Realterm: Serial Terminal

***Realterm* is an engineers terminal program specially designed for capturing, controlling and debugging binary and other difficult data streams. It is the best tool for debugging comms.**

§ **Contents**

News
Latest News

RSS
Feed

Sourceforge Project Page
ChangeLog

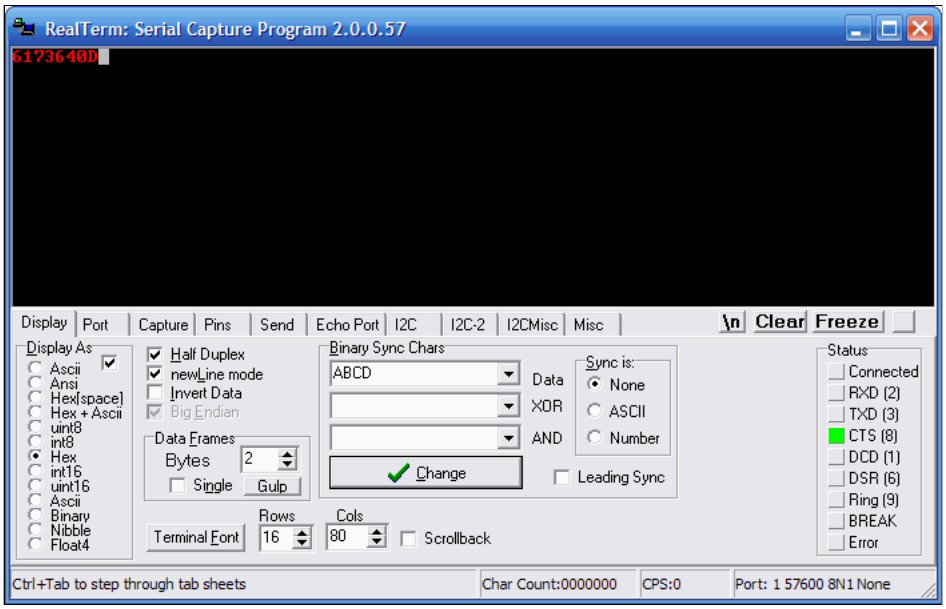
V3 Beta version under development (3.0.0.30). See Latest News

- Custom Timestamp format
- Capture Restart and autonaming for long logging.
- Post-Processing capture files

Donate : Donations pay for Delphi, code signing certificates , and more Realterm for *you* .

- Code Signed Exe's and installer now for Win 7,8,10
- Serial Ports, USB Serial and TCP/IP & Telnet
- I2C Bus, SPI& 1-Wire chip control via BL233B / I2C2PC
- Binary viewed as hex, 8,16,32 bit,little/big endian, signed, unsigned, float
- Fullscreen, MiniTerminal, Screen Scaling
- Global Hotkeys (system-wide) to send strings
- colorised: rx and tx data are different colors
- ansi/VT100 terminal or plain text or binary modes
- protocol analyser / "portspying" mode
- fixed frame sizes/line lengths
- sync patterns with masks and xors, and display only match data
- Timestamps on pattern match or newlines
- full remote control through activeX/COM
- extensive command-line for batch files and INI files
- can be used for serial I/O component of other programs via activeX. Full support for minimize,hide,iconize, tooltray
- special ascii+hex font to see hidden control chars
- capture to file, set capture size or capture duration
- timestamping capture files for simple data logging
- capture-restart, auto filenaming, data post-processing
- view and change control lines(cts,rts, dcd etc)
- easy to send binary sequences
- arbitary baud rates
- hideable to run in invisible or on tool-tray
- Bluetooth BLE support for I2C2PC-BLE and HM10,11,15
- reset / power buttons for PicProgrammer

- **Read the popup hints carefully! They give lots of information. Many controls have hidden functions**
- **Right mouse context menu has many special functions and Hotkeys**
- **There is an "examples" directory to see how to use it in scripts and with other programs**



Help & Hints:

Press **F1** to bring up the help screen. Amongst other things you get a list of the actual commandline parameters that version supports.

Tool Tips are the primary source of help and explanation when using Realterm. Take the time to move the mouse over every control, and read the hints that pop up. If you doubleclick on the status bar at the bottom it will toggle to a longer hint string.

The Popup hints are also displayed (and don't timeout) on the status bar. Double click the status bar to show them the full screen width.

ENTER send CR, <ctrl>+ENTER sends LF, <shift>+ENTER sends CRLF

§ Installing Realterm

§ Download

Monitor New File Releases

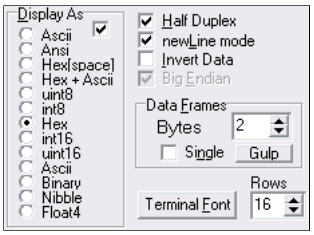
- Download and run the installer. Realterm should be installed by an Administrator user on XP, Vista, Win7,8. Active X should be registered automatically.
- Special TermHex Fonts should be installed automatically. If you need to, install the font file TERM_HEX.FON from control panel -> fonts. This font is also useful in editors etc.
- Example directory will be created with more up to date examples than you will find here.
- Make shortcuts with all the configurations you need. (see commandline options) eg "Port=2 baud=9600 flow=2"
- Subscribe to RSS News feed for development versions, and documentation changes. This is now the main news notification
- Monitor New File Releases to be notified of updates. USE RSS NOW!
- If you are using an FTDI usb adaptor you need to install the FTDI driver first
- To use Spy mode, or PicProg functions, install the separate drivers required.
- Read all the Tool Tips carefully. Check the examples directory for programming examples.
- F1 brings up the About page with lots of links, and helplinks.
- Project Page on SOURCEFORGE is the place to find all up-to-date information, make bugreports etc.
- Help is *Here*

back to **contents**

---

§ Display Formatting

Reaterm displays data in different ways to suit. "Display As" selects the display emulator:



- **ASCII All Chars** displays All chars including control codes.Hex Font lets you see non-printable values. Special chars like BS,FF,TAB are not actioned.
- **ANSI/VT100** is terminal emulation, interpreting ANSI escape sequences - best for formatting complex text.
- **Ascii Plain** displays only printable chars, and actions control codes
- Others (including Ascii plain) are **Hex** emulators and display binary values in various ways, and do binary sync.
  - Data can be synced with byte/char sequences, or selective displayed *on match*
  - Data can be in 1 or 2 byte binary and 4 byte int/float views, and Big and Little endian
  - Data can be inverted (pager IC's do this)
  - Timestamps can be added in terminal

§ Ansi/VT100 Terminal Emulation

This gives a traditional terminal with extensive control codes that navigate the cursor around and clear selected parts of the screen. If you need to format the display use this emulation, not ASCII. [The ASCII emulator displays all 256 characters, and so will not even act on *backspace* or *formfeed. It does not properly act on CR and LF, so as to display them best*]

http://www.termsys.demon.co.uk/vtansi.htm

http://wiki.bash-hackers.org/scripting/terminalcodes

Notes:

- When positioning cursor, {Row,Col} begins at 1,1 not 0,0
- If using dynamic screen redraw, eg moving progress bars, you need to uncheck LAZY display checkbox (3.0.0.27+), so that screen is redrawn after every character
- VT100 codes don't include color commands. (Feel free to add them to the apro terminal library)

§ Terminal Colors

Colors can be set from the commandline (V2.0.0.64+), or on the Misc tab.
Colors are set by a string of color chars below. The sequence is: Kbd,Port,SendStr,SpyTX,SpyRX,Background
Default is 'RYLRYK'

'R': clRed;
'G': clGreen;
'B': clBlue;
'C': clAqua;
'Y': clYellow;
'M': clFuchsia;
'K': clBlack;
'W': clWhite;
'T': clTeal;
'P': clPurple;
'L': clLime; (bright green)
'O': clOlive;
'N': clMaroon;

Tray Icon & Popup Menu

Right mouse click on the main window or on the Tray Icon  tp bring up the popup menu. You can doubleclick the tray icon to hide/show Realterm (ie make it disappear from the taskbar)

The Tray Icon and main icon changes to show a red dot  when it is capturing. The dot rotates as data bytes are actually being received. The dot is Green for normal chars, Red when capturing,

Yellow when Data Triggers or Binary Sync Matches occur. 

Small tablets, high resolutions screens etc, can make the default screen difficult to use. There are several ways to change the size of Realterm

Scaling:

The whole UI can be scaled to make all text and controls bigger. See *Misc*

FullScreen Mode:

For small tablets, the app can be run in fullscreen mode. See *Misc* tab, and right click menu, or command line WINDOWSTATE param. You can increase the COLS to fill the screen by double-clicking COLS. Use the Hotkey to toggle screen mode <ctrl+alt+F>

MiniTerminal:

Sets up a small Realterm, hides the controls and sets StayOnTop. Both Rows and Cols are changed when you drag resize the MiniTerminal. Use the Hotkey to toggle screen mode <ctrl+alt+M>

§ Hiding Controls / *FullTerminal*

Hide Controls either from the popup menu, or the commandline or activeX interfaces. This is ideal for making a shortcut that sets up Realterm for your field staff or users, then hides all the controls, to make it less confusing. Use the Hotkey to toggle screen mode <ctrl+alt+H>

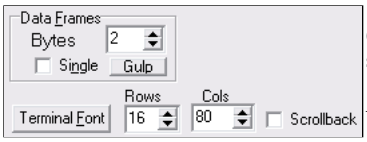
Show / *Hiding Everything*

Visible option that will completely hide Realterm. Unlike minimising, it disappears from the taskbar. Only the Tray Icon is left.

