



wxTerm

Version 1.0.3

www.iftools.com

Inhaltsverzeichnis

1	Introduction	1
1.1	Simplest command editing	1
1.2	RS232 and GPIB connections	2
1.3	Latest version	2
2	Operating	3
2.1	The output window	4
2.2	The input window	4
	ASCII Mode	4
	Mixed Mode	5
	Hex Mode	5
	Terminal Mode	5
	Line repetition	5
2.3	The interface status window	5
	RS232	5
	GPIB IEEE-488	6
2.4	Statusbar	7
2.5	Settings	7
	Setup of the serial interface	7
	Settings of the GPIB interface	8
	The log file	8
	Send a file	8
	Rules for coloring the hex output	8
2.6	The tool bar	9
2.7	Short keys	10
A	GPIB IEEE-488	11
A.1	GPIB Status	11
A.2	Device Status	12
A.3	GPIB Line Status	13
A.4	GPIB Error	14
B	FAQs	15

INHALTSVERZEICHNIS

1

Introduction

wxTerm is a program for communication through serial ports like RS232, RS422, RS485 but also works with GPIB IEEE-488 interfaces and is used for testing purposes in the automation technology.

Even though wxTerm is called a terminal program (it sends inputs to a connected device and displays the answers) it should not be mixed up with programs like HyperTerm or minicom.

The usage for wxTerm is more in the field of automation. Everywhere where devices (measuring or other laboratory devices) are controlled via PC this program can be used. The automatic collecting of data is one example.

Especially if some devices from different manufacturers shall work together at first the exact command sequences and connection parameters have to be investigated. Sometimes the operating manuals of the devices regarding the interface commands are very short or even incorrect and shown without examples so that there is nothing else to left than to test the single commands manually.

The programming of automatic test and measuring devices which are equipped with standard interfaces (e.g. GPIB IEEE488, RS232 but also RS422, RS485 a.s.o.) requested a lot of knowledge about different programming languages. Therefore in the year 1990 the manufacturers of these devices agreed on a common language for the controlling of their devices. This language was called SCPI (Standard Commands for Programmable Instruments).

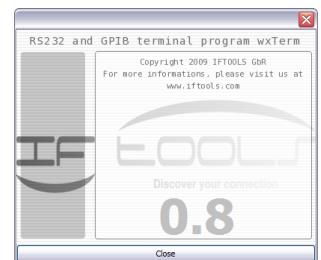
SCPI merges the manifold request and command sequences in a kind of tree structure which admittedly has the disadvantage of getting very long command strings. The usual terminal programs only offer rudimentary possibilities to edit, correct and repeat such extensive command strings.

To be sure, that the transmitted data is correct, protocols often use some kind of checksums to validate the integrity of the data, whereas the checksum is appended at the end of the data sequence.

wxTerm offers the automatical checksum generation of LTU (Longitudinal Redundancy Check) and CRC16 (Cyclic Redundancy Check, 16 bit), which are used in Modbus ASCII and Modbus RTU protocols (RS485).

1.1 Simplest command editing

Regardless of which command set, SCPI or proprietary command, wxTerm makes the communication easy by reasonable input modes where the string can be corrected before sending and also the input of control characters are allowed anywhere in the



Program start
with version display

KAPITEL 1. INTRODUCTION

send string.

The output (the data from the connected device) can be displayed as normal Text or in hexadecimal notation. In the latter mode any character or character ranges can be colored, e.g. for a better recognition of control characters.

The received data are buffered so that older data can be watched at any time.

1.2 RS232 and GPIB connections

wxTerm handles not only the communication via serial interfaces like RS232, RS422 and RS485 but also can communicate with devices over a GPIB (IEEE488) Bus (presumed a PCI GPIB card from National Instruments or compatible). Every interface can be configured in a wide range.

At the same time the information, necessary for communication or error investigation, are shown in an clearly-arranged way. That affects the single line states of a RS232 connection or the status of the GPIB controller, the GPIB Bus and the bus devices.

