MSB Analyzer Concept

Michael Hungershausen, www.iftools.com

January 2020

The following article tries to get to the bottom of the question: 'Why using a hardware interface analyzer when getting low priced software analyzing tools at every corner'. There must be a *hidden* advantage.

We assure you, there are *many obvious* advantages. Features which software solutions can not provide.

General differences

Analysis by Software

Data acquisition

To get access to the transferred data the data signal has to be tapped, decoded and passed through a RS232/485 USB-converter to the PC.
In general a converter can not detect if a correct signal is applied. Only a solution with additional hardware can achieve this task.

RS485

RS485 signals can be directly tapped from the bus and passed to the converter. Only one channel is necessary as send and receive data are transferred on the same bus wires (2-wire, bidirectional).
Since the converter can not detect the inactive bus (undriven, tristate, floating) this important information can not be evaluated and processed. Also the data direction and data source is undetermined.

RS232

RS232 data are transferred on two lines Rxd and Txd. Both data streams have to be tapped and set into a time relationship to detect the sequence of the data.
The tapping is done through 'X' or 'Y' adapters. Both types have in common that they have two jacks for a direct pass-through connection.

The 'X' type passes the signals to two RS232 to USB converters. The data has to be combined from two input channels.
The time relationship between the channels can only be detected in parts, caused by variant processing times inside the PC.

The 'Y' Type uses a diode logic to combine the two serial channels to one which is passed to the converter. This combination should not be used if the data streams are unknown because:
Simultaneous data streams at Rxd and Txd interfere and produce unreadable characters. In addition the receiver does not know if the received data comes from Txd or Rxd. So it is up to the user to sort out the single characters.

(Dis)advantages

The advantage of this software solution is surely the favorable price. Some reasonable free programs exist if the adapter and converter is provided by the user.
Opposed are grave disadvantages:
The time relationship between the single characters can not be easily assigned and only with rough resolution in the milliseconds range. The reason are the different reaction and processing times inside the PC. Time stamps are set not directly when the events occur on the bus but much later, when the data is read from the data buffers.
Also important additional information is missing: Is the bus active or not? Is the data enabling correct? What is the data direction? and more.

Analysis by Hardware

In contrast to pure software solutions the hardware analyzer taps the signals on the bus and routes them to the processing unit. All signals - for RS232 all modem lines - are processed, decoded and all occurring events are directly marked with a time stamp.
A unique special evaluation unit can detect the inactive (tristate) bus condition for every signal. Especially for RS485 this is an important infor-

mation to examine the data direction and the correct bus enabling.
The time stamped data of the recording unit are sent to the PC for further processing, storage and display.
Of course multiple analyzers can be synchronized, so that all data get the same time stamps.

What is the advantage ?

By developing an own evaluation system we are fully independent of the limited and unreliable examination capabilities of the PC hardware. With using modern gate array technology all necessary analysis tools are already available in the analyzer.
The content of the gate array and therewith the functionality of the analyzer can be changed at each time as variable hardware together with the evaluation software. As a result the analyzer can quickly be adapted to the needs of different requirements, simply by using the latest control software.

Read about some advantages in detail.

Discover the real data sequences and timing

How does the software solution work?

When software solutions try to evaluate the timing they depend on the unequal reaction and processing times of the operating system for the interrupt requests (Latency time). Using RS232/485 to USB converter also add the delays of the USB subsystem.
The resulting time stamps are the times of the interrupt processing of the operating system, but not the real times of the occurring events.
Especially if only one combined input is used to examine Rxd and Txd you can not be sure to see the real sequence of the transferred data at Rxd and Txd.
In worst case the sequence of the data is twisted. It is no problem if both devices send the data alternatingly with some time between the data packets.
But as soon as both devices send at the same time mixed and unreadable data information will happen. Even when you use different inputs for recording the time relationship is not clear and correct because of the different delays of the processing software.
You also can not get a time resolution better than in a millisecond range.
Try to get precise information about these reaction times:
How many characters are sent after RTS is set to stop? How fast does a device process the request for data? Overlap data streams because of faulty processing or reaction? Are the pause times correct in time dependent protocols? And many other questions.
A resolution of milliseconds is not sufficient by far. In an 115200bps (bit per second) transmission 10 characters are transferred in one millisecond. In an RS485 system with 10Mbps 1000 Characters are on the line within one millisecond. Impossible to get a precise evaluation from the PC hardware.
All this information can not be provided by pure software solutions. They are based on insufficient and indeterminate PC hardware which is not intended for this set of problems.