If you want it to be totally hidden the activeX interface lets you hide even the Tray Icon. This is ideal if (like us) you have 16 Realterms running in the background at once, all the time.

mechanism uses the window name, and this disappears when the window is hidden.

Setting Rows and Columns



Rows sets the *min* number of rows.

Drag resizing will increase the actual number of rows (but not change the setting).

Cols are not changed by dragging, except in MiniTerminal mode. Double Click in *COLS* will set number of cols to fill the screen width. The status bar will show the *actual* Rows & Cols during resizing.

Bytes and check "Single". Realterm tries to set the actual number of Cols, to be multiples of Data Frame sizes.

You can set the display rows from the commandline to launch it the size you want.

The example shown will be a 80x16 window.

back to **contents**

---

§ Frame Sync

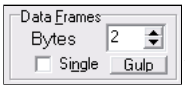
Binary data is arranged in frames. These frames are either

- Fixed size N byte frames
- variable length, delimited by a *trailing* byte sequence
- Single byte *Leading Sync* where the first char of the line delimits lines.

(of course Text is arranged as lines ending with LF or CR -ie numbers 10,13)

Sync starts a new line, and optionally highlights the last sync char, and adds a timestamp. Sync can also be used to *only* display chars following the sync - N chars, or (N=-1) until next CR/LF

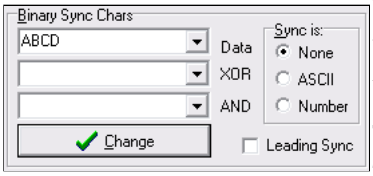
A count of sync matches is shown on the Binary Sync Chars panel. When a sync match occurs, the tray icon square changes to yellow for a few seconds.



Fixed size frames are self evident. You will notice that theterminal resizes to always have a whole number of frames across, unless "Single" is checked.

Unfortunately frames will randomly begin somewhere on the line.GULP swallows a character each time it is pressed. Press it until the frames correctly start at the beginning of a line.

Delimited frames start a new line when they detect the sync sequence. A sync sequence can be any number of bytes long



Here sync is detected when 2 bytes match 0xA55A, or moreaccurately, when 0xA55A XOR 0x0000 AND 0xFFFF > 0. ["SyncIs"should be set to "Number"]. If you are syncing off ASCIIchars, then select "Synch is: ASCII" and put the chars in the top editbox.

You can have as many bytes as you want in the sync word.

The XOR term allows you to invert some or all data bits. ($00 is normal $FF is inverted). *If you don't need the XOR and AND fields, just leave them empty to get the defaults.*

The AND term lets you ignore some of the bits. For example youcould use this to use bit 7 as a sync bit, by setting the AND term to $80 $80. (Note hex numbers are preceded by $)

back to **contents**

---

§ Baud Rates & Ports

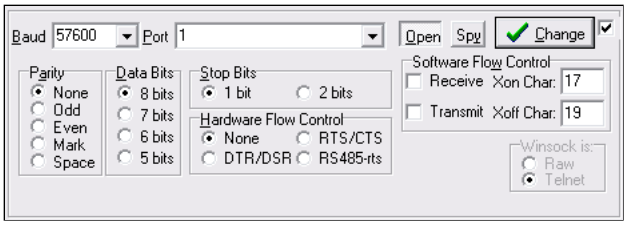
Baudrates depend on the exact hardware port. Realterm accepts anything. Some ports complain about invalid baud rates, others just ignore them, some coerce to the nearest rate.

Most PC ports accept non-standard values that the chips divider is capable of generating.

Realterm can connect to both SERIAL ports (real uarts, as well as USB, and network virtual uarts) or TCP/Telnet ports.

- Windows Serial Port# eg "2"
- Port Name from Registry ifpreceded by "\" eg "\VCP0" or "\Serial0"
- ip_address:port eg 192.168.20.1:23

- port can be a number or servicename eg "telnet"
- server:port eg "server:telnet" or "server:9876"



USB Virtual Comports

USB serial ports appear at some port number. Look under "my computer->properties->hardware->ports" to find where they are. Unfortunately the same device will often appear at different comport numbers when it is on different USB hub ports.

The currently present devices and associated ports are also listed in the registry at HKLM\HARDWARE\DEVICEMAP\SERIALCOMM. You can use these device names directly in the port selection eg "\VCP0" or "\Serial0"

For example when you plug in an I2C2PC adaptor it will normally appear as \VCP0, regardless of the comport# that it is assigned, or which usb hub port it is connected to.

On Win9X there don't seem to be any special names for devices, and this won't help you.

Virtual Comports - Talking to Software.

If you would like to use Realterm to interact with software on your PC, not with a physical port as normal, the you need com0com . This makes a pair of virtual comports, linked together. You software connects to one port, and Realterm connects to the other. Now Realterm can talk to your software and see what it is sending.

§ Scanning for Ports, Startup Delay, and Bluetooth

Version 2.0.0.70 onwards

Realterm now uses the registry to find ports rather than trying to open them all. Note if you get a registry key error when starting, this is because you do not have any serial ports installed. Install one.

Exhaustive Search by Opening

V3.0.0.28+ If you have ports which do not show up (e.g. com0com VCP's) then, at the end of the Port dropdown is [Exhaustive Search by Opening] . This scans for ports by trying to open every port number (see below). The SCANPORTS commandline option sets the highest port that will be tried.

Versions Before 2.0.0.70 (and Win 9X)

Realterm scans by trying to open every comport number.

Realterm will scan for ports at startup if an explicit port is not given on the commandline. This can cause long delays where Bluetooth is running. The SCANPORTS commandline option sets the highest port that will be tried.When starting normally, Realterm will try to open the first existent port that it finds. (V2.0.0.57+)

When started as an ActiveX automation server, it does not open the port until explicitly requested.

§ What Baud Rates does Realterm support?

Realterm will pass *any* requested baud rate through to Windows. Many other applications have a list of baud rates, but Realterm does not, it will request anything.

Mostly, if a baud rate is not accepted, there is no error or warning - it just does not work. Whether a baud rate will be work depends on two things Reaterm has no control over:

- Will the driver and/or Windows accept a requested but possible rate?
- Can the hardware divide its clock down to the requested rate?

Microsoft says this : *"For all other cases, as long as the requested baud rate is within 1 percent of the nearest baud rate that can be found with an integer divisor, the baud rate request will succeed"*

The most basic PC uart has a maximum baud rate of 115,200. Any frequency of 115,200 / N, can usually be requested. More modern PC's and laptops, usually have a higher maximum baud rate of 230,400, or 460,800, or 921,600. Actual serial ports usually have a maxium that is a multiple of 115,200. Once you have found the maximum, any baud rate of <Max Baud Rate>/N should work.

USB-Serial adaptors usually have higher clock rates, and support a wider range of different rates. FTDI's usb-serial adaptors have high maximum rates, and many possible rates. See FTDI info:

- http://www.ftdichip.com/Support/Knowledgebase/index.html?whatbaudratesareachieveabl.htm
- http://www.ftdichip.com/Support/Documents/AppNotes/AN232B-05_BaudRates.pdf
- http://www.ftdichip.com/Support/Documents/AppNotes/AN_120_Aliasing_VCP_Baud_Rates.pdf

If the actual PC baud rate is within 3% of your actual device baud rate, it will usually work OK. If you are using RS232 connections, modern drivers will usually work at 230kbd for short cable lengths, but are less likely to work at 460kbd, or 921kbd

Baud Rate Multiplier and 16C95X Uarts

The16C95X family of advanced uarts are able to support very high baud rates. They have 64byte FIFO's, which is a give away that they are in your serial card. When trying to get very high baud rates, there is an option in the hardware configuration of the uart to enable the "baud rate multiplier". This will result in higher than requested baud rates. ie the actual baud rate = requested rate * multiplier

§ RS485

Microsoft says:

> *For all versions of Windows > NT AsyncPro handles the toggling of RTS via the RTS_CONTROL_TOGGLE flag and the Windows SetCommState function. I have found other people complaining about a "significant lag" when using this flag to control RS485 devices from non-ASyncPro programs. This seems to be a problem with Windows in general and not AsyncPro in particular.*

Here is a quote from a user in a different forum:

> *"I tested the RTS_CONTROL_TOGGLE mode with RS485 devices, and I noticed that Windows has a significant delay between the end of transmission and the control of the RTS line, so this mode does not work properly with a RS485 equipment that replies "too fast" for Windows, due to a conflict between the RS422 amplifiers simultaneously active on the RS485 line. Your best bet might be to get an actual RS485 card for your windows system that handles the RTS toggle in hardware.*

§ TCP/IP: Telnet and Raw modes: Missing FF's

The TCP connections default to using Telnet protocol. However if you are connecting to a socket with raw data, you might notice that some characters **(eg 0xFF)** are missing or doubled up. You need to change between Telnet and Raw modes. This is not a bug!