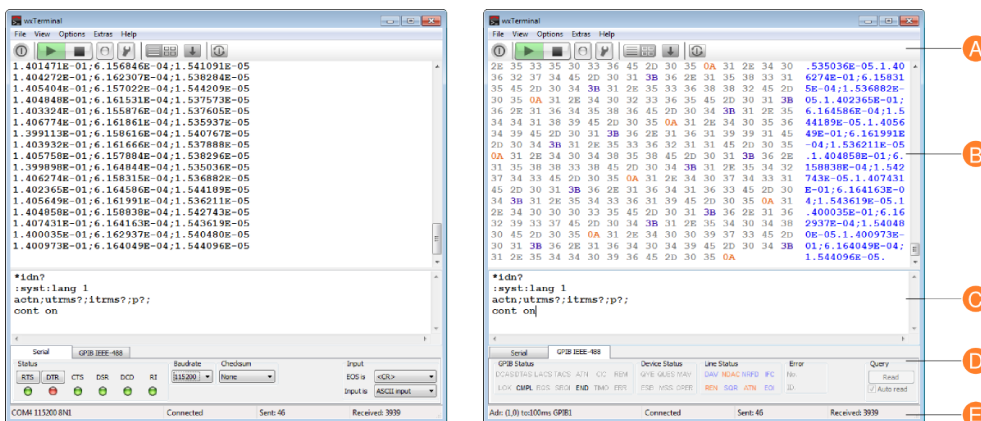
1.3 Latest version

The newest wxTerm version inclusive the automatic installer for Windows and Linux can be found at: <https://iftools.com/download/index.en.php>. You also can find the program sources of this open source program.

2 Operating

Easy sending of data and commands to the connected device and a visualization of the received answer at the same time allows a clear focus on the essential.

Inputs and outputs are done in separate windows. This has the advantage that inputs are not always interrupted and the outputs do not clutter up the window. So outputs to the device can be clearly distinguished from inputs from the device. In addition to these two windows wxTerm contains a third section to show the current interface status.



With the toolbar and statusbar wxTerm is divide into five sections:

- A Toolbar:** The tool bar is used for fast access to often needed functions like connect/disconnect interface, start logging and view selection.
- B Data output:** The received data can be displayed either in ASCII or in hexadecimal with any coloring.
- C Data input:** Input of the data including binary values to be sent to the device.
- D Interface status:** Depending on the selected interface (RS232 or GPIB (IE-EE488)) the current status is displayed.

E Statusbar: The statusbar informs about the interface name and its state, also the count of sent and received data bytes.

2.1 The output window

In this window all received data bytes are displayed. The representation can take place either as normal text or in hexadecimal notation. The latter one is useful if the output strings contain non-printable characters or if control characters shall be made visible.

Open the preference dialog and select the entry **Colors**.

You can switch between both display modes by clicking onto the respective button in the toolbar or by pressing the corresponding key combination.

Echo Mode

If you like to see your sent commands in the output window too, just enable the local echo mode in the menu View→Local Echo or press Ctrl+E.

2.2 The input window

Enter here all characters or commands which shall be sent to the connected device. The input format depends on the used protocol and/or the selected interface. Devices connected via GPIB mostly communicate with a simple ASCII protocol. Serial connections however often use binary data transmissions (which means including none printable characters) with and without an additional checksum.

wxTerm offers you an intelligent and simple input mode for every case. Beside the trivial ASCII input it also allows you to mix ASCII and binary data, input a raw hex string and operate in a terminal mode (send each hit key immediately to the device.

The input window distinguishes between four different input modes (serial interfaces):

- 1 ASCII
- 2 Mixed
- 3 Hex
- 4 Terminal

With exception of the Terminal mode all inputs are editable before they will sent by hit the Enter key.

Whether or which EOS (End of String) character is attached to the entered characters is decided by the user. You can select the EOS from a list. wxTerm will add it automatically at the end of the string to be send.

ASCII Mode

All entered characters are sent when hitting the Enter key. Until then the input can be corrected in any way.

2.3. THE INTERFACE STATUS WINDOW

Mixed Mode

Corresponds to the Cooked Mode but the character '\$' gets a special functionality. The two characters directly behind the '\$' are interpreted as hexadecimal values. Of course only if they are from the range '0' to '9' or 'A' to 'F'. This is useful to insert any control character or characters which can not be found on the keyboard into an ASCII send string.

As example the Input of \$FFHello World\$00 leads to the sending of a byte with the value 255, followed by the string 'Hello World' and a closing null byte.

To transmit the '\$' itself in this mode enter \$\$ or its hex value \$24.

Hex Mode

The hex mode is a special mode to input raw binary data in an easy way. This is very helpful, if you have to communicate via a protocol like Modbus RTU. In Hex Mode you only can input valid hexdigits like 0-9, A-F (upper and lower case) and the space as separator character between each byte. The space is necessary. The following example shows the input of eight bytes in the range of 0 to 7.