What does the hardware analyzer do?

Every signal level change, including the mentioned tristate condition, is marked with sequential time stamps with a resolution of one micro second or even 10ns. Additionally the data on the Rxd and Txd lines are decoded with oversampling and also marked.
All these ‘events’ are sent to the operating program where they are displayed and evaluated in various formats.

Because of stamping the data *before* sending them via USB to the PC there is no dependency on the above mentioned OS scheduler latency times. Also the analyzer has its own memory of 512kB to buffer the irregular operating and answering times of the PC. This avoids data gaps at highest data throughput. The data themselves however are stored directly on the PC.
Now it is easy to answer all the questions above. With totally independent signal capturing and one micro second and less event resolution possible trouble spots of the transmission will quickly be detected.

The following features can only be provided by a hardware analyzer, so we refrain from the comparison with the software solution.

See the logic behind the signals

Having sampled the signals and their level changes in a time raster it is easy to display them in a scope-like form. Even the serial channels Rxd and Txd and their bit sequences are sampled and can be watched in the scope window of the MSB operating program.

So the analyzer can be used as a logic scope with one or 100 MHz resolution, showing all three possible logic states for evaluation.

Does it help to find transmission errors?

Three main transmission timing errors can occur. The first is a wrong or insufficient reaction time of one device to the requests from the other one. If the reaction time is too slow there might be an overload of the computing unit. That could lead to a data overflow which ends up in data loss.

Also violations against protocol times are common error sources and lead to a wrong interpretation of the data flow.

A similar problem can arise when the reaction to a change in the protocol lines is too slow. Setting the RTS line to stop must quickly lead to quit sending characters, ideally after the current character has been sent out. If the transmitter sends some more characters or in worst case does not think of stopping sending data, overflows and data losses will intermittently occur, since the receiver's internal cache can not be unloaded fast enough.

The third error is seldom, but also possible and can not be detected with inappropriate tools. The serial data itself can be sent false. The bit rate can jitter or change (mostly inadvertently evoked by the operating program). Also the serial sender may be faulty and produce incorrect timed bits.

In this case the MSB serial receiver will produce a framing error event. If you synchronize the scope display of the Rxd and Txd line to this error you will directly see, how the data stream looked at the moment when the error occurred.

Dealing with any baud rate

One unique feature, found nowhere else, is the ability to set the baudrate for the serial channels to any baudrate between 1bps and 1Mbps (20Mbps at MSB-RS485-Plus), within an average accuracy of 0.1 percent. Of course you can also measure the baudrate of the examined connection within the same accuracy.

Why is the high resolution necessary?

When asynchronous transmission is used the receiver synchronizes and starts its data acquisition with the negative edge of the start bit. With a default oversampling baud clock of 16 times the baudrate it samples the data all 16 clocks, starting after the first 8 clocks. So the bits are captured in their middle, which is assumed as the most stable point, with 8 clocks in each case before and after the sampling point.

For 8 bit data plus start stop bits, the last bit, which is the stop bit, is sampled after 8+9*16=152 baud clocks.

Now let's assume, one baud clock generator is not exact but has a deviation of 4 percent. Or both generators have only 2 percent deviation, but in different directions. That results in a difference of 152*0.04 = about 6 clocks for sampling over the whole data byte. This is very near the borders of the sampled stop bit. Misinterpretation can happen. And it will happen when the physical signal is not perfect, provoked by slow transmitters or receivers, long cables, or high input capacities. High baudrate will increase the probability for erroneous bit detection. And if the summed deviations are more than 4 percent you can be sure about malfunctional devices.

Do not let the baud rates differ from each other more than a maximum of 3 percent. Unfortunately, due to cost reasons, modern integrated devices do not have own baud rate generation clock systems. The baud clock is derived from the available system clock whose frequency may be not appropriate to divide down to the baud rate clock for common baud rates. Compromises have to be made and sporadic and irregular errors are 'preprogramed'.

The MSB analyzer automatically measures the data rate on both serial lines to give hints for baud rate mismatch between both devices. The status display directly issues a warning if the measured baud rate deviates more than 3 percent from the selected baud rate.