The telnet connection lets two copies of Realterm talk to each other on the same machine. Just set the first to "server:telnet" and the second to "127.0.0.1:telnet". This is very useful for testing and experimentation.

back to **contents**

---

§ Fonts

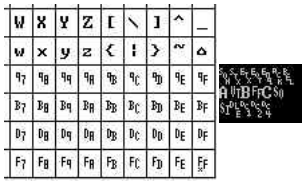
By defualt Realterm should be using the Hex fonts (below). The terminal can be changed from *Terminal Font* button. Only fixed pitch fonts show, and only some of these seem to be satisfactory. *Lucidas Console* and *Fixedsys* are good starts. Commandline options FONTNAME and FONTSIZE allow you to make a permanent change.

e.g. **`realterm.exe FONTNAME="lucida console" fontsize=11`**

§ Hex Font

Our Hex fonts are included. The Installer should install the fonts for you automatically. You can also go to the windows font installer in Control Panel to install it.

The hex font contains all 8 bit values. The non-ascii values <32 are shown as either HEX or CONTROL chars, depending on the font you select. (There are 3 different fonts in the .FNT file)



This is very useful for seeing control codes, invalid hidden codes and errors, in serial comms. It's equally useful in a programmers editor.

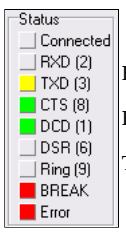
(Note that you won't see them in ANSI mode, as the control codes will be processed)

You can now get just the fonts from the downloads page. **If you can convert these fonts for use with Linux or another OS, please do! If you would like to add a larger size to the font, please do.**

back to **contents**

---

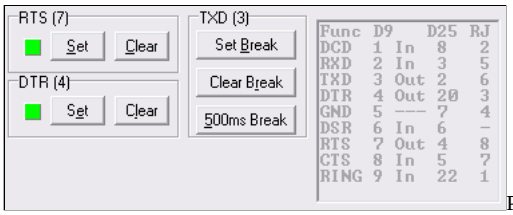
§ Pins & Status



Handshake Pins and comms status can be monitored.

Handshake outputs can be controlled directly (and from the command-line, and via activeX)

The error cause is displayed when you hover the mouse over the error light.



Pin states can be set manually. Set means data flow is enabled. Note that if CTS/RTS or DTR/DSR handshaking is enabled, then you cannot control that pin from the buttons.

back to **contents**

---

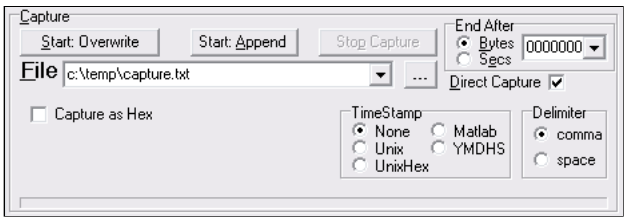
§ Capture

Incoming data can be captured to file. The capture can automatically stop (and restart) after a certain time or number of chars. Capture files can be post-processed. Realterm can be hidden, and capture controlled from the tray icon. , popupmenu, and automation interfaces. Combine capture with filesend to make simple datalogging applications. Data lines can be Timestamped . For more about capturing from the commandline see: Do then Quit

This provides a very easy way to (say) collect serial data, and graph it live using Matlab.

It can either capture "direct" or via the terminal window. When you use DIRECT capture, the terminal window is turned off, and the echo port operation will cease. This means less processor load, screen draws etc. This is best for embedded type uses. (V3.0.0.28+ Display checkbox enables terminal in Direct mode to check data)

If you want to capture what you are seeing in the terminal, don't use Direct Capture.



Char Count and CPS (chars per sec) are displayed 

during capture.

The Tray Icon and main icon changes to show a red dot  when it is capturing. The dot rotates as data bytes are actually being received.

Capture As HEX

Sometimes it is easier to look at binary data when it is saved as hex. So each received char is converted to two hex chars and saved to file. This option only works with Direct Capture. For best speed don't do this: capture normally, and use a binary/hex editor to examine the file

Restarting, AutoName, and Long Captures

Some users leave Realterm capturing for long periods. Files can grow very large, or be unwieldy to find data in.

Restart and AutoName (V3.0.0.28+), lets Realterm begin a new file, after N secs or bytes, with the DateTime appended to the filename. Previously this was done via a batch file.

eg to capture temperature data, with a new data file automatically created every day, and the datetime suffix in filename.....

```
realterm capsecs=86400 capautoname=1 capfile="c:\temp\temperature_log.txt" capture=3
```

Post Processing

When the capture file is closed, it can be post processed by a batch file. (V3.0.0.28+). This could be used to:

- Zip
- Thin/Decimate lines using SED
- Convert e.g. Hex2CSV, Bin2Hex
- Upload to web FTP, SCP,SFTP, WGET
- Graph or display

e.g. To capture daily data and compress it:

```
realterm capsecs=86400 capautoname=1 capfile="c:\temp\temperature_log.txt" capture=3
capprocess="%ProgramFiles%\BEL\Realterm\Utils\postZip.bat"
```

§ Diagnostic Files: Trace and Log

RealTerm can also write LOG and TRACE files to help debugging difficult serial problems. These are completely separate from the Capture function and provided by the Turbo Async comport component. These are a reports from a dignostic queue. You can CLEAR the queue or DUMP it to a file. The Log buffer is 10000 long and Trace buffer 1000 long (V2.0.0.69). Select *hex* if you need to see non-printing chars. For more information see the Turbo Async Reference Guide , Chapter 2, "Debugging Facilities" (pg 33)

back to **contents**

---

§ Timestamps

(V3.0.0.24+) Timestamps can be shown in the terminal, or added into capture file.

§ Terminal Display Mode Timestamps

The timestamping behaviour is different for different ShowAs emulators

- **ASCII All Chars** will add timestamps as the first char of a new line is received.
- **ANSI-VT100** display no timestamps - they don't make sense
- **HEX (all others)** , the time stamps are triggered by a SYNC match, and show the time of the last sync match char.

Custom timestamp format can be set by right-click on Timestamp checkbox. (V3.0.0.30+)

Capture File Timestamps

Timestamping is very useful for data logging, or where you want to know when an occaisional string arrived. This is most useful for comma separated (CSV) type text data. Timestamp is triggered by CR or LF.

Unix timestamps are the number of seconds from 1/1/1970.

Matlab timestamps are floating point days since 0 Jan 0. Matlab timestamps are given to the PC's clock resolution, this should be 10ms for NT and later and 55ms for Win98 and earlier. Using Matlab timestamps should give you finer resolution than 1 second.

UnixHex is provided for convenience when all the data being captured is in hex. In this case the whole file including timestamps can be converted to decimal by the *HEXCSV2DEC* utility that is bundled with Realterm.

System Format (labelled YMDHS before v3.0.0.30) timestamps display the time using the system locale formatting function. This means that it varies with your preferences and place to place (month and day will swap with country). (It also has some slightly odd behavious like not including the HMS value what it is 00:00:00)

§ Custom Timestamp Format

(V3.0.0.30+) uses a Delphi custom format string . Default is "2016-05-29 15:47:23". This will always be the same from system to system. The format string can be set from the command line by TIMESTAMP=<format string>. You can edit it by right-click on the TimeStamp radiogroup. If you are using the commandline, and double-quotes, note how they are escaped:

```
TIMESTAMP="'""'yyyy-mm-dd hh:nn:ss'""'"
```

Note that Unix, Matlab, and format yymmddhhnnss will sort (in data analysis). System and other customary date formats do not sort.

Timestamping also slows down file capture somewhat, so it is probably not ideal for very fast and dense data streams.

back to **contents**

---

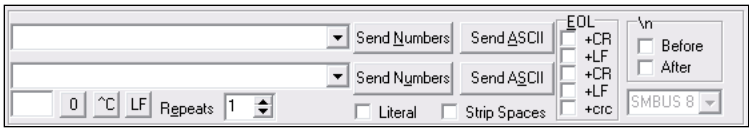
§ Sending Char Sequences

Read the popup Hints carefully. They give lots of information especially in the Send tab!

Often you want to send special chars strings or strings repeatedly, or send them quickly.

Both ascii and binary strings can be sent. Ascii strings include python escape sequences eg \n for LF. See table below. Literal disables the escape sequences and send just the string. Sent chars aren't echoed to the terminal.

(V2) If half-duplex is set, then sent strings will be shown on the terminal. This also applies to strings/chars sent via the ActiveX interface.



When Sending ASCII, you can optionally end the line with CR and/or LF. You can also strip spaces from sent data. This is useful for the I2CChip adaptor. While it ignores spaces, they take time, and buffer space, and won't be used in the final application. (but they make it much easier to read!)

(V3.0.0.27+) Send AsHex, expects just hex, without $, 0x, h, etc. It ignores all chars that aren't hex. You can just enter hex, with commas and spaces if you want.

Commandline commands STRING1 and STRING2 push strings into the comboboxes, so you can pre-load them. SENDSTR , SENDNUM , SENDHEX , and SENDLIT will send string directly from the commandline

Python Escape sequences:

Escape Sequence	Meaning
\\	Backslash (\)
\'	Single quote (')
\"	Double quote (")
\a	ASCII Bell (BEL)
\b	ASCII Backspace (BS)
\f	ASCII Formfeed (FF)
\n	ASCII Linefeed (LF)
\r	ASCII Carriage Return (CR)
\t	ASCII Horizontal Tab (TAB)
\v	ASCII Vertical Tab (VT)
ooo	ASCII character with octal value *ooo*
\x*hh...*	ASCII character with hex value *hh*...

Hotkeys

(V3.0.0.25+) Local hotkeys work throughout now. Look at the popup menu to see what local Hotkeys exist.