00 01 02 03 04 05 06 07 In case of a selected checksum (Modbus LTU or Modbus RTU), it will be calculated from the binary data and not from the inputed line. Means: The checksum input are the binary values 0...7.

Terminal Mode

The Raw mode is used to directly send all characters which are entered. This includes also correcting keys like backspace or the arrow keys.

Line repetition

While working with long command strings it is very annoying to enter them again and again for repeating the command in original or slightly changed.

wxTerm prevents you from this nerving action. Just hit Enter to send the current line (which includes the cursor). Naturally you can edit each line (except for the terminal mode) before you send it again.

A linefeed only occurs if you are edit/send the last line in the input window. If you like to insert a new empty line between two data sequences, just press Shift+Enter.

2.3 The interface status window

In this area all important information of the currently used interface (RS232 or GPIB) is displayed. Additionally interactive switching and settings can be done, corresponding to the interface. For example the changing of RTS or DTR lines.

RS232

The RS232 interface status shows the current condition of all modem control lines. RTS and DTR are outputs whose level can be switched by the respective key, e.g. to test hardware protocols. The display changes its color according to the current line state.

Quick baudrate switch

A quick selection of the baudrate allows to change the baudrate without closing the serial port. This selection is limited to the standard baudrates.

KAPITEL 2. OPERATING

Checksum

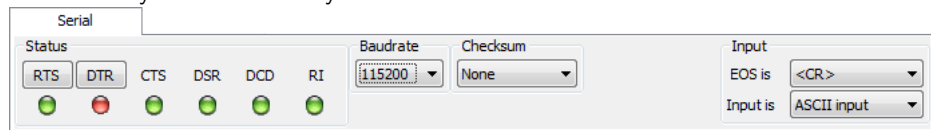
In the current version wxTerm offers two checksum generators for the LTU and CRC16. Both are mainly used in the communication with Modbus systems. The checksum will automatically appended to the data sequence to be send.

Please note! If you choosed an EOS to, the checksum will be add **before** the EOS character(s).

EOS and input mode

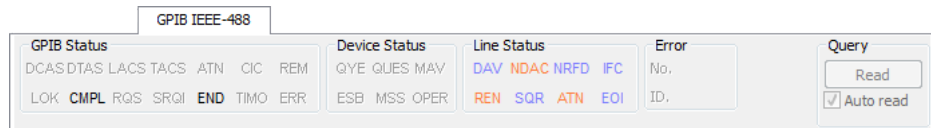
In the input window you can select the EOS (End of String) character which is automatically appended to the send string.

The input modes 'Cooked', 'Cooked Hex' or 'Raw' applies for the RS232 communication only and was already described above.



GPIB IEEE-488

The GPIB interface is a bus system with up to 15 devices (30 devices can be addressed but only 15 can be physically connected according to the specification). The display of the interface status is more complex due to the variety of conditions.



When communicating over IEEE488 wxTerm automatically handles the functions of the master or controller. The communication partner is selected in the preference dialog by entering its bus address.

The GPIB status window is divided into the following sections:

- 1 GPIB Status
- 2 Device Status
- 3 Line Status
- 4 Error

wxTerm keeps this information current in regular intervals. Not all values are active long enough to be displayed. However wxTerm is an appropriate tool for finding errors in communication or at least get a feeling where to search.

A detailed list of the identifiers can be found in the appendix.

Query (Serial Poll)

Normally wxTerm polls the connected device and reads its news and information as soon as the device signals the availability of new values. Regarding the complex cooperation between 'serial poll' and MAV bit in the device status register it sometimes makes more sense to suppress the automatic polling and start it manually by a click. By removing the default cross at 'Auto read' you deactivate the automatic polling. With a click onto the read button you can read the messages one by one while controlling the different GPIB status displays.

2.4 Statusbar

The statusbar informs you about the name and state (connect or disconnect) of the selected interface and about the count of sent or received data too.

If you clear the output window (via Strg+L), both counter will be reseted to zero again.

2.5 Settings

To keep the configuration as clear as possible all necessary settings are combined in one dialog and divided into the following areas:

- Setup for the serial interface RS232
- Setup for the GPIB Interface (requires the appropriate hardware)
- Path and name of the logging file
- Rules for coloring the hexadecimal display

Setup of the serial interface

wxTerm only offers the serial interfaces (COM Ports) for selection which are physical available AND not used by another application¹.

