Sample rate up to 200 MHz

Time resolution 10 Nanoseconds

"The perfect combination of sophisticated hardware and software analysis"

	Port A Status RXD CTS	Fieldbus Protocol Analyzer
		Made for RS232

recording • exploring • visualizing

An essential tool for RS232 analysis/optimizing

As an autonomous device the analyzer gathers exact information about every line change with 10 nanoseconds precision, independent from the PC and its operating system and mandatory for all time relevant protocols like Modbus RTU, USS or many others. Two special inhouse developed serial input channels driven by a sample rate up to 200 MHz take over the task of data decoding either for all kind of asynchronous as also synchronous SSI transmission.

Equipped with a multitude of visualization tools the analyzer allows a detailed view of every transmission layer in a RS232 communication. It detects errors coming from incorrect data formats, invalid timings, erroneous bytes or telegram spacings or wrong telegram contents and/or checksums.

Due to its high adaptable protocol template mechanism based on the Lua script language the analyzer is not only able to visualize telegrams of common field busses - it is recommended especially for all kind of individual or closed/proprietary protocols and the perfect education tool.

An integrated schematic switch editor (optional) extends the analyzer even by the capability to interactively alter the signals. Lines can be switched, rerouted or even inverted. Data can be injected in any output line (inclusive simulation of framing and parity errors). Also break conditions can be sent.

Run it on your favorite OS - It's your choice! Supports Windows and Linux Multi-language - Software in English and German

Wodbus USS SSI 3964(R)...

Product Features

The unerring eye into your transmission

- OS independent real time resolution of 10 nanoseconds
- Detects invalid signal levels (tri-state)
- Shows the correct time relationship between all lines
- Display the real logic changes of the TxD and RxD signals
- Handles logic signal inputs with a trigger level of 1.3V
- Integrated HW support for synchronous RS232 transmissions (SSI)

Prepared for the unexpected

- Supports any bit rate from 1 Bps to 1 MBps
- 200 MHz sample rate ensures most accurate signal and data display
- Automatic recognition of bit rate, databits and parity
- Supports protocols with 9 data bits
- Detects breaks, framing and parity errors
- Individual protocols through highly adaptable templates and Lua

Autonomous USB device

- Independent analyzer box, controlled and sourced via USB
- Fast real-time signal/data processing realized in hardware
- Easily adaptable to various bus systems
- Data and telegram recording with direction recognition
- Automatic time stamping and event generation
- Integrated smart LED tester showing connection errors & signal states
- Data transfer to unlimited PC storage facilities
- Updatable firmware for future-proof extensions or improvements

IFTOOLS GmbH Matterhornstrasse 17 65199 Wiesbaden Germany

www.iftools.com Email: info@iftools.com Fon: +49 611 7237389-3/4 We ♥ quality □ Made In □ Germay Technical changes and improvements can be done without prior notice Page 1 January 2024

Analyzer software with unique MultiView concept

When analyzing data transmissions you often have to switch between several protocol lavers, in case you want to see the raw data bytes behind a telegram or the logical tri-state signal forming the bytes. More than that: You need to compare different transmission parts, check

the time response amid telegrams and simultaneously look for the signal relationship.

The answer - The MultiView concept

The analyzer program uses a multi-process architecture for stability and scalability. It draws a clear distinction between the recording and realtime displaying of data or signal. Even though working like a single application, the software delegates the display part to modular 'View' programs, each one specially designed and highly adapted to its protocol layer or particular task. There are Views for:

- Showing the telegram (ProtocolView)
- Displaying the raw (also 9-bit) data (DataView)
- Plotting the logical tri-state signal (SignalView)
- Monitoring all occurring events (EventView)

Already while recording you can open as many Views as you want and assign them to certain parts of the current or stored recording.

Views can act independently or in step with others. They update their content depending on the selection/click in another view, or just display the currently received data.

You can resize and place them on your screen(s) according to your wishes without disturbing an active recording!

Easv to use

Every View follows the concept to offer a specially optimized display and toolbox for its kind of examination. Gone are multiline toolbars and overloaded menues.

The ProtocolView stands on top of the transmission layers and splits the data stream into a given protocol like ModBus, USS, 3964(R) etc. You can modify the protocol specifications in real-time to check how the telegrams come up with different settings.

The **DataView** shows you the received data including direction, framing, parity or break informations. It scores with a powerful search engine using regular expressions to find special sequences or data patterns.

The SignalView acts like a digital scope. You can zoom in and out, measure signal distances with two cursors, validate data frames and parities, check the logical signal levels or look for a jittering baud rate.

The EventView isn't associated with a protocol layer and monitoring all kind of events. It is especially useful when you are searching for level changes, a given level duration, bus states or bus errors.

Various export and data sharing abilities

let you further evaluate all recorded data in spreadsheet programs like Excel® or Calc® and make it easy to document the analysis results in most text processing applications.

IFTOOLS GmbH Matterhornstrasse 17 65199 Wiesbaden Germany

www.iftools.com Email: info@iftools.com Fon: +49 611 7237389-3/4

We 🎔	quality
⊡Mac	le
In	
Ger	mav

Concurrent analysis from physical tri-state to top-most levels.