- Terminal copy/paste
- Window: Fullscreen, Miniterminal, Hide Controls
- SendStrings
- Terminal Clear, Newline

Global Hotkeys can be enabled from the SendStrings popupmenu. This allows F key to send the strings from anywhere in Windows - even when Realterm does not have focus or is minimised.

Loading Canned Strings, Loading SendString drop-downs

From the ActiveX interface you can prepopulate the Send dropdowns with strings by using the “AddCannedString” function.

The String1, String2 commandline options load strings into the dropdowns. If you want to preload lots of strings, or long strings, then use an INIFILE. (3.0.0.24+) Previous versions used SendStr, this is now used to send strings.

Sending ASCII with special/non-printable values

When you press "*SendASCII* " any valid backslash sequences are converted to special values in the style of Python.

If *Literal* is checked, then the string is sent raw. Note that the special chars must be lower case

\n	LF (0x0A)
\r	CR
\a	BEL
\b	BS
\f	FF
\v	VT
\t	HT
\xXX	hex value. Also accepts \0xXX
\OOO	Octal value like Python /C. 1-3 digits can be used. (Before V3.0.0.30, was decimal, NOT octal like python)
\\	\ (backslash)

Sending Non-printable binary chars

You can do this in several ways.

- In the terminal window you can send most control chars by holding the control key down eg ctrl+m = CR. (You cannot send 0x00 or 0x03 this way)
- You can send special chars NUL(0x00) and ^C (0x03) using the buttons
- You can enter a string of hex or decimal numbers in the Send comboboxes eg "51 0x31 $32" and press "*Send Numbers*
- To send a series of binary numbers use the unmarked edit box.Type in numeric values here, separated by spaces. When you hitSPACE, each char will be sent. Chars can be decimal, or hex eg "13""$1A" or "0x1A". Just keep typing out numbers.

CRC and Checksum

You can add a variety of CRC's and Checksums to the *end* of the send string. See the drop down list.

Modbus binary packets end with a 16bit CRC. This can be appended to each string sent.

CRC's can be sent as HEX-ASCII (V3.0.0.30+) , this is useful when they are appended to ASCII data. eg NMEA.

NMEA packages the ascii message string with $ ... *

§ Sending Files

You can dump a file directly to the port. Files are sent *raw* , the exact bytes in the file are sent out. There is no "protocol" and just sends everything. (versions before 1.14 swallowed ^Z / 0x1A). Hex values, python/c style backslash sequences etc are ***NOT*** converted to anything, just sent literal.

If you want to send binary values you need to create your file with a binary or hex editor. eg Frhed , wxHexEditor , MultiEdit

Padding File Dump with Delays

Two delay settings are provided to add delays after each char, and at the end of each line. (EOL is denoted by CR at present). This affects file dump, but not sending char sequences above.

Sending Repeatedly

You can set the number of repeats, and the delay after sending the file, .

These are particularly useful from the commandline for data logging, when combined with capture.

For example you can make a simple, one line file that commands a multimeter to read a voltage. Set repeats to 0, so it will loop for ever, and delay to 1000ms, and Capture. Now the data will be read to file every 1 second.

§ Data Logging

Using Capture and SendFile from the commandline, you can log data and control intruments directly from the commandline, without extra software.

```
realterm.exe senddly=10000 sendrep=0 sendfile=commands.txt capture=results.txt
```

This will send "commands.txt" endlessly, with a 10sec pause between sends, and capture the replies to "results.txt". This is all you need to do to turn (say) and RS232 multimeter into a datalogger.

(V2.0.0.46) The TIMESTAMP option can be used to prepend a timestamp to each line. This is most useful for CSV type text data. See capture section

back to **contents**

---

§ I2C & SPI & 1-Wire Bus

I2C, SPI, 1-Wire and other Serial buses can be read using the I2CChip products "I2C2PC" and the "BL233B" IC. *Realterm cannot read I2C without these external devices. It cannot use printer ports, the PC's SMBUS interface etc.*

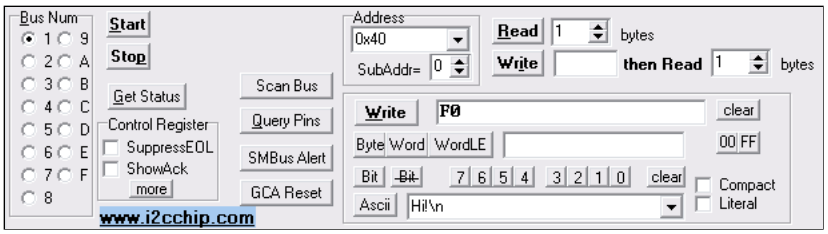
The provides a USB/RS232 interface to 3 I2C buses. Realterm is an easy way to use it. Using the ActiveX interface you can easily send strings from excel or other apps. Using the Echo port provides a way to make hardware devices that can be controlled over the internet.

Common commands you want to send to play with the adaptor are provided by the controls. Using this you can quickly try out an I2C adaptor or 1-wire device.

- Write to any I2C Address
- Read N bytes from an I2C Address
- Write then Read and I2C Address. (used to read from I2C devices that have internal registers
- Read SMBus Alert register
- Control adaptor pin states directly (bit-bash pins), and read pin states
- Read Status register
- Read a Dallas 1-Wire ID chip
- Write to 4 digit 7 seg LED display modules

- Control various special I2C chips: BL301, PCA9544, PCA9545,MAX127,

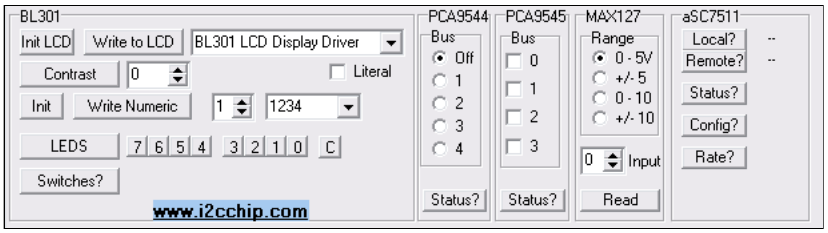
The main I2C tab has controls to select one of the serial buses.



Refer to the datasheet for the BL233B chip used, where its commands are detailed.

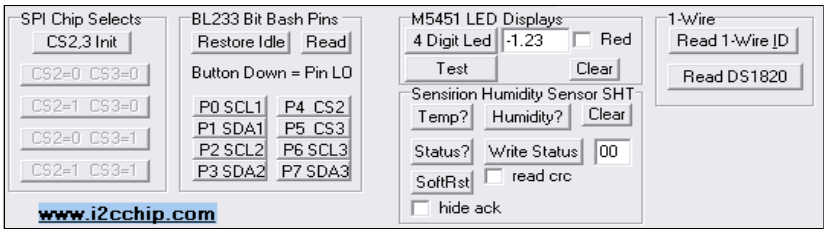
Note that you can use it with SPI, Dallas 1-wire, and other serial IC's, as well as I2C

The I2C-2 tab has controls to support a number of common IC's. The MAX127 12Bit Precision ADC and PCA9545 Bus Switch/multiplexor are available as standard modules from*I2CChip*



The tab has support for non-I2C devices:

- SPI Chip select pins
-
- 4 Digit, 7 Segment LED and LCD display modules based on MM5451. These are standard modules from *I2CChip*
- Dallas 1-Wire devices



---

§ Echo Port: Redirect Ports across Network

The main port can be passed through or echoed to the Echo Port. This is partcularly useful when the echoport is a TCP port. This allows a real serial port to be aliased across the network. (the echo port can be a real comm port too)

Lets say the remote (unattended) machine (192.168.0.99) has a datalogger connected to COM1. It runs Realterm at startup with a command line like this:

```
realterm -port=1 -baud=9600 -echo=server:9876 -caption=Mirror_Multimeter_To_Internet
```

ie realterm connects to the datalogger on COM1 at 9600 bd, and presents a telnet server on port 9876 (or any other suitable number).

On the local (attended) machine run a copy of Realterm like this:

```
realterm -port=192.168.0.99:9876
```

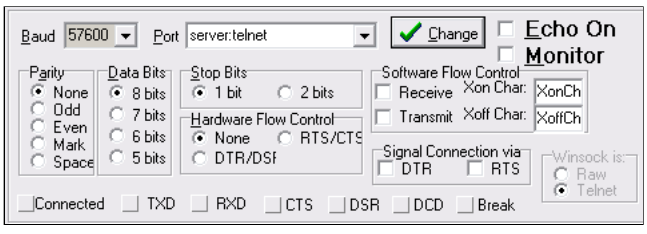
This makes a telnet connection to the remote machine. Now you can sit at your desk, and control and monitor the remote serial device.

Note that as Realterm has a full ActiveX interface, you can use Windows remote DCOM to start,stop and control the remote Realterm, as well as the local copy.

When there is a connection, chars are echoed. When the connection is broken, or the buffers are full for some reason, it simply stops attempting to echo chars. When the connection is broken, the server end just waits for another connection. At the client end, the port need to be manully restarted.

If you are using the activeX interface you can check the open property to see if the link is up, and use it to re-establish a dropped link.

