to 127 "junk-bytes" to the file.

QLTerm, along with some other QL communication packages, will recognise the header during receive and restore the QDOS file header when writing the file to disk.

When uploading text files, sending a header is **not** recommended as the header contains binary data. Furthermore, it should be noted that this header is specific to QLTerm and does **not** form part of the XMODEM protocol.

The header is stored in the entire first block of the transmitted file. The first 64 bytes of this block contain 32 times the 2-byte value \$4AFB. This is followed by the 64 bytes of the QDOS file header.

APPENDIX 3: QLTERM REVISION HISTORY

This chapter describes the changes since version 2.30.

Version 2.35 (09/11/2016, re-released under GPLv3 licence in 2019)

- This is another quick-fix release to 'fix' the problems caused by the different colour handling in v2.34. On platforms supporting only the standard QL screen mode of 512x256x4, blue ink is converted to black which doesn't show up on a black background! To avoid this, pre-2.34 versions converted blue to red.
The Colour option in the Options menu now has three modes: Black and White, 4-Colour and 8-Colour. Black and White should be obvious, 4-Colour has the same behaviour as pre-v2.34 versions and 8-Colour behaves as in v2.34.

By default, QLTerm will auto-detect the colour capabilities of the platform it runs on. If it runs on a standard QL, it will place itself in 4-colour mode; if it detects an extended graphics environment such as QPC2 it selects 8 colours. This behaviour may be changed by configuring QLTerm with the supplied qltconfig_bas program.

Note: when you run QLTerm in 4-colour mode on a platform that supports 8 colours, you still see more than 4 colours but blue will be converted to red. Also, MODE 8 on a standard QL is not supported as this doesn't fit the requirement of at least 80 characters per line.

Since platforms such as QPC2 support more than 8 colours, it would make sense to make these available in QLterm too. However this requires a different palette setting which I need to sort out first.

Version 2.34 (28/10/2016)

- This is a quick-fix release to improve usability of QLTerm as a Telnet client.
After some experimenting, I found out that QLTerm can be used as a Telnet client by calling it from SuperBASIC with a previously-opened channel to the TCP device which is present in QDOS emulators such as QPC2, Qemulator and uqlx. A copy of the SuperBASIC code is included with this release as file TELNET_BAS. You can merge this into your BOOT code so that you can use the command:

TELNET "<name or IP of BBS>"

to connect to a Telnet-based BBS (e.g. Telnet "bbs.bredenbeek.net", note that quotes are required as this is a string parameter).

This release is based on version 2.33 which was completed in 1994 but never released. Apart from the changes in v2.33, version 2.34 contains a few changes to make use over the TCP device more reliable (in particular, ANSI escape codes are now sent as a single packet which is required by some BBS software) and the colour rendition has been fixed so that the full 8 QL colours are now available if the hardware supports it.

- Colour handling has been fixed so that blue is no longer changed to red. When using QLTerm under QPC2, eight colours are now available in hi-res mode.

Version 2.33 (27/09/94)

- Corrected major bugs in file logging (which produced totally unintelligible data!). Introduced "Cooked" log mode (which filters out VT52- and ANSI escape sequences).
- When the serial channel was redirected, the status line display went haywire. Fixed now...
- The parity bit wasn't always stripped from incoming data when specified. Fixed now.
- Renamed **Screen** command to **Options** and added Raw/Cooked log option.

Version 2.32 (20/12/93)

- Added new baudrates 19200 and 12H75. The first one is obvious, the second one generates a 1200/75 bps split-rate in conjunction with the **Hermes** replacement IPC chip.
- QLTerm now displays a status line at the bottom instead of the rather boring copyright message.
- Added option to set serial port to SER1I, SER1H, SER2I or SER2H.
- XMODEM upload now accepts a "C" as NAK, even when not the first block. This fixes problems with newer QBOX versions.
- Fixed a bug (introduced in 2.31) that caused the Call command to hang after dialling.
- Made some cosmetrical changes to the Screen- and Interface commands.
- At last: an English manual!

Version 2.31 (23/07/92)

- Fixed bug in ANSI processing that caused QLTerm to hang when a DSR (Device Status Report) code was received.
- Speeded up the display significantly.

Version 2.30 (16/07/88)

- Initial release...

=== END OF MANUAL ===