OSI Layer

support

0

MSR05001 Data M

0000-0110

000.0150 00

in Lua scrint lidle

MSB05001 Signal Monitor - leaflet

Edit View Setting

000:0100 OC 16 10 C0 01

MSB-RS232-PLUS S/N MSB05001 - leaflet Record View Settings Options Help

Easy recording

MSB05001 Protocol Monitor - leaflet

New/Edit
 Operation

DIR Start Len Len2 Start2 PRM FCB FCV Port B 68 242 242 68 1.0 0.0 1.0

 Start
 PRM
 Constraint

 10
 1.0
 0.0
 0.0

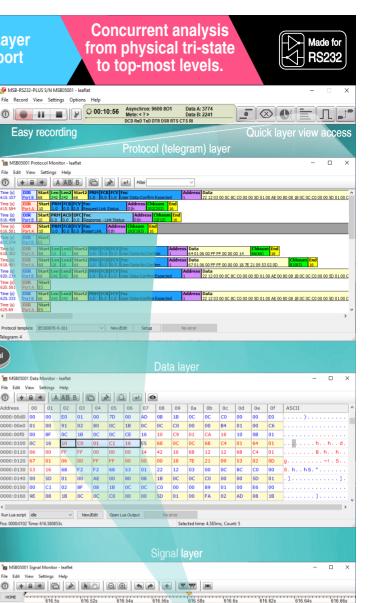
 Start
 PRM
 ACD
 DFC
 Fm

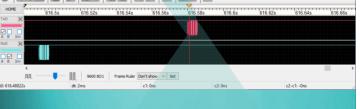
 10
 0.0
 0.0
 0.0
 Re

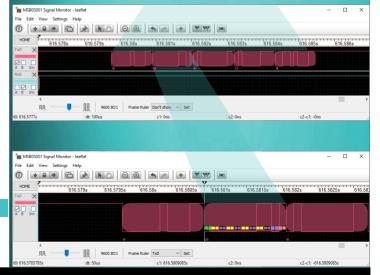
 10
 0.0
 0.0
 0.0
 Re

 10
 0.0
 0.0
 0.0
 Re

 10
 0.0
 0.0
 0.0
 Re







Technical changes and improvements can be done without prior notice Page 2

January 2024



Tir

Protocol display extremly adaptable by integrated Lua 5.3



Lua powered protocol templates

Equipped with a powerful Lua script engine the protocol view is able to parse all thinkable RS232 based protocols. And together with an unique box model it allows the user to format and summarize the telegram output in an extremely wide range.

Handles all telegram definitions

The Protocol View handles the splitting of the continuous data stream into single telegrams by rules and functions, defined in the script language Lua. This makes the Protocol View widely adaptable also to very special and uncommon or even proprietary protocols.

It even allows you to write your own protocol setup and filter dialogs.

Telegram definitions with an unusual EOS, a telegram with a specified length encoded as a byte at a certain position or an idle pause between every telegram like Modbus RTU or USS. Thanks to the integrated Lua all this is handled in a few lines.

What's more: You can play around with the protocol definition in a well equipped editor and see your modifications immediately in the display and without any affection to the recorded data.

Individual telegram output

Every telegram should be shown with certain informations: date and time, address (bus participant), function code, data (in various formats), checksum, telegram delimiter and other things which will become needful when you have to interpret or analyze a communication.

A special box output model combined with Lua makes it very easy to adapt the telegram data to your needs or write your very own telegram decoder.

Real-time display of own checksums

Don't worry about individual checksum validations! With the integrated Lua you can calculate any checksum according to your specifications and display the result right on every telegram.

Comes with many ready-to-use protocol templates

3964R	MDB/ICP	SAE-J1922
BACNet	Modbus ASCII/RTU	SMA-Net
Allen Bradley DF-1	MOVILINK	SSI
DNP3	NMEA	Smith Meter
Executive	P-NET	SRecord
IEC60870-5-101/3	Profibus	USS
Lin BUS (tbd)	SAE-J1587	9 bit protocols

and many others more general templates (STX/ETX, CR/LF, Break,...). All templates are written in Lua themselves and can be easily adapted or extended for your own purposes.

CSV and HTML telegram export

The analyzer program is well equipped for most analyzing intentions. Nevertheless there are situations when you have a need for processing the recorded data - here the telegrams - with additional tools or external applications. Therefore you can export them as:

HTML for documentation

Simple insertion of choosen telegrams in text processing applications.

CSV for spreadsheat applications

Processing of telegram informations e.g. in Excel for statistical analysis.

IFTOOLS GmbH Matterhornstrasse 17 65199 Wiesbaden Germany www.iftools.com Email: info@iftools.com Fon: +49 611 7237389-3/4 We ♥ quality ☐ Made In Germay sheet1/1