The Monitor checkbox lets the terminal window display both sides of the conversation (ie both the data received through the ain port, and the echo port)



Signalling Winsock Connection state to Comport

When a telnet/winsock connection is being echoed to a physical comport, we can use one of the comports handshake output lines (DTR,RTS) to signal a remote system whether winsock is connected. You must make sure that this doesn't interfere with the operation of the handshake lines. (ie use DTR for signalling if you are using RTS/CTS handshaking for flow control)

Com0Com, Com2TCP: Virtual Comports

Echo port an connect a physical port to the newtwork. However if you want to connect other software on your PC, you need Virtual Comports. See http://com0com.sourceforge.net Com0Com . Com0Com can make vcp's with numbered ports - Realterm can use these, and named ports which RT cannot use. Com0Com can also make 3 and N way connections that can be used to create a tap that RT can connect to, for monitoring.

back to **contents**

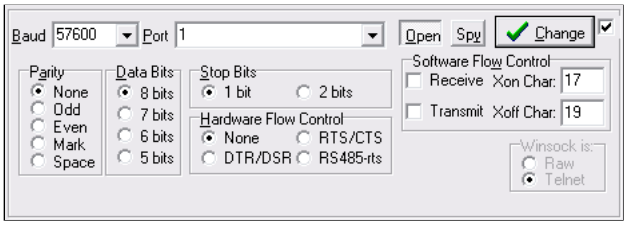
---

§ Monitor RX and TX data: Protocol Analyser

Realterm has two ways to monitor serial communications, and let you see the RXD and TXD data interleaved in the same terminal window:

- "Spy" on Port tab uses aspecial driver to intercept the comport messages and display them
- "Monitor" on EchoPort tab uses 2 comports anda special cable to monitor the actual RS232 cable

Spy mode allows you to monitor the communications between a program running on your PC and the com device, by installing a device driver to intercept the port messages,
Spy mode must be activated before opening the comport, and the comport must be closed before spy mode can be released. Spy button is on the port tab.



Note that you cannot currently capture the spy mode. However you can copy and paste from the terminal window.

Monitor Mode

Monitor mode is useful to monitor communications between external devices by connecting to the RS232 cable.

The Echo port can be used to give you a second receiver. The data is put into the terminal screen, in a different color. Now you can see both sides of the conversation.

The interleaving on screen, only shows you when the data arrived at Realterms handlers. The indeterminate delays in Windows mean that you can't rely on the sequence being exactly as it happened. Obviously with slow data, or decent gaps between send and receive, it will work better than with very fast data streams.

Special Monitor Cable

You need a special adaptor with 2 plugs for the PC's 2 serial ports. Only connect RXD and GND on those plugs at the PC end.

§ Monitor and Echoing

You can select both Echo and Monitor. Now Realterm echos as normal, but the terminal window displays the data from both directions.

back to **contents**

---

§ Command Line Parameters

The command line is morphing into a general purpose Command system. You can run INIFILEs, send commands to running isntances with first, and execute commands from ActiveX/COM using DoCommands. Future versions will allow command strings and hotkeys.

Realterm doesn't save its settings. Instead it is set up from either the command line, for basic setups, or using its extensive ActiveX interface. It is also possible to send commands to a running instance of Realterm from the commandline. (fix 3.0.0.25+)

These examples show the command-line params you can use. Generally they set the corresponding widgets. For radiobuttons and checkboxes, a number selects the state. For booleans (eg visible) use either 1 or 0. (n.b. No "/" or "-" before parameters). Double-quotes enclose values with spaces or other control characters eg \. Double quotes inside strings are possible (V3.0.0.30+) eg `send1="abc=""def"""`

For complicated settings, a file of parameters can be loaded using INIFILE, and settings generated and saved from the "INI File" button on the Misc tab. (V3.0.0.18+). When debugging problems, the *View Params* button will show what Realterm *actually* received on its commandline.

§ INIFile Dialog:

- Create an INI file with all current settings in it, and edit that to only keep the ones you want to set
- Save and Load INI files
- Execute / test whole infile with *Execute* button
- Execute single lines in editor by double clicking them, or selections with *Selected* button.
- See the *actual* last Command Line with View Params
- Go direct to the commandline help for parameters by double clicking a parameter in the *All Parameters Help*
- list all paramters actually recognised by the program. (may be more complete than this help list below)

Example Command Lines

- realterm.exe baud=9600 port=1flow=2 capfile=c:\temp\junk.dat visible=0 display=5 bigend=1 capcount=9876 framesize=7
- realterm.exe port=server:telnet
- realterm.exe inifile=realterm.ini
- realterm.exe scale=auto windowstate=full
- realterm.exe port=127.0.0.1:21
- realterm.exe RTS=1 DTR=0
- realterm.exe tab=Send
- realterm.exe echo=server:9876

- realterm.exe capfile=junk.txt capsecs=10 capture
- realterm.exe sendfile=junk.txt chardly=3 linedly=50
- realterm.exe capture=in.txt sendrep=10 senddly=10000 sendquit=out.txt
- realterm send1="abc=""def""" (loads string abc="def" V3.0.0.30+)

§ Table of Commandline Parameters

§ BAUD, BD	#	Set the baud rate. Non standard baudrates are fine
§ PORT, PT	Port or IP	sets the port.
§ PORTQUIT, PQ	Port or IP	Sets the port, exits if port does not exist or can't open
§ DATA	7E1	Sets DataBits,Parity,StopBits. eg DATA=7E1 DataBits is 8-5, Parity is None,Even,Odd,Mark,Space, StopBits 1-2
§ FRAMESIZE, FS		terminal frame size. Interacts with COLS. Use one or other
§ CAPFILE, CF		name of the capture file to use
§ CAPCOUNT, CC		Length of capture in bytes (either CAPCOUNT or CAPSECS)
§ CAPSECS, CS		Time of capture in SECS. (-ve values makes it quit as soon as idle or until timeout). If using SENDQUIT, then waits this time after send finishes, before stopping capture. See SENDQUIT. (V3.0.0.17+)
§ CAPTURE, CP	0/1/2/3 filename	Capture starts immediately. Stops at count OR secs. Can use CAPTURE=filename Capture=0 is off; 1 is ON; 2 is APPEND; 3 is RESTART
§ CAPAUTONAME	0/1/2	Append DateTime to filenames. Used with RESTART for logging (V3.0.0.28+)
§ CAPPROCESS	0/<filename>	Set Capture Post-Processing batch file (V3.0.0.28+)
§ CAPQUIT, CQ		Capture (as above), but program quits when capture ends. If you manually stop capture, then Autoquit is cancelled. Can us
§ CAPHEX, CX		Capture as Hex ie turns all chars to a 2 char hex value.
§ CAPDIRECT CD		Capture Direct checkbox
§ TSDELIMITER	Delim char	Set timestamp delimiter eg TSDELIMITER=%
§ TIMESTAMP	0-5, <format >	During capture, Prepends a timestamp to each line. Only works in text files with EOL character. A custom format string ca
§ VISIBLE, VS	0/1	starts hidden, only tray-icon is visible. 0=Hidden,1=Visible
§ DISPLAY, DS	#	sets the display format (ascii, hex,int etc). eg DISPLAY=5 Put *before* other display params such as COLS (v3.0.0.30+) -ve values for signed int eg DISPLAY=-8 for Int16
§ BIGEND, BE		set big-endian checkbox
§ FLOW, FW	0,1,2,3, X, 0-3+4+8	Sets hardware flow control mode. FLOW=X to enable XON/XOFF mode or Flow=N+8 for RX-XOn, +4 for TX XON or Xon/off: +8=TX, +4=RX. eg FLOW=10 is RTS/CTS+RX-Xon/off. Also you can use R or RTSCTS or D or DTSDSR
§ RTS	1/0	sets RTS pin
§ DTR	1/0	sets DTR pin
§ CLOSED	1/0	Starts with port closed (default is open)
§ TAB		Selects the opening tabsheet by name or tab number (case in-sensitive)
§ ECHO		Sets the echo port, and enables echoing
§ EBAUD	n	Set the baud rate for echo port.
§ EDATA	7E1	Sets Echoport DataBits,Parity,StopBits. (not working properly yet)
§ HALF		Sets HALF Duplex
§ CAPTION		Sets window caption (can't accept spaces, use underscores). See caution below if using with FIRST
§ SENDFILE, SF § SENDFNAME		Sends the file immediately. (capture started before sending starts) Just set send filename.(V3.0.0.27+)
§ CONTROLS	1/0	Hide Controls and expand terminal window to full screen. (depreciated V3) Show controls.
§ MONITOR	1/0	Monitor Echo port RX onto terminal
§ CHARDLY	#	Sets delay (in ms) after each character when sending files and strings
§ LINEDLY	#	Sets delay (in ms) after each line when sending files and strings
§ ROWS	#	
§ COLS	#	Set number of cols. (n.b. FRAMESIZE interacts with this, use COLS last)
§ SENDDLY	#	set delay (in ms) after file is sent, until next send begins. If Send Delay is set, SENDREP will be set to 0.
§ SENDREP	#	set number of times file will be sent. 0 sends for ever. Should follow *after* SENDDLY
§ SENDQUIT, SQ		quit when sendfile ends.If you manually stop send, then Autoquit is cancelled. Can use SENDQUIT=filename
§ SENDSTR, SS	"string"	DEPRECIATED (V3.0.0.23-30) Use STRING1, STRING2 . Loads "string" into the Send String comboboxes. Can be calle Does not send the string unless sent to a running instance using FIRST (see below) CHANGED (V3.0.0.31+) Sends to port as escaped ASCII. Puts in STRING1
§ SENDNUM	"string"	send string as NUMBERS (see SENDSTR above)
§ SENDHEX	"string"	send string as HEX (V3.0.0.31+)
§ SENDLIT	"string"	send string as Literal (V3.0.0.31+)
§ CR	0/1	send CR after string (sets for following strings)
§ LF	0/1	send LF after string
§ FIRST	[0/1/2]	*Should be first param with the same Caption / Window Title* If only parameter, stops more than one instance running. Brings FIRST instance of realterm to front, and quits. Note this fi title (CAPTION) see details below.
§ LFNL	1/0	sets Newline Mode checkbox. See CRLF for enter key
§ SPY	1/0	Spy mode. Port you are spying on must be closed when starting. Be sure to specify the port on the command line with *port*
§ SCANPORTS	1/0	(obsolete) realterm scans for actual comports by trying to open them all.This can cause problems with Bluetooth serial dev OS can say that a port is not there. This option can be used to suppress scanning
§ I2CADD		set the I2C Address. Should be hex string eg "0x40"