Beside the usual RS232 parameters like baud rate word length, parity and number of stop bits wxTerm also supports the protocols RTS/CTS and XON/XOFF.

A special feature is the possibility to set non standard baudrate like 100000 Baud. This applies only for external USB to RS232 converters which are based on a converter chip from FTDI. An example is the converter from IFTOOLS USB232CONV or ISO232CONV (for isolated applications).

- **Port selection** : Selection of the COM port which shall be used for communication. Only available and free ports are listed.
- **Standard baudrate** : wxTerm supports all standard baudrates in the range from 150 Baud up to 921600 Baud. Please note, that not all baud rates are supported by every hardware.
- **Unusual baudrates** : Activate this field if you want to set up a serial interface with a chip from FTDI and want to use a non-standard baudrate.
- **Word length** : Word length: Number of the bits which are used for one character. wxTerm supports word length from 5 to 8 bits.
- **Parity** : Beside the common parity settings none, even, odd parity wxTerm also allows the selection of Mark and Space which clears or sets the parity permanently.
- **Stopbits** : You can allow 1 or 2 stop bits. The default is one stop bit.
- **RTS/CTS** : Activates the RTS/CTS protocol. This has to be supported by both communication partners and prevents from data losses by buffer overflow in the UART (serial interface chip).
- **Xon/Xoff** : This is a matter of a software protocol where the data flow is controlled by two special data bytes Xon (decimal 17) and Xoff (decimal 19). Because the controls are part of the data stream this protocol cannot prevent from possible data losses. Therefore some manufacturers implement this protocol directly into the hardware like FTDI based converters (e.g. USB232CONV or ISO232CONV from IFTOOLS).

¹This is valid mainly for Windows. Linux systems allow the simultaneous opening of a serial port from different applications. To avoid this so called lock files can be used. But the way of this using them is not standardized and not error free so that wxTerm do not support them

KAPITEL 2. OPERATING

Please note that changes in the interface settings can only be made if it is not active. When the interface is already opened all input fields are gray and can not be changed. So you have to close the interface (close button in the tool bar) before you can alter the parameters.

Settings of the GPIB interface

GPIB IEEE-488 is a bus system with one active controller (in this case the wxTerm) and one or more connected devices. To purposefully communicate with one selected device you have at first enter its address. The other entries are already set by default with reasonable values which have the following meanings:

- **Timeout** : Specifies the maximum time the controller has to wait for the end of an operation before it signals a time out. Depending on the device you have to adapt this time. wxTerm reports a time out with set TIMO in the GPIB status field.
- **Set EOI at end of write** : If activated the controller automatically sets the EOI line while sending the last byte to mark the end of the message. Set by default.
- **Terminate read on EOS** : A message is seen as complete as soon as a EOS is received. Any character can be defined as EOS (End of String). Set this function only in connection with a valid EOS character.
- **Set EOI with EOS on write** : If set the controller adds the EOS character to the message and activates the EOI line.
- **8-Bit EOS Compare** : By default only the lower 7 bits are used for the EOS compare (standard ASCII set). If your EOS is in the range from 128 to 255 you have to activate this function.
- **EOS byte** : Enter the EOS character which is used by the device you want to communicate with. Do not forget to activate the above mentioned EOS function to make sure that the EOS is regarded and added to the message.

The log file

wxTerm saves all received data in a special logging file. That offers the possibility to examine the data later or to use it in other applications.

Path and filename can be entered here. The default target is your desktop. As soon as you operate the logging button in the tool bar all received data are stored in the named file.

Send a file

With wxTerm you are able to send any file to the connected device. Just click *File* → *Send file* or press Strg+O. The transmission starts immediately after you have select a file from the file dialog.

Please note! You cannot send a file during a closed (disconnected) interface. In this case, Strg+O doesn't work and the according menu item is out greyed.

You can abort the sending each time. The count of sent bytes is displayed in the statusbar.

Rules for coloring the hex output

The program allows for input of a maximum of 5 rules for coloring any data or data ranges. For each of these rules you can select a color from a predefined list.