	001 Prot	ocol Mer	nitor - N	Andhur	RTU-FOILP	Jupley						-		×
MSB050				vioabus-	KIU-FUII-D	rupiex						_		~
- L-	View	- 10		1										
<u> </u>		•	A AB	В			Filter		~					
No Time 131 219 0		Transn Respons	nission	Device	Function 04 Read In	out Register	Count F	Reg:1 Reg:2 65307	Cks D329					^
No Time	e	Transn	nission	Device	Function		Addre	ss Quantity	Cks					
132 219.9	9985s	Query =	>	3	03 Read Ho	olding Register	40026	37	3454	4 10	.cla	in		I Own
No Time 133 220.0		Transn Respons		3 3	Function 03 Read Ho	olding Register			g:2 Reg:3 Reg 57759 5775					-9:9
No Time 134 220.0		Ouery =	ission	Device	Function	ultiple Registers			ity Byte Count					
No Time	e					Code Desc		3 Cks			J 12	JU 1033	2	
135 220.	1047s	Respons	:e <=	3	16	4 SLAVE	DEVICE FA	ILURE 03E	c.					
No Time 136 220.1	e 1279s	Query =	>	2	Function 03 Read Ho	olding Register	Addre 42000	Modbus	Setup and T	eleara	m Filte	r		
No Time	e		nission	Device	Function	olding Register	Count		-p 2000 1	-9.0				
137 220.1 No Time					Function	olang Register	4 Addre	Setup						~
138 220.3	2486s	Query =	>	2	03 Read Ho	olding Register	42100	Jetup						-
No Time 139 220.3	e 354s	Transn Respons	nission	Device 2	Function 03 Read Ho	olding Register	Count 16		_					
No Time 140 220.3	e			Device	Function	the number	Addre	Modbus	Туре	04	SCII (RTU		
140 220.3 No Time		Query =			Function	olding Register	42109 Count							
141 220.4	4751s	Respons				olding Register	2	Interfra	me Delay/s	0.00	182292			
No Time 142 220.4		Transn Query =		Device 2	Function 03 Read Ho	ding Register	Addre 42111			_				
No Time	e	Transn	nission	Device	Function		Addre	Respons	se Timeout/s	0.02				
143 223.5 No Time		Query =			03 Read Ho Function	olding Register	42111 Count							
144 223.6	6096s	Respons	æ<=	2	03 Read Ho	olding Register	8	Bus Mas	ter Chn.	•	ute	OData	• •	ats P
No Time 145 223.6	e 6225s	Transm Query =	nission :>	Device 2	Function 03 Read Ho	olding Register	Addre 42117				uto	Uata		did D
No Time	e	Transn	nission	Device	Function		Count							
146 223.3 No Time		Respons			03 Read Ho Function	olding Register	12 Addre	Modicon	convention r display)	ON		Yes		
147 223.3	753s	Query =				olding Register	43260	(Registe	a uispidy)	0	- 3			
¢														
Protocol tem	olate M	lodbus			~	New/Edit	Setup	Colour fi	unctions	ON	•	Yes		
		-stands				. Actinguit	octup			0	9			
elegram: 14	£								D	tore d	-	there		
									Kes	core de	efault se	.curiyS		
								Clos	se A	pply				
											_			
)	Ad													
	AU	ldr	Fun	ction			Qu	antitv	Startad	dr	Cks	OK _		
16/60		ldr				Rogicto		antity I	Startad	dr	Cks			
0646s	1		Rea	dHo	lding I	Registe \ddr", te	114	Ļ	0	dr	Cks EFC			
0646s box.t	1	pox.te	Rea ext{	dHo capti	lding I ion="A		xt=tele	l egram:I	0	dr				
	1	box.te capti	Rea ext{ on=' a sin	dHol capti 'Timo nple	lding ion="A e", tex checks	ddr", te t=telegra sum alg	r 114 xt=tele am:tim orithm	egram:I ne() }	0	dr				
	1	box.te capti	Rea ext{ on=' a sin	dHol capti 'Time nple on ch	lding f ion="A e", tex checks iecksu	ddr", te t=telegra	r 114 xt=tele am:tim orithm	egram:I ne() }	0	dr				
	1	box.te capti	Rea ext{ on=' a sin	dHol capti 'Time nple on ch	lding ion="A e", tex checks	ddr", te t=telegra sum alg	r 114 xt=tele am:tim orithm	egram:I ne() }	0	dr				
	1	box.te capti	Rea ext{ on=' a sin nctic	dHol capti 'Time nple on ch al su	lding f ion="A e", tex checksu m = 0	Addr", te t=telegra sum alg im(from	r 114 xt=tele am:tin orithm , to)	egram:I	0	dr				
	1	box.te capti	Rea ext{ on=' a sin nctic loca	dHol capti 'Time nple on ch al sui dd al	lding ion="A e", tex checksu m = 0 I teleg	Addr", te t=telegra sum alg m(from ram byte	r 114 xt=tele am:tin orithm , to)	egram:I	0	dr				
	1	box.te capti	Rea ext{ on=' a sin nctio loca a for	dHol capti 'Time on ch al sui dd al i=fro	lding I ion="A e", tex checks m = 0 I teleg m,to d	Addr", te t=telegra sum alg im(from ram byte o	r 114 xt=tele am:tim orithm , to) es fror	egram:I	0	dr				
	1	box.te capti	Rea ext{ on=' a sin nctic loca a for	dHol capti 'Time nple on ch al su dd al i=fro sum :	lding I ion="A e", tex checks m = 0 I teleg m,to d	Addr", te t=telegra sum alg m(from ram byte	r 114 xt=tele am:tim orithm , to) es fror	egram:I	0	dr				
	1	box.te capti	Rea ext{ on=' a sin nctic loca a for s end	dHol capti 'Time nple on ch al su dd al i=fro sum :	ding f ion="A e", tex checksu m = 0 I teleg m,to d = sum	Addr", te t=telegra sum alg m(from ram byte o + tg.dat	xt=tele am:tin orithm , to) es fror a(i)	egram:Ine() }	0	dr				
	1	box.te capti	Rea ext{ on=' a sin nctic loca a for s end w	dHol capti 'Time on ch al sui dd al i=fro sum : e on	iding I ion="A e", tex checksu m = 0 I teleg m,to d = sum ly need	Addr", te t=telegra sum alg m(from ram byte o + tg.dat d a 8 bit	xt=tele am:tin orithm , to) es fror a(i)	egram:Ine() }	0	dr				