§ HELP		Same as F1 key
§ INSTALL		Only used by installer for post-install special messages and behaviour.
§ SCROLLBACK	#	Enable Scrollback. Just SCROLLBACK enables, =0 disables; =N Sets SCROLLBACK lines=N (-ve disables)
§ COLORS	string	Terminal colors are set by a sting of color chars. The sequence is: Kbd,Port,SendStr,SpyTX,SpyRX,Background Default is 'RYLRYK' see Terminal Colors
§ HEXCSV	"format"	Sets format string for HexCSV formatting. (when implemented) will set the capture formatting sting to the same.
§ WINSOCK	0,1	Sets winsock to raw or telnet
§ EWINSOCK	0,1	As above for Echo port
§ INIFILE	<filename>	Load commandline parameters from a file. One param per line. (V3.0.0.24+) INIFILE is loaded after commandline has been processed. INIFILE can also be passed to a running instance by FIRST.
§ MSGBOX	Message	Show a messagebox, normally at startup. Prompt user e.g. "Plug in Voltmeter"
§ SCALE	%,0,Auto	Scale the whole application (so you can make it bigger on high pixel screens).0 or Auto will scale to fill screen (V3.0.0.23+
§ WINDOWSTATE	0-3, Full,Max,Min,Norm	Set the WindowState NORMal,MINimized,MAXimized,FULLscreen. Fullscreen is especially useful for small tablets.
§ CLEAR	1+2+4	Sets the various ClearTerminalONxxx Checkboxes. Value is sum of 1=DisplayChange, 2=PortChange, 4=PortOpen just CLEAR sent from remote will clear the terminal
§ STRING1, STRING2, S1,S2	string	Pushes strings into SendString comboboxes on Send tab. Can be called more than once to push more strings into the combo
§ CRLF	[0]1	Set CRLF checkbox. ENTER key send CRLF instead of CR
§ FONTNAME	name	These are the fontnames shown in the *Terminal Font* dialog.
§ FONTSIZE	size	Size will probably need to be set after setting font.
§ BSYNCIS	[0],1,2	Binary Sync IS select mode. (V3.0.0.29+) Put this after DAT,AND,XOR
§ BSYNCDAT	string	Binary Sync data string
§ BSYNCAND	string	AND
§ BSYNCXOR	string	XOR
§ BSYNCHI	0[1]	Highlight
§ BSYNCLEAD	0[1]	Leading Sync
§ BSYNCSHOWCOUNT	N[-1]	Show Count. Num chars to show after sync. -1 is show to end of line
§ KEYMAPVT	filename	Load Keymap for VT100/ANSI emulation (V3.0.0.30+)
§ KEYMAP	filename	Load Keymap for other emulators (ASCII, HEX etc)
§ CRC	-N,0,N	Set CRC type on Send tab. 0=none, -ve=HEX. Can use twice to set, but turn CRC off. eg CRC=-3 CRC=0 (v3.0.0.30+)
§ VERSION	<version>	Does nothing. Output by INI writer, so you can see what version of Realterm made the file. This is useful for debugging w a new version.
/REGSERVER /UNREGSERVER		Register / remove ActiveX COM Server. Done by installer, but you may need to do this when changing versions etc.

Hints: Parameters that put a string into a drop down (eg SendStr, SendNum, HexCSV), can be called multiple times to push more than a single value into the drop down.

eg realterm.exe hexcsv=uv hexcsv=st hexcsv=ab

§

If you are doing batch file automation, you will often want Realterm to do something (e.g. send a file), then quit.

Realterm has SENDQUIT, CAPQUIT, and PORTQUIT commands.

CAPQUIT can capture either CAPSECS seconds of data, or CAPCOUNT characters, then quit.

SENDQUIT sends files then quit. When capturing Realterm will continue the capture after the file send for up to 1 second if chars are still coming in.

```
realterm port=\vcp0 flow=2 capture=infile.txt sendquit=outfile.txt
```

If the return data is going to take longer to complete, then you can use CAPSECS with SENDQUIT to force the time after sending ends, until capture stops. If CAPSECS is -ve, then it will keep capturing until the input goes idle, to a max of capsecs.(V3.0.0.17+) Note that if nothing is happening it will quit quicker - capsecs sets the capture time, not the idle time.

So below, the capture will remain open, until the flow stops, (for a max of 5 seconds open):

```
realterm port=\vcp0 flow=2 capture=infile.txt capsecs=-5 sendquit=outfile.txt
```

And here capture will remain open for 20 seconds regardless:

```
realterm port=\vcp0 flow=2 capture=infile.txt capsecs=20 sendquit=outfile.txt
```

When you are calling Realterm within a batch file, you will normally want the batch file to pause until Realterm exits. You must use the batch command *START* with the */WAIT* parameter for this to happen.

```
start "Reading I2C EEProm" /wait realterm.exe port=\vcp0 flow=2 capture=infile.srec sendquit=CmdFile.i2c
```

When you look at the examples, you will see that I tend to set environment variables for the names of the files, ports, and Realterm itself. So it looks more like this...

```
start "Reading I2C EEProm" /wait %RT% port=%ComPort% flow=2 capture=%TempFileName%.srec sendquit=%CmdFileName%
```

PORTQUIT: Waiting for USB Ports

When testing USB devices via batch file, there are two problems that arise.

- Faulty devices do not have a port, and so Realterm will stay open

- After connecting a device, Windows takes a quite variable amount of time to enumerate and connect them

PORTQUIT quits the program with an error code, if the port is not able to be opened. This allows your batch file to retry until the port is open or timeout. See examples

Exit Codes and Errors

In batch files you may need to check for error exits (using errorlevel)

type TExitCode=(excNormal=0,

excNoSuchPort=1,

excPortBusy=2,

excFileError=3,

excUIPIBlockedMessage=5,

excSendFileTimeout=6

);

§ Controlling a Running Instance

(V3.0.0.24+). The first instance must be started with FIRST=1 now [security improvement].

If the sending instance shows an error message that it is blocked the try starting the first instance with realterm.exe first=2. This unblocks the UIPI protection. (V3.0.0.24+)

which has the same CAPTION / Window Title , and the second instance will terminate. This allows you to pass many of the above parameters (not all will do anything), and some special ones below. Note that Windows Scripting/ActiveX is a better way to do complicated tasks.

- realterm.exe first LF send1="S42F0"

QUIT		Quits Realterm
EXIT		same as Quit
CAPTURE	0/1/*filename*	Start/stop capture. 1 starts capture; 0 stops; *filename* starts and sets filename
SENDFILE	0/1/*filename*	start/stop send file
SENDSTR SENDNUM SENDHEX SENDLIT	"*string* "	send string out. (<V3.0.0.31 only when sent to running instance)
CR	0/1	send CR after string (sets for following strings)
LF	0/1	send LF after string
	"*numeric string* "	send numeric string as binary (only when sent to a running instance)
CAPTION		The caption (window title) is used to find the running instance see caveat below if you use this to change it.

When you change the caption of the first instance, subsequently the control instance must be given the same caption.

```
realterm.exe first caption="New Realterm"  //now the running instance has a new title
```

```
realterm.exe caption="New Realterm" first DisplayAs=5  //so now we also have to change the controller instance to be able to find it next time.
```

FIRST mechanism uses the window name, and this disappears when the window is hidden (but minimsed windows are OK)

The best idea is to create a special shortcut for each setup you want to use, and set the params in its properties. If you need other parameters added, contact us.

back to **contents**

---

§ ActiveX/COM Interface

RealTerm is an out-of-process server. Use a property browser (eg from excel or delphi) to see what it can do. (must be registered first run: REALTERM /REGSERVER)

(V3.0.0.27+) ***The DoCommands method makes all commandline parameters available to ActiveX/COM user.*** If the ActX lacks a command, check the commandline.

You can launch Realterm from another application, eg matlab ,excel , VB,OpenOffice , delphi. Alternatively use windows scripting, and write a simple .SCP file to launch and control it. You could even launch it from a web page to use as telnet client!

Registering the COM interface

This should be done automatically when you run the installer. Sometimes this fails, or perhaps you don't have the right permissions under XP or Vista.

- To register the ActiveX (onlyneeded if you want to use activex) run: REALTERM.EXE /regserver
- When the type library changes (iewe extend it) you may need to unregister it first, then register it. Run: REALTERM.EXE /unregserver

If you get EOleError messages, you need to run the command window (CMD) as administrator.
In Windows 7, search for CMD in the start menu. ctrl+right-mouse on CMD.EXE, and you get an administrator command window.