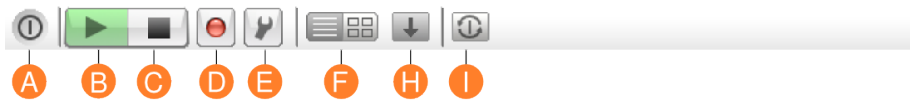
Single data bytes or ranges are always defined by a start and end value. If both are the same only this byte is displayed in the selected color.

2.6. THE TOOL BAR

All settings are stored automatically at program end. This applies also for the color configuration.

2.6 The tool bar

The tool bar is used for fast access to often needed functions.



-
- A End:** Saves all settings and closes the window.

 - B Starts the connection:** Opens the selected interface (RS232 COM port or GPIB).

 - C Close the connection:** Unlinks the connection and closes the active interface.

 - D Record:** Activates the recording of all received data and stores them in the selected log file.

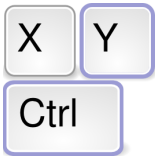
 - E Settings** Opens the setup dialog to enter the interface parameter. Selection of log file and color definitions.

 - F Display mode** Changes the representation between ASCII and Hex.

 - H Autoscroll** Activates the scrolling of the output window to always display the last received data bytes.

 - I Reset command** Depending on the interface a *Break* (RS232) or a *Device Clear* (GPIB) is sent.

2.7 Short keys



Short keys
of the most important
functions

Action	Short key
Online Help to wxTerm	F1
Opens the currently selected interface	F2
Closes the currently selected interface	F3
Input window: Send the current line to the connected device	Enter
Input window: Insert a new empty line	Shift + Enter
Shows the program version and GPIB support	Strg + A
Sends a Break	Strg + B
Toggle the DTR line	Strg + D
Switch the echo mode in the output window on or off	Strg + E
Switches to hex display	Strg + H
Clears the output window	Strg + L
Opens a file for output over the active interface	Strg + O
Toggle the RTS line	Strg + R
Saves the data of the output window into a file ctrl	Strg + S
Switches to ASCII display	Strg + T
Saves all settings and closes the program	Strg + X
Activates or deactivates the scrolling of the output window	Strg + Umschalt + S



GPIB IEEE-488

A.1 GPIB Status

With exception of the function `ibfind` and `ibdev` all internal NI-488 functions which are used for the GPIB communication controlling store their status in the global variable `ibsta.wxTerm` periodically updates the content of this variable and displays it in the status field.

The status variable is divided into 14 bits which have the following meaning:

Name	Meaning	Description
DCAS	Device Clear State	If set, a device clear command was detected.
DTAS	Device Trigger State	The GPIB interface board detected a device trigger signal.
LACS	Listener	If set, the GPIB interface board has been addressed as listener.
TACS	Talker	If set, the GPIB interface board has been addressed as talker.
ATN	Attention is asserted	Indicate the ATN signal line.
CIC	Controller-In-Charge	If set, indicates that the GPIB interface board is the active controller.
REM	Remote State	Indicates whether the board is in remote state.
LOK	Lockout State	Indicates whether the board is in lockout state.
CMPL	I/O completed	All pending I/O operations are finished.
RQS	Device requesting service	A device is requesting a service.
SRQI	SRQ interrupt received	A device or controller is requesting a service.
END	END or EOS detected	EOL has been asserted or an EOS byte has been detected.
TIMO	time limit exceeded	A timeout has occurred.
ERR	GPIB error	An error occurred during a GPIB function call.

A.2 Device Status

Every device with GPIB interface has an internal status register which can be polled by the controller in regularly intervals (e.g. 'serial poll') to react on possible requests.

The status register defines the following bits which are partially specific to the device:

Name	Meaning	Description
QYE	Query error bit	The Query Error bit indicates that the instrument attempted to read an empty output buffer, or that data in the output buffer was lost.
QUES	Question state register request	
MAV	Message available	The Message Available bit indicates if data is available in the Output Queue.
ESB	Event status bit	The Event Status bit indicates if one or more enabled events have occurred.
MSS	Master summary bit	The Master Summary Status summarizes the ESB and MAV bits. MSS is 1 if either MAV or ESB is 1.
OPER	Operation status bit	

A.3. GPIB LINE STATUS

A.3 GPIB Line Status

The GPI Bus contains 8 data lines and 8 control lines where three are used for controlling the data flow (hand shake) and five are used for bus management. The line status represents the condition of these eight control lines:

Group	Signal	Name	Description
HS	DAV	Data Valid	Indicates that the value being sent with DIO1-DIO8 is valid.
HS	NDAC	No Data Accepted	Asserted by listeners to indicate they have not yet read the byte contained on the data lines.
HS	NRFD	Not Ready For Data	Asserted by listeners to indicate they are not ready to receive a new data byte.
P	IFC	Interface Clear	The system controller can assert this line to reset the bus and make itself controller-in-charge.
P	REN	Remote Enable	Asserted by the system controller, it enables devices to enter remote mode. When REN is false, all devices will immediately return to local mode.
P	SQR	Service Request	Devices on the bus can assert this line to request service from the controller-in-charge.
P	ATN	Attention	Indicates that the data (DIO) lines contain a command byte.
P	EOI	End Of Identify	This line is asserted with the last byte of data during a write, to indicate the end of the message.

Group: HS = Handshake, P = Protocol

A.4 GPIB Error

Number	Name	Description
0	EDVR	A system call has failed.
1	ECIC	Spezifizierte GPIB Schnittstellenkarte ist nicht aktiver Kontroller.
2	ENOL	You have attempted to write data or command bytes, but there are no listeners currently addressed.
3	EADR	The interface board has failed to address itself properly before starting an io operation.
4	EARG	Invalid argument(s).
5	ESAC	The specified GPIB interface board has to be the system controller, but isn't it.
6	EABO	The I/O operation was aborted by a timeout or reception of a device clear command.
7	ENEB	The GPIB interface board doesn't exist, the driver wasn't loaded or some configuration failure.
10	EOIP	Function call can not proceed due to an asynchronous IO operation in progress.
11	ECAP	Incapable of executing function call (GPIB board lacking the capability or the capability was disabled in software).
12	EFSO	File system error.
14	EBUS	An attempt to write command bytes to the bus has timed out.
15	ESTB	One or more serial poll status bytes have been lost.
16	ESQR	The serial poll request service line is stuck on.
20	ETAB	There was a problem with a table, for instance a return buffer is full.
247	EINT	No interrupt level is configured for the board.
248	EWMD	Windows not in enhanced mode (16 bit only).
249	EVDD	The required the CBGPIB.386 device driver isn't installed (16 bit only).
250	EOVR	Buffer overflow (16 bit only).
251	ESML	Library calls running simultaneously (16 bit only).
252	ECFG	Config file GPIB.CFG not founded or doesn't match the GPIB interface board.
253	ETMR	No Windows timer available (16 bit only).
254	ESLC	No Windows selectors available (16 bit only).
255	EBRK	Control-Break pressed, exiting program (16 bit only).

B

FAQs

Frequently Asked Questions, short FAQ concerning `wxTerm`.

No GPIB setting possible

Under Linux `wxTerm` has to be compiled with GPIB support, see the topic below. In Windows the GPIB communication is already integrated into the program. However, the appropriate hardware from National Instruments, Keithley or Ines has to be available to activate the respective settings.

No GPIB support under Linux

Unfortunately the GPIB support is not yet part of the standard kernel. A respective kernel module is mandatory for the using the GPIB controller hardware and has to be generated suitable for the used kernel. Precompiled binaries with GPIB support are not possible because of the different kernel versions. To use `wxTerm` under Linux with GPIB you have to compile it yourself from the sources.

Do I need admin rights?

Under Linux as well as under Windows you can install the program as a normal user. But the program has to have access to the interface ports. Under Windows this should be no problem (If the access is not explicitly forbidden for the normal user). Under Linux the user has to own the group `tty`. if necessary the access to the GPIB entries has to be enabled.

Where do I find the source code?

The program code and all necessary drivers can be found under:
<https://iftools.com/opensource/wxterm.en.php>.

Where do I get support?

`wxTerm` is a open source project from IFTOOLS. Questions, suggestions and improvements are always welcome. But a right for support does this not include.

May I copy the program or pass it to someone other?

Yes of course. The license conditions expressly allows it. `wxTerm` is under the `wx-Widgets` license. More information can be found under:
<https://iftools.com/opensource/index.en.php>.

ANHANG B. FAQs

Unusual baud rates

I need a non standard baud rate for RS232 communication. What do I need? The used RS232 hardware (the UART) has to have the respective support. These are all converters based on the FTDI chipsets as offered by IFTOOLS and other manufacturers.

Under Windows and Linux are the converter USB232CONV and ISO232 tested. You can find them at <https://iftools.com>.