	1	capti 	Rea ext{ on=' a sinn ctic loca for s end w retu	dHol capti 'Time on ch al sui dd al i=fro sum : e on	ding f ion="A e", tex checksu m = 0 I teleg m,to d = sum	Addr", te t=telegra sum alg m(from ram byte o + tg.dat d a 8 bit	xt=tele am:tin orithm , to) es fror a(i)	egram:Ine() }	0	dr				
	1	box.te capti	Rea ext{ on=' a sinn ctic loca for s end w retu	dHol capti 'Time on ch al sui dd al i=fro sum : e on	iding I ion="A e", tex checksu m = 0 I teleg m,to d = sum ly need	Addr", te t=telegra sum alg m(from ram byte o + tg.dat d a 8 bit	xt=tele am:tin orithm , to) es fror a(i)	egram:Ine() }	0	dr				
	1	capti 	Rea ext{ on=' a sinn ctic loca for s end w retu	dHol capti 'Time on ch al sui dd al i=fro sum : e on	iding I ion="A e", tex checksu m = 0 I teleg m,to d = sum ly need	Addr", te t=telegra sum alg m(from ram byte o + tg.dat d a 8 bit	xt=tele am:tin orithm , to) es fror a(i)	egram:Ine() }	0	dr				
	1	capti 	Rea ext{ on=' a sinn ctic loca for s end w retu	dHol capti 'Time on ch al sui dd al i=fro sum : e on	iding I ion="A e", tex checksu m = 0 I teleg m,to d = sum ly need	Addr", te t=telegra sum alg m(from ram byte o + tg.dat d a 8 bit	xt=tele am:tin orithm , to) es fror a(i)	egram:Ine() }	0	dr				
	1	capti 	Rea ext{ on=' a sinn ctic loca for s end w retu	dHol capti 'Time on ch al sui dd al i=fro sum : e on	iding I ion="A e", tex checksu m = 0 I teleg m,to d = sum ly need	Addr", te t=telegra sum alg m(from ram byte o + tg.dat d a 8 bit	xt=tele am:tin orithm , to) es fror a(i)	egram:Ine() }	0	dr				
	1	capti 	Rea ext{ on=' a sinn ctic loca for s end w retu	dHol capti 'Time on ch al sui dd al i=fro sum : e on	iding I ion="A e", tex checksu m = 0 I teleg m,to d = sum ly need	Addr", te t=telegra sum alg m(from ram byte o + tg.dat d a 8 bit	xt=tele am:tin orithm , to) es fror a(i)	egram:Ine() }	0	dr				
	1	capti 	Rea ext{ on=' a sinn ctic loca for s end w retu	dHol capti 'Time on ch al sui dd al i=fro sum : e on	iding I ion="A e", tex checksu m = 0 I teleg m,to d = sum ly need	Addr", te t=telegra sum alg m(from ram byte o + tg.dat d a 8 bit	xt=tele am:tin orithm , to) es fror a(i)	egram:Ine() }	0	dr				
	1	capti 	Rea ext{ on=' a sinn ctic loca for s end w retu	dHol capti 'Timo pn ch al sun dd al =fro sum : e onl rrn su	Iding I ion="A e", tex ecksu m = 0 I teleg m,to d = sum ly need um % :	vddr", te t=telegra sum alg m(from ram byte o tram byte o + tg.dat d a 8 bit 256	xt=tele am:tin orithm , to) es fror a(i)	egram:Ine() }	0	dr				
	1	capti 	Rea ext{ on=' a sinn ctic loca for s end w retu	dHol captii 'Timo pn ch al sun dd al i=fro sum : e onl irm su	Iding I ion="A p", tex check: ecksu m = 0 I teleg m,to d = sum % : sum % :	kddr", te t=telegra sum alg m(from ram byte o + tg.dat d a 8 bit 256	r 114 xt=tele am:tin orithm , to) es fror a(i) check	egram:Ine() }	0	dr				
	1	capti 	Rea ext{ on=' a sinn ctic loca for s end w retu	dHol captii 'Timu nple on ch al suu dd al i=fro sum : e oni irrn su	ding I ion="A" check: check: m = 0 teleg m,to d = sum ty need sum % :	kddr", te t=telegr sum alg m(from ram byte 0 + tg.dat d a 8 bit 256	r 114 xt=tele am:tin orithm, to) es fror a(i) check	egram:Ine() } nto sum	0 Data(2) }		EFC			
	1	capti 	Rea ext{ on=' a sinn ctic loca for s end w retu	dHol capti 'Time nple on ch al suu dd al sum : e oni rm si e oni rm si	Iding I ion="A checksu ecksu m = 0 I teleg m,to d = sum ly nee: y nee: nee: nee: nee: nee: nee: nee: nee:	tetelegri sum alg m(from ram byte o + tg.dat d a 8 bit 256	r 114 xt=tele am:tin orithm , to) es fror a(i) check	egram:I ne() } nto ssum	0 Data(2) }	@ - 1	EFC) <mark>*</mark>	Find Text
	1	capti 	Rea ext{ on=' a sinn ctic loca for s end w retu	dHol capti 'Time nple on ch al suu dd al sum : e oni rm si Ele gai Ele gai	ding I ion="A"	Addr", te t=telegra sum alg m(from ram byta o + tg.dat d a 8 bit 256	r 114 xt=tele am:tin orithm to) es fror a(i) check	eggram: he() } nto ssum	0 Data(2) }		EF() . : .	Find Text
	1	capti 	Rea ext{ on=' a sinn ctic loca for s end w retu	dHo capti 'Time on ch al sun dd al =fro sum : e oni irm su	ding I ion="A"	Addr", te t=telegra sum alg m(from ram byta o + tg.dat d a 8 bit 256	r 114 xt=tele am:tin orithm to) es fror a(i) check	eggram: he() } nto ssum	0 Data(2) }		EF() . : .	Find Text
	1	capti 	Rea ext{ on=' a sinn ctic loca for s end w retu	dHol capti 'Time nple on ch al suu dd al sum : e oni rm si Ele gai Ele gai	ding I ion="A"	Addr", te t=telegri sum alg m(from ram byte 0 + tg.dat d a 8 bit 256	r 114 xt=tele am:tin orithm (i 0) es fror a(i) check 2	eggram: he() } nto ssum	0 Data(2) }		EF() . : .	Find Test