Why not the microsoft comms ocx

It has many problems. The best thing about using RealTerm is that you can see exactly what is happening when you want. Save yourself time and headaches. If you don't want people to see RealTerm, you can completely hide it.

Opening Ports and Errors

(<V3.0.0.20)Trying to open a port that is non-existent or has some other problem, results in an error, and Realterm exits.

(V3.21+) Errors are trapped, and Realterm does not exit. After opening the port, check the *PortOpen* property to see if it succeeded. The *LastErrorMessage* property is a multiline string you can examine or display. On the *Misc* tab, the *Show Last Error* button allows you to display it manually to help with debugging.

Sending Strings, Chars and 0x00

Windows has one big problem for users of binary data. Strings are *null-terminated* . ie they end with 0x00. Amazing as it seems, you can't pass a string containing char 0 (0x00) through the activeX interface.

Putstring can send any 8 bit chars except 0. If you need to send char 0 to an application use the **Putchar** function.

(v3+) **Putstring(String,SendAs)** has optional SendAs parameter so you can send ASCII Escaped strings (SendAs=1) or Numeric strings (SendAs=2) or Hex strings (V3.0.0.27+), the same as the send tab. All of these will send "123 <linefeed>". (v3.0.0.30+) **Putstring(String,SendAs,CRCType)** adds CRC to end of string. See CRC parameter

```
RT.PutString('123'+char(10));

RT.PutString('123\n', 1);

RT.PutString('49 50 51 10', 2); //numeric strings

RT.PutString('31,3233 0A', 3); //saHex ignores all non hex chars

RT.PutString('27.67Volts', 0, -1); //add SMBUS8 CRC to end in HEX
```

N.B in Excel macros note:

```
RT.clsRT.PutString "DEAD", saHex   //WORKS

RT.clsRT.PutString("DEAD", saHex)  //DOES NOT WORK
```

From Matlab From Excel

Getting Data

With the ActiveX interface you can get data in several ways:

- Capture to File. Poll properties like CaptureCount to know when to read thefile.
- Capture to File. Use Events to notify your application that the file can be read
- Use Data Trigger events
- Use WaitForDataTrigger .

Note that the capture file interface is mature, thoroughly tested and reliable.

§ Browsing the COM interface

Many languages include a browsing tool to examine the properties and methods of the COM interface. Many properties and methods have help comments that you can see. Use the browser tool.
One free tool to do this is ActiveX/COM Inspector from Oakland Software
Browsing the interface will show you how it *actually is* , as this page may be out of date.
Some languages might want the type library (.tlb). If you need it, it is part of the source package in the installer.

§ Properties & Methods @ V3.0.0.27

Open Realterm using a property editor/browser. Or open the typelibrary (.tlb) or the realterm_tlb.pas file, which are in the source directory.

(V3.0.0.27+) ***The DoCommands method makes all commandline parameters available to ActiveX/COM user.***

```
    property TimerPeriod: Integer dispid 1;
    property EnableTimerCallbacks: WordBool dispid 2;
```

```
    property CaptureFile: WideString dispid 3;
    property Capture: EnumCaptureMode dispid 4;
    property baud: Integer dispid 5;
    property Port: WideString dispid 6;
    property PortOpen: WordBool dispid 7;
    property CaptureCountForCallback: Integer dispid 8;
    property EnableCaptureCallbacks: WordBool dispid 9;
    procedure Close; dispid 10;
    procedure StartCapture; dispid 11;
    procedure StartCaptureAppend; dispid 12;
    procedure StopCapture; dispid 13;
    property FrameSize: Integer dispid 14;
    property DisplayAs: Integer dispid 15;

    property CPS: Integer dispid 17;
    property WindowState: EnumWindowState dispid 18;
    property Caption: WideString dispid 19;
    property Visible: WordBool dispid 20;
    property CaptureEnd: Integer dispid 21;
    property CaptureEndUnits: EnumUnits dispid 23;
    procedure PutString(const S: WideString; SendAs: EnumPutStringAs); dispid 22;
    function SelectTabSheet(const TabCaption: WideString): WordBool; dispid 25;
    function DiskFree(Drive: Integer): Double; dispid 24;
    property CaptureTimeLeft: Integer readonly dispid 26;
    procedure PutChar(C: Byte); dispid 27;

    function DiskSize(Drive: Integer): Double; dispid 29;

    property EchoPort: WideString dispid 31;
    property EchoPortOpen: WordBool dispid 32;
    property HalfDuplex: WordBool dispid 33;
    property HideControls: WordBool dispid 34;
    property Parity: WideString dispid 35;
    property DataBits: Integer dispid 36;
    property StopBits: Integer dispid 37;

    property EchoParity: WideString dispid 39;
    property EchoDataBits: Integer dispid 40;
    property EchoStopBits: Integer dispid 41;
    property FlowControl: Integer dispid 42;
    property EchoFlowControl: Integer dispid 43;
    property CharDelay: Integer dispid 44;
    property LineDelay: Integer dispid 45;
    property Rows: Integer dispid 46;
    property SendFileDelay: Integer dispid 47;
    property SendFileRepeats: Integer dispid 48;
    property SendFile: WideString dispid 49;
    property Send: WordBool dispid 50;
    procedure ClearTerminal; dispid 51;
    property MonitorOn: WordBool dispid 52;
    property LinefeedIsNewline: WordBool dispid 53;
    procedure NewlineTerminal; dispid 54;

    property DTR: WordBool dispid 56;
    property CaptureDirect: WordBool dispid 57;
    function AddCannedSendString(const SendString: WideString; ControlNum: Integer): WordBool; dispid 58;
    property Version: WideString readonly dispid 59;
    procedure TimeStamp(Style: Integer; Delimiter: Byte); dispid 201;
    procedure EnableDataTrigger(

    procedure OnTimer; dispid 1;

    procedure OnCaptureStop; dispid 3;
    function OnDataTrigger(Index: Integer; Timeout: WordBool; Data: OleVariant; Size: Integer;

    property LastErrorMessage: WideString readonly dispid 207;

    property Scale: Integer dispid 208;
    property Winsock: Integer dispid 209;
    procedure DoCommands(const CommandLine: WideString); dispid 210;
```

Enumerations

There are several special enumerations used, eg for putstring. If you need to explicitly use the number for these (eg some versions of Matlab), they are here:

```
  EnumCaptureMode = TOleEnum;
const
  cmOff = $00000000;
  cmOn = $00000001;

  EnumWindowState = TOleEnum;
const
  wsNormal = $00000000;
  wsMinimized = $00000001;
  wsMaximized = $00000002;
  wsFullScreen = $00000003;

  EnumUnits = TOleEnum;
const
  Bytes = $00000000;
  Secs = $00000001;

  EnumPutStringAs = TOleEnum;
const

  saASCII = $00000001;
  saNumbers = $00000002;
```

```
saHex = $00000003;
```

§ Callback Events

(V2.0.0.46+) Realterm can send events to an application. From the *Events* tab you can manually trigger events, to make it easier to test your software interface. The *Events* tab is only visible when Realterm started as an ActiveX. (But there is a "Show Events Tab" button on the misc tab)

Events usually have an associated property to enable them. By default they are disabled, so you will need to explicitly enable them before anything happens.

The Timer is a utility. It does nothing within Realterm, ie it only provides callbacks to your application. This exists as some languages do not have a convenient timer arrangement.

DataTrigger is new, (fixed: V3.0.0.9+), the interface may change in future. Note that when you are capturing a lot of data, or the data rate is high, using the a capture file is probably a better approach than Data Triggers.

```
procedure OnTimer; dispid 1;
procedure OnCaptureCount; dispid 2;
procedure OnCaptureStop; dispid 3;
procedure OnDataTrigger(Index: Integer; Timeout: WordBool; Data: OleVariant; Size: Integer; Reenable: WordBool);
dispid 201;
```

There are several properties and methods to control the events:

```
property TimerPeriod: Integer
property EnableTimerCallbacks: WordBool

procedure EnableDataTrigger(Index: Integer); dispid 202;
procedure DisableDataTrigger(Index: Integer);dispid 203;


                            const EndString: WideString; PacketSIze: Integer; Timeout: Integer;
                            AutoEnable: WordBool; IgnoreCase: WordBool; IncludeStrings: WordBool);
```

Testing Events and Events Tab

When Realterm has been started as an ActiveX/COM server, there will be an event tab. On this are controls to manully send the various events. This makes testing your code very much easier, as you can getting the event handler going, without having to get the actual serial data working at the same time.

Events in Excel, IE and some others

(V3.0.0.27+) Excel get events, and new example included in examples dir.

For some reasons events are visible to C#, Matlab and others, but not to Excel, Internet Explorer and some others. A special wrapper DLL has been created that can be used. Install Realterm_2.0.0.70_Signed_Wrapper_setup.exe or later, and it will be automatically installed.

V3.0.0.XXX Signed Wrapper is not tested with these versions yet - please test.

To see an example of web pages that use the wrapper and V2.0.0.70 from IE10, see: and the library that calls Realterm at
http://www.i2cchip.com/i2c_front_panels/commonFuncLib.js

(Obviously) only Internet Explorer supports ActiveX interfaces, you cannot control Realterm from Chrome or Firefox.

