MSO2000 / DPO2000 Series

RS-232/422/485 / UART Triggering and Decoding Lab

Equipment List

One MSO/DPO2000 Series oscilloscope with firmware version ≥1.03
One DPO2COMP application module
One P2221 passive probe
DPO Demo Board and USB cable (020-2694-XX and 174-4401-00) or
DPO Demo 2 board and USB cable (020-2924-XX and 174-4401-00)
[Note: screen shots in this document were made with 020-2924-XX demo board]

Understanding the RS-232 Bus

Introduction

RS-232 stands for **R**ecommended **S**tandard 232, a communication standard from the **E**lectronic **I**ndustries **A**lliance (EIA), which was developed in the early 1960s for interconnection between teletype terminals and modems. The standard was updated to RS-232**C** in 1969 to specify electrical signal characteristics, mechanical interconnects, etc.