box.t	1 Iteext{	oox.tu capti 	Rea ext{ on=' a sinnctio loca a for end w retu	dHo captii 'Timo nple on ch al sun dd al sum : e oni sum : e oni irrn su	ding I ion="A"	Addr", te t=telegra sum alg m(from ram byte + tg.dat d a 8 bit 256	r 114 xt=tele am:tin orithm (to) es fror a(i) check	eggram: he() } nto ssum	0 Data(2) }		EF() . : .	Find Text
box.t	1 Iteext{	enryvods	Rea ext{ on=' a sin nctic loca a for s w retu	dHo capti 'Time on ch al sun dd al =fro sum : e oni irm su	ding I ion="A"	Addr", te t=telegra sum alg m(from ram byte o + tg.dat d a 8 bit 256	r 114 xt=tele am:tin orithm t to) es fror a(i) check	eggram: he() } nto ssum	0 Data(2) }		EF() . : .	Find Test
box.tt	1 H H H H H H H H H H H H H	ency.ods	Rea ext{ on=' a sin for for end retuind	dHol captii 'Timu nple on ch al suu dd al =fro sum : e onl irm su e onl irm su	ding I ion="A"	Addr", te t=telegra sum alg m(from ram byte o + tg.dat d a 8 bit 256	r 114 xxt=tele am:tin orithm , to) es fror a(i) check y & \$\$ (i) check	eggram: he() } nto ssum	0 Data(2) }		EF() . : .	Find Text
box.tt	1 H H H H H H H H H H H H H	ency.ods	Rea ext{ on=' a sin nctic loca a for s w retu	dHol captii 'Time nple on ch al sui dd al sui dd al sui e onl i=fro sum : e onl i=fro sum : E on the E on the E on the E on the the E on the the the the the the the the the the	ding I ion="A"	tetelegra sum alg m(from ram byte o + tg.dat d a 8 bit 256	r 114 xt=telele am:tin orithm , to) ces from a(i) check a 100 10 check a 100 100 check a 100 100 chec	eggram: he() } nto ssum	0 Data(2) }		EF() . : .	Find Text
box.t	1 H H H H H H H H H H H H H	envyeds	Rea ext{ on=' a sin for for end retuind	dHol captii 'Time on ch al su dd al =fro sum : E e on rm si	ding I ion="A"	Addr", te t=telegra sum alg m(from ram byte 0 + tg.dat da 8 bit 256	Interference Interference am:tin orithm orithm to) es fror a(i) check s i 9 \$\$	eggram: he() } nto ssum	0 Data(2) }		EF() . : .	Find Text
	1 teext{	encysds	Rea ext{ on=' a sinnetic loca an for end w retu	dHol captii 'Time nple on ch al sui dd al sui dd al sui e onl i=fro sum : e onl i=fro sum : E on the E on the E on the E on the the E on the the the the the the the the the the	ding I ion="A"	Addr", te t=telegra sum alg m(from ram byte 0 + tg.dat d a 8 bit 256	<pre>114 xt=tele am:tin orithm to) as from a(i) check check</pre>	eggram: he() } nto ssum	0 Data(2) }		EF() . : .	Find Text
box.t	1 treext{	eny.eds	Rea ext{ on=' a sinnetic loca an for end w retu	dHol captii 'Time on chal suu dd al =fro sum : e onl rm su e onl	ding I ion="A"	teter teter sum alg m(from ram byte o + tg.dat da 8 bit 256 fice Writer tr Fermer a fais 1 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1	r 114 xxt=tele am:tin orithm r orithm r a(i) check	eggram: he() } nto ssum	0 Data(2) }		EF() . : .	Find Text
	1 teext{	ency.ods	Rea ext{ on=' a sinn for end w retuind	dHol captii 'Time on chail sund dd al =fro sum : e oni rrn su	ding I ion="A"	Addr", te t=telegra sum alg m(from ram byte o + tg.dat da 8 bit 256	Interference Interference am:tin orithm orithm to) es from a(i) checks y y y a(i) checks y y	eggram: he() } nto ssum	0 Data(2) }		EF() . : .	Find Text
	1 treext{	envy.ods	Rea ext{ on=' a sinn for end w retuind	dHol captii 'Time on chal suu dd al =fro sum : e onl rm su e onl	ding I ion="A"	teter teter sum alg m(from ram byte o + tg.dat da 8 bit 256 fice Writer tr Fermer a fais 1 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1	Interference Interference am:tin orithm orithm to) es from a(i) checks y y y a(i) checks y y	eggram: he() } nto ssum	0 Data(2) }		EF() . : .	Find Text
	1 treext{	ency.ods	Rea ext{ on=' a sinn for end w retuind	dHol capti 'Time on ch al sun dd al =fro sum : e on i tro su	Iding I ion="A e", tex check ecksu m = 0 i teleg m,to d = sum l teleg m,to d = sum % : '''''''''''''''''''''''''''''''''''	Addr", te t=telegra sum alg m(from ram byte o + tg.dat da 8 bit 256	Intermediate xxt=tele am:tin orithm to) es from a(i) check y y y y a y y y a y y a b a a b </td <td></td> <td>O Data(2) }</td> <td></td> <td></td> <td></td> <td>A CAPD</td> <td>Find Test</td>		O Data(2) }				A CAPD	Find Test
	1 treext{	envy.ods	Rea ext{ on=' a sinn for end w retuind	dHol captii 'Time on chail sund dd al =fro sum : e oni rrn su	Iding I ion="A e", tex check ecksu m = 0 i teleg m,to d = sum l teleg m,to d = sum % : '''''''''''''''''''''''''''''''''''	Addr", te t=telegra sum alg m(from ram byte o + tg.dat da 8 bit 256	Intermediate xxt=tele am:tin orithm to) es from a(i) check y y y y a y y y a y y a b a a b </td <td>eggram: he() } nto ssum</td> <td>0 Data(2) }</td> <td></td> <td>EF(</td> <td></td> <td>) . : .</td> <td>Find Text</td>	eggram: he() } nto ssum	0 Data(2) }		EF() . : .	Find Text