WaitForDataTrigger was provided for programs that cannot use callbacks. See below.

§

OnCaptureCount and OnCaptureStop are provided to work with capture.

OnCaptureCount is used with the properties CharCount and CaptureCountForCallback.
OnCaptureEvent is fired when CharCount>=CaptureCountForCallback. At the end of the event handler you should update CaptureCountForCallback to a new value. Otherwise the event will not happen again. Alternatively it is possible to clear CharCount.

§ Data Trigger

The default trigger is a LF character at the end of line. So it is ideal to capture ascii data lines eg from the I2C2PC adaptor.
Before using Data Triggers you should enable them.
All the methods include an index, to support multiple triggers in future. At present there is only 1 data trigger and index is ignored. Index should be set to 1.

Data triggers can be automatically re-enabled or manually re-enabled.
In your OnDataTrigger handler you can reenable the trigger by setting Reenable.

DataTriggerSet is used to configure the trigger from the COM interface. The Events tab also has a button to edit these settings.

The TrayIcon changes to Yellow when Data Triggers or Binary Sync Matches occur. 

Timeouts

Timeout numbers are in "ticks" (18ms)

Testing

The Events tab will show when Realterm is started as an ActiveX/COM.
You can manually display it using the "show Events Tab" button on the misc tab.
Now you can edit and tests the trigger. The light will flash when a trigger is matched.

§ WaitForDataTrigger

This function is provided for convenience. It is a blocking function. This means that your application will completely stall while it waits.
This is poor programming practice, and not recommended. But sometimes it is the only way.

```
function WaitforDataTrigger(Timeout: Integer): WideString;
```

Misc Notes

Don't rely on the defaults when using ActiveX (eg baud rate). These may change with different revisions.

Do: Close the port, make all changes, open the port, check port open, if not open, check LastErrorMessage.

Port opening is trapped for errors and won't bring up error dialogs or throw errors. Other functions may.

This is only a snapshot when this web page was written. Always check your actual version using a browser, as new functions are always being added. If you need other parameters added, contact us.

back to **contents**

---

§ Utilities:

Realterm has useful utilities in the *utils* directory.

These utility files are very helpful adjuncts to testing batch files with Realterm.
The UNXUTILS package has many command line utils taken from the unix/linux world.

CHOICE Prompts in batch files. Also useful to put in a pause, as it can timeout.
CTEXT Colorise text output in batch file. makes test batch files more readable
SLEEP pause batch files for time
CMP File compare which returns an error code (native windows one does not, so can't be used)
SED Modifies/search-replace in files. Useful to convert hex files to i2c eeprom programming commands, as well as remving whole lines eg keep 1 of N, remove 1st and last lines etc
NOWSTR Returns date/time as a string. Used to put date/time into filenames
CMDOW Control Window states eg batch files used for capture post-processing are launched minimised/hidden, and must be shown before any error messages can be seen. Also use to minimise or hide batch files while running.

SRECORD Convert/check/manipulate hex data files, S19, Inhex and everything else

POSTZIP.BAT Example capture post-processing batch file

§ HexCSV2DEC Utility

This command line utility is installed with Realterm and is normally found at:
"c:\Program Files\BEL\Realterm\HexCSV2DEC.exe"

With no parameters it will display its help.
It converts a file of hexadecimal data values to decimal. Unlike some other hex2dec utilitys it converts the binary format of the hex into decimal numbers. It is especially useful to convert a file of hex captured by reading I2C data with the I2C2PC and BL233, into useful decimal numbers.

HexCSV2DEC V0.3
(c)2008 Broadcast Equipment Ltd http://www.i2cchip.com

Converts HEX in CSV files to Decimal CSV files

Command Line may have these forms:
HexCSV2DEC <InFile> <OutFile>
HexCSV2DEC <InFile>
HexCSV2DEC -<option> <InFile> <OutFile>

Uses stdin, stdout if <file> is omitted
Only valid UPPERCASE hex substrings will be converted
Quoted strings, and non-hex data is passed through unchanged EXCEPT
explicit decimal/bcd strings can be passed though by preceding with "d"
eg "d1234" will be passed through as "1234"
Floats (numbers with a decimal point) will pass through
Default is Big Endian, Unsigned
Precede hex with the special explicit chars to override the default
"s" signed big endian eg "s8000"
"u" unsigned big endian eg "u8000"
"t" signed little endian eg "t0080"

"v" unsigned little endian eg "v0080"
"f" IEEE Floating point big endian (4byte singles only)
"g" IEEE Floating Point little endian
"b" Binary (only big endian)
"a" ASCII Chars

The first parameter can be options -s,-u,-t,-v to set the default format
eg HEXCSV2DEC -s <infile> <outfile> Convert hex as signed bigendian
Alternatively it can be a series of format chars for each hex substring
eg HEXCSV2DEC -sufbd <infile> <outfile> Convert hex using format string

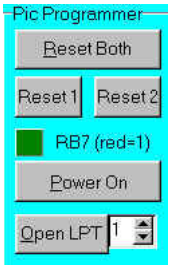
§ NowStr - Date and Time String

This returns Now as a string. It is useful to make filenames that contain the current date and time. eg to capture data to a file with a unique name including the time of capture:

back to contents

---

§ PIC Programmer



Realterm can has buttons for RESET and POWER if you are using the BEL Dual **PIC Programmer** .

Its watches the RB7 line.

This makes it easy to control a serial PIC when you are using ICP.

If you are using NT/2K/XP you need to install the DLPORTIO driver using its install program.

back to **contents**

---

§ Compiling Realterm

The current source code for your version is in the installer and will be installed in the "source" subdirectory. It is easy to recompile for specific setups.

You will need additional components. Async4.07 and Systools are on sourceforge. Other components are on the Download page.

V3.x is compiled with Delphi XE2 and Apro 4.07

V2.x is compiled with Delphi 7 and Apro 4.06 library (now free on Sourceforge)

Version 1.x was compiled with Delphi 3 and used the turbopower Async 2.11 library.

- Async Pro (windows)
- AsyncPro CLX cross platform
- Systools
- Paul Brennermans AsyncPro tips

§ Linux

Realterm has been tested with Wine 1.0, Opensuse 11.0, Realterm 2.0.0.62.
A cursory test with Com1, running as root, was successful. If you have tried this with a recent version, please report your result.

back to **contents**

---

§ Translations

The translation component is not currently working, however if you would like to volunteer, I keep a list for the day it is working. Translators are wanted for all languages. I am not sure if this version will work with non-european languages.

Email if you can help: crun@users.sourceforge.net

---

§ Who Uses it? / What For?

If you are using Realterm in your company, please drop me a line so I can include your company below...

- Rakon ,NZ. Test machinery runs 16 copies per PC, 24hr/7days, automaticallytesting Quartz crystals for most of the worlds GPS receivers
- Macfab Mfg, Canada: Upload andDownload programs to its CNC mills and lathes, and troubleshootcable runs for new machines.
- C.P. Bourg, Printing Machinery:Used to support field technicians
- Novell: Debugging Telnetconnections to mainframe through gateway
- CNC, Netherlands: Used inlaboratory with INOLAB Conductivity Meter
- JASMIN Plc, UK: Collecting datafrom a Seimens SCOOT UTC (Urban Traffic Control) system for offlineprocessing.
- Institute for Lasertechnologies atUniverity of ULM: Controlling 2 spectrometers and stepper motorcontrol in a spectral reflection interferometer
- Escort Memory Systems, USA: RFIDtags. Send intel-hex files to thier systems
- Thrane & Thrane, Denmark:Remote debugging
- Oculex: Collects data from phonesystem
- Omitec: Test and confirgurebluetooth modules
- EZ ARM :Serial terminal used for debugging and development with ARM7 EV kit
- MCSi: Setting up custom controlsystems for very large A/V and Presentation systems
- Thorlabs: test nanopositioningcontrollers during product development
- Woehler: Debugging Pressuremeasuring instruments and flue gas analysers
- SmartHome: SDK tool for protocoldevelopment
- Ditchwitch
- Cinetech: Discovering serial strings used in equipment control to develop automation
- Sparklike: debugging GassGlass devices and automated calibration
- Grand River Rubber: Monitoring and debugging Production line scale systems

Help Spread Realterm: Put a link on your companies web page today!

Also of Interest:

- **I2C** : Chips, Modules, I2C-USB/RS232 Host Adaptor
- **PIC Programmer** Windows/Linux:
- How to decode RS232 with an Oscilloscope, and pinouts
- BeyondLogic :RS232Info, and info on most any hardware port on your PC
- Serial Port Central : Heaps of serial port related links
- The Art and Science of RS-485
- :Programmers editor for Windows
- FTDI :These make an easy way to convert your serial products to USB
- Embarcadero : who make it all possible. V3 is built with Delphi XE2

§ Code Signing:

Realterm is now Code Signed, so it can install and run freely, many thanks for the many Donations that make this possible.

Thanks to Mitchell at KSoftware whose certificates are affordable, and who sorted out authentication difficulties with the obstructive folk at Comodo. (SSL certs for your website too).
(fyi: PDF's can also be signed to tamper-proof them.)

---

§ Books

---

§ Old Versions

The latest version of a program isn't always the one you want. Sometimes it works differently, or isn't as stable as an older one. If installing an old version you may need to start it as Administrator the forst time.

Old Installers

Contact Us

crun@users.sourceforge.net

back to **contents**