When using the switch option you are also able to send data at varying baud rates to the devices. Simply to check their ability to deal with not perfect data transmissions.

By the feature to log data at any baudrate you can also check connections at uncommon transmission rates which sometimes are used inside systems, where the available system clock is directly divided to the baud clock. The resulting baud rate is not a multiple of 9600, but any other, which can not be correctly decoded by PC communication ports.

Investigate unknown transmission protocols

The analyzer provides its own decoding hardware for the serial transmission data. So it is easy to extend its features to scan the transfer protocol to find out which data format is used.
Precise baud rate measurement adjusts the receiver's baudrate for exact bit sampling. Further processing stages examine the data stream to find out how many user bits are transferred per character and if the highest bit is used as parity.

When is this analysis used?

Sometimes the real data format is not known if you have older devices for which the interface descriptions got lost. Or you are not sure if the sender really adds the correct parity bit.
You always should verify the data structure before starting the examination of the communication.
Simply insert the analyzer into the active connection, press 'scan' and after having decoded some characters you will get information about the probable data format.

Having active influence on the RS232 communication

By breaking the interface lines and feeding them through a gate array some optional features are available. The signal do not have to be connected directly 1:1, but can be rerouted, inverted and manipulated. Loop back functionality, switching, breaking and changing protocol lines are possible. Even the undriven state is supported to break lines completely.
Also sending data manually with well defined settable space between characters are supported. The independency from PC latency times allows generating totally homogeneous data streams.
Also the fixed sampling and receiving line scheme can be altered. All lines can be used for the data transport.
An advanced schematic editor makes it easy to draw all kinds of circuits which connect inputs and outputs of the analyzer ports to each other or to additional objects like the serial input output channels. The only limitation is the fact, that input pins can not be changed to outputs and vice versa.

What can you do with this option?

There are three basic application fields. First you mostly can save the time to make an appropriate adaptor when uncommon lines are used for data and control purposes. Also you can create loop back adaptors for checking both devices individually. This is a time saving advantage.
The second field of application is the active switching of the control lines. So it is easy to simulate protocol line reaction or inversion. Especially in the interface development phase this access to the control lines can be of big advantage. Testing the reaction on rts/cts protocol is now made easy.
The third way for using the switch option is the most important one. You can break the normal data stream and send data manually instead. While the normal communication is paused you can send to and receive data from both devices to set new configurations or read out the current states. Sending random data or a file with certain application specific data with decreasing the spaces between the single characters will show how much the device will endure before it is overloaded. A lot of tests can be thought of.

Examine single RS485 bus segments

The segment mode is a great functionality of the RS485 analyzer to find erroneous activities or data on the bus. If two or more devices share the same bus it is hard or even impossible to assign the faulty data to a certain device.
Most time at least an idea exists which device does not work correctly. If this device, or a group of devices, can be watched isolated from the rest of the bus, the data can easily assigned.
This is the speciality of the segment mode.

How is it achieved?

The bus is cut and passed through the analyzer which generates two bus segments. The analyzer cares for a correct transmission between both segments and determines the data direction.
With this information the data can be assigned to both data segments. By varying of the segments the interesting devices can easily be watched isolated from the rest of the bus.

What else can the analyzer do for you?

The above mentioned advantages are all based on the independent and unique hardware concept of the MSB analyzer. Of course a lot more features are implemented in hardware and software. Many ways to evaluate, display and log will help to trace the communication.

Outstanding Software concept

The multiview concept let you have any number of evaluating and display windows on the screen, showing different parts of the recorded data events, synchronized or singular.
Watching the data while logging is not self-evident. The data can be examined in different stages from the logical signal level to sophisticated protocol displaying.
Great stress was put on the outstanding feature to display any data protocol in detail and colorized. Lots of different protocols are already included, new ones are continually added.
And even if the analyzer does not (yet) support your favorite protocol or if you use proprietary data structures you do not get a problem.
Thanks to the integrated script language LUA you can extend the display to the used protocol by yourself, this is a worldwide unique capability.

Permanent further development

The analyzer is under steady improvement in software and hardware. Many additional features will extend the capabilities and advantages and can be loaded from our web page as new software versions.
Of course free of charge.

More information can be found here:
www.iftools.com

For questions contact us at:
support@iftools.com