		enveds	Rea ext{ on=' a sinn for end w retuind	dHol capti 'Time on ch al sun dd al =fro sum : e on i tro su	Iding I ion="A e", tex check ecksu m = 0 i teleg m,to d = sum l teleg m,to d = sum % : '''''''''''''''''''''''''''''''''''	Addr", te t=telegra sum alg m(from ram byte o + tg.dat da 8 bit 256	Intermediate xxt=tele am:tin orithm to) es from a(i) check y y y y a y y y a y y a b a a b </td <td></td> <td>O Data(2) }</td> <td></td> <td></td> <td></td> <td>A CAPD</td> <td>Find Test</td>		O Data(2) }				A CAPD	Find Test
		enveds	Rea ext{ on=' a sinn for end w retuind	dHol capti 'Time on ch al sun dd al =fro sum : e on i tro su	Iding I ion="A e", tex check ecksu m = 0 i teleg m,to d = sum l teleg m,to d = sum % : '''''''''''''''''''''''''''''''''''	Addr", te t=telegra sum alg m(from ram byte o + tg.dat da 8 bit 256	Intermediate xxt=tele am:tin orithm to) es from a(i) check y y y y a y y y a y y a b a a b </td <td></td> <td>O Data(2) }</td> <td></td> <td></td> <td></td> <td>A CAPD</td> <td>Find Test</td>		O Data(2) }				A CAPD	Find Test
	1 treext{	enveds	Rea ext{ on=' a sinn for end w retuind	dHol capti 'Time on ch al sun dd al =fro sum : e on i tro su	Iding I ion="A e", tex check ecksu m = 0 i teleg m,to d = sum l teleg m,to d = sum % : '''''''''''''''''''''''''''''''''''	Addr", te t=telegra sum alg m(from ram byte o + tg.dat da 8 bit 256	Intermediate xxt=tele am:tin orithm to) es from a(i) check y y y y a y y y a y y a b a a b </td <td></td> <td>O Data(2) }</td> <td></td> <td></td> <td></td> <td>A CAPD</td> <td>Find Test</td>		O Data(2) }				A CAPD	Find Test
		envesting For a function of the function of t	Rea ext{ on=' a sinn for end w retuind	dHol capti 'Time on ch al sun dd al =fro sum : e on i tro su	Iding I ion="A e", tex check ecksu m = 0 i teleg m,to d = sum l teleg m,to d = sum % : '''''''''''''''''''''''''''''''''''	Addr", te t=telegra sum alg m(from ram byte o + tg.dat da 8 bit 256	Intermediate xxt=tele am:tin orithm to) es from a(i) check y y y y a y y y a y y a b a a b </td <td></td> <td></td> <td></td> <td></td> <td></td> <td>A CAPD</td> <td>Find Test</td>						A CAPD	Find Test
		ency.eds	Rea ext{ on=' a sinn for end w retuind	dHol capti 'Time on ch al sun dd al =fro sum : e on i tro su	Iding I ion="A e", tex check ecksu m = 0 i teleg m,to d = sum l teleg m,to d = sum % : '''''''''''''''''''''''''''''''''''	Addr", te t=telegra sum alg m(from ram byte o + tg.dat da 8 bit 256	Intermediate xxt=tele am:tin orithm to) es from a(i) check y y y y a y y y a y y a b a a b </td <td></td> <td></td> <td></td> <td></td> <td></td> <td>A CAPD</td> <td>Find Test</td>						A CAPD	Find Test
		envesting For a function of the function of t	Rea ext{ on=' a sinn for end w retuind	dHol capti 'Time on ch al sun dd al =fro sum : e on i tro su	Iding I ion="A e", tex check ecksu m = 0 i teleg m,to d = sum l teleg m,to d = sum % : '''''''''''''''''''''''''''''''''''	Addr", te t=telegra sum alg m(from ram byte o + tg.dat da 8 bit 256	Intermediate xxt=tele am:tin orithm to) es from a(i) check y y y y a y y y a y y a b a a b </td <td></td> <td></td> <td></td> <td></td> <td></td> <td>A CAPD</td> <td>Find Test</td>						A CAPD	Find Test
box.t		envesting For a function of the function of t	Rea ext{ on=' a sinn for end w retuind	dHol capti 'Time on ch al sun dd al =fro sum : e on i tro su	Iding I ion="A e", tex check ecksu m = 0 i teleg m,to d = sum l teleg m,to d = sum % : '''''''''''''''''''''''''''''''''''	Addr", te t=telegra sum alg m(from ram byte o + tg.dat da 8 bit 256	Intermediate xxt=tele am:tin orithm to) es from a(i) check y y y y a y y y a y y a b a a b </td <td></td> <td>0 Data(2) } • • • • • • • • • • • • • • • • • • •</td> <td></td> <td></td> <td></td> <td>A CAPD</td> <td>Find Test</td>		0 Data(2) } • • • • • • • • • • • • • • • • • • •				A CAPD	Find Test
box.t		envesting For a function of the function of t	Rea ext{ on=' a sinn nctic loca for s end 	dHol capti 'Time on ch al sun dd al =fro sum : e on i tro su	Iding I ion="A p=", tex checksu m = 0 m,to d = sum ly neer um % : i @ W i @ Q i @ Q	Addr", te t=telegra sum alg m(from ram byte o + tg.dat d a 8 bit 256	Intermediate xxt=tele am:tin orithm to) es from a(i) check y y y y a y y y a y y a b a a b </td <td></td> <td>0 Data(2) } • • • • • • • • • • • • • • • • • • •</td> <td></td> <td></td> <td></td> <td>A CAPD</td> <td>Find Test</td>		0 Data(2) } • • • • • • • • • • • • • • • • • • •				A CAPD	Find Test
box.t		ency.ods	Rea ext{ on=' a sin nctic loca for s end 	dHol captii 'Time nple on chu al suu e oni i=fro sum : e oni i=fro sum : Page 1/	Iding I ion="A p", tex check: ecksu m = 0 m, to d = sum ly nee: um % : i @e w i @e w i @e w i @e i @e i i @e i @e i i @ i @e i i @ i @e i i @ i @ i @ i @ i @ i @ i @ i @ i @ i	Addr", te t=telegra sum alg m(from ram byte o + tg.dat d a 8 bit 256	Interference Interference am:tim am:tim orithm am:tim orithm am:tim orithm am:tim a(i) check a(i)		0 Data(2) }				A CAPD	Find Test

Technical changes and improvements can be done without prior notice Page 3

STD

January 2024



Direct sampling and serial decoding

Precise measurement of the time relationship between data and signals is the backbone for serious protocol analysis. The direct sampling of all lines with up to 200 MSamples together with an independent decoding of the serial data, performed by two special in-house developed USART, provides:

10 nanosec precise and PC independent data timestamps

Essential when analyzing protocols with strict time specifications like BACnet, ccTalk, Modbus RTU, USS and others.

Display of all logical signal levels with 10 nanosec resolution 200 MHz sample rate ensures most accurate display of the signal and timing even with highest bit rates. This includes also the recognition of open lines, stand-by conditions of the data drivers or short circuits.

Support of any bit rate from 1 Bps to 1 MBps

Don't worry about unusual bit rates in individual applications.

Correct detection of the break conditions

which is an issue for reset or high level protocols using a real break as a telegram frame delimiter (e.g. Sync Break in a LIN or DMX512 Bus) or for packet synchronization.

Integrated HW support for synchronous RS232 transmissions

like SSI, no need for different tools, one device, countless use cases.

Data format scanner and 9-bit data word support

By providing its own decoding hardware for the serial transmission data the analyzer is not only able to work with uncommon bit rates but also supports 9 bit data words and comes with an unique data format scanner (bit rate, data bits, parity, synchronous SSI parameters).

Detect unknown data formats (bit rate, data bits, parity)

Simply insert the analyzer in an active connection and it detects the right (a)synchronous data format and bit rate settings for you.

Decoding and analyzing protocols with 9-bit data

Certain bus protocols are using the parity bit as a 9th bit to differ between an address and the user data (MDB/ICP, P-NET). With the 9N1 setting you are able to turn the analyzer in a pure 9 bit recorder with full 9-bit data display and protocol support.

Synchronous recording with two or more analyzers

To compare two records the data have to be in a precise time relationship. Otherwise you cannot decide about the chronological sequence or check the synchronicity of certain events.

Synchronizing of multiple analyzers is simple

Just connect them with a standard network 1:1 cable and all analyzers share the same time base accurace of 10 nanoseconds.

Parallel recording of different connections

Examine independent transmissions with unequal data formats, e.g. IN and OUT of a protocol converter or different bus segments. are connected to the same or different computers.

Central record control by Master/Slave principle

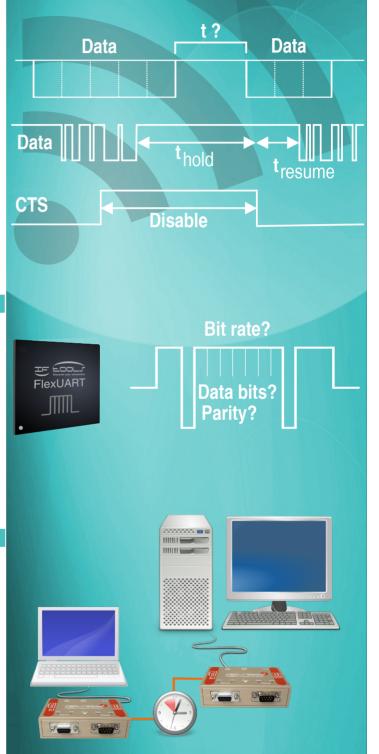
One analyzer controls the recording. It doesn't matter if the analyzers are connected to the same or different computers.

IFTOOLS GmbH Matterhornstrasse 17 65199 Wiesbaden Germany

www.iftools.com Email: info@iftools.com Fon: +49 611 7237389-3/4



"Direct sampling of all lines with 10nsec precision and two in-house developed USARTs ensure that you won't miss anything"



10 Nanoseconds synchron accuracy

Technical changes and improvements can be done without prior notice Page 4 January 2024

The Switch Option - a virtual Breakout Box

The MSB-RS232-PLUS analyzer not only records all data and signals with 10 nanoseconds precision. Thanks to an integrated switch option it also let you specifically influence lines and data for testing purpose.

See how your application reacts when you switch a control line, interrupt the transmission or when you feed it with a wrong parity or a frame error.

How does it work?

Imagine the switch option as a kind of patch panel between the inputs and outputs of port A and B (similar like a breakout box). On this you can place switches, inverters and further elements and wire them to the input or output pins of both ports.

But with the difference, that you do not use real wires and switches. You only place and connect them virtually on a drawing area.

Advanced features

Additional to switch and reroute lines there are a lot of benefits, which a normal breakout box cannot provide.

Line inverting

Inverter elements allow the negation of each data or signal line.

Simulate Transmission errors and breaks

Send data together with a framing or parity error.

Data injection

Input any desired data sequences, also binaries.

Data probe of any line (UART assignment)

Free selection of lines, which are interpreted as data signal.

What's the need for?

Reroute lines and build your own adapter when uncommen lines are used. Invert signals when necessary.

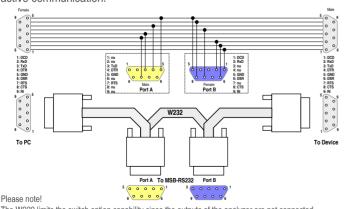
Simulate/test data flow by active switching control lines. Testing the reaction on e.g. rts/cts protocols is now made easy.

Intercept and manipulate data to simulate special protocol and/or error sequences.

W232 Cable

Prepare your connection for non-interruptable analysis.

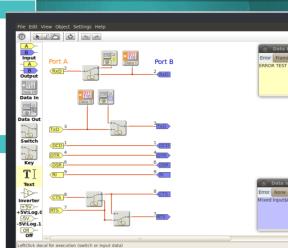
Plug-in the analyzer at any time and without breaking or disturbing an active communication!



The W232 limits the switch option capability since the outputs of the analyzer are not connected.

IFTOOLS GmbH Matterhornstrasse 17 65199 Wiesbaden Germany www.iftools.com Email: info@iftools.com Fon: +49 611 7237389-3/4

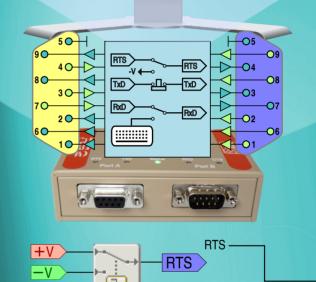


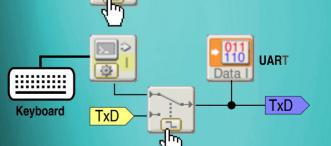


Optional Extensions

-1

ata II







Technical changes and improvements can be done without prior notice Page 5 January 2024





Υ

Download this leaflet



Measurement					
Bus Systems	Recording and analysis of asynchronous (all) and synchronous (SSI) bus systems based on standard RS232 and logic signals.			100mm	
Bit rates	High-precise setting and measuring of any bit rate from 1 Bps to 1 MBps with 0,1% precision .		•	PC MSB	
Asynchronous bus	59 Data bits, none, even, odd, mark, space parity, any stop bits.			USB Link	CE
Synchronous bus Logical Linestates	SSI with 563 Data bits, 11 MBps.				
Logical Linesiales	Detection of the logical states (A-B): 1 (V+), 0 (V-), invalid (-1.3V < In < +1.3V).				
Time resolution	10 nanoseconds (hardware based) for all data and line change events (sample rate up to 200 MHz). Several analyzers synchronizable.	60mm		Fieldbus Protocol A	nalyzer
Connectors				H)	
Signal levels	Standard RS232 Levels $\pm 3V$ to $\pm 15V,$ ESD protected inputs 5kOhm .			RTS TXD Status	RXD CTS
Bus connectors Electrical	Standard D-Sub 9pin connectors, male and female.			Port A	Port B
Logic Mode	Connection between port A and port B through high speed RS232 buffer, each switchable . All RS232 inputs adjustable as logical inputs with trigger level		•		
Ū	1.3V, 5kOhm.				
Device-PC	Power supply (200mA) and data connection via USB 2.0 high speed.			Four multi-coloured LEDs for RS2 working as a smart Led tester. On	
Recording				indicating the device USB and ope	eration mode
Principle	The analyzer marks every event (data byte, signal alternation) with a time stamp in 10 nanosecond resolution (independent of the PC) and sends the information via USB to the connected PC where it is stored in a special file.				
Featuring	Real time analyzing and simultaneous access/display of different recorparts even during an active recording.	rd			
Capacity	4 GByte max. record file size on PC for real time analyzing. Unlimited (resp. limited by the free hard disk space) records when usi the special command line API for long time recordings.	ng		D-Sub 9pin female	D-Sub 9pin male
Record time	The record time depends on the selected kind of events and data rate of the connection.		1		
Protocols		28mm			
Ready to use Write your own	Includes templates for 3964(R), BACNet, DF1, DNP3, IEC60870, MDB/ICP, Modbus, MOVILINK, NMEA, P-Net, Profibus, SAE-J1587, SAE-J1922, Smith Meter, SMA-Net, SSI, USS, 9-Bit and more. Templates are continously extended by free software updates. Equipped with a Lua script engine the Analyzer is able to parse all thinkable RS232 based protocols and format and summarize the telegram output in an extremely wide range.		•	Standard D-Sub 9pin RS232 male	and female
Display LEDs				connectors for simple bus connecti Interactively editable signal linkage	ion.
USB Connection	Multi-coloured LED shows the USB/Analyzer connection and record			(signal switching, inverting, rerouting	ng, see SwitchOption)
USB Connection	state. Also errors caused by overload and low USB voltage.				
RS232 Connection	Four multi-coloured leds act like a smart LED tester and indicate the state of the RS232 connection (correct, broken or inverted) as well as the received/sent data (even single bytes) and the RTS/CTS line state				
Requirements	& Supported OS				
Hardware	A graphic display with minimum 1024 x 768 resolution and 16 Bit colo or better recommended, 200 MByte free hard disk space for installatic	ur			
	additional space for further record files. 256 MByte RAM or more, one free USB 2.0 port.	л,			
Microsoft Windows	Windows XP, Vista, 7, 8, 10, 11, all 32 and 64 bit.				
Linux	All Linux with kernel 2.4.18 or higher, GLIBC 2.4 or higher and installed Gtk2 libs, 32 and 64 bit systems.				
Vhat you get					
Content	MSB-RS232-PLUS analyzer, CDROM with installer for Windows and Linux, USB cable for connection with PC, DS232 cable 2m at 1 Opel Dsub Connectors male to female			RJ45 socket to synchronizes several analyzers with a	USB Type B socket for PC connection
Warranty	RS232 Cable, 2m, 1:1, 9Pol DSub-Connectors male to female Made in Germany, 3 years warranty and free product lifetime updates for firmware and software.	3		resolution of 10 Nanosec	and power supply
Optional Exten					
SwitchOption Price: 99 €	The integrated Switch Editor turns the analyzer in a breakout box and let you switch, reroute, invert lines, inject data and simulate errors jus			Price: 479 € wit	hout tax
W232-Cable Price: 39 €	with a mouse click. See page Options. The special W232 cable allows you to connect and disconnect the analyzer without interrupting the connection lines. See page Options.				
IFTOOLS GmbH Matterhornstrasse	www.iftools.com We ❤ quali 17 Email: info@ittools.com □Made	ty	Tech	nical changes and improvements ca	
65199 Wiesbaden	Email: info@iftools.com				Page
Germany	Fon: +49 611 7237389-3/4 Germay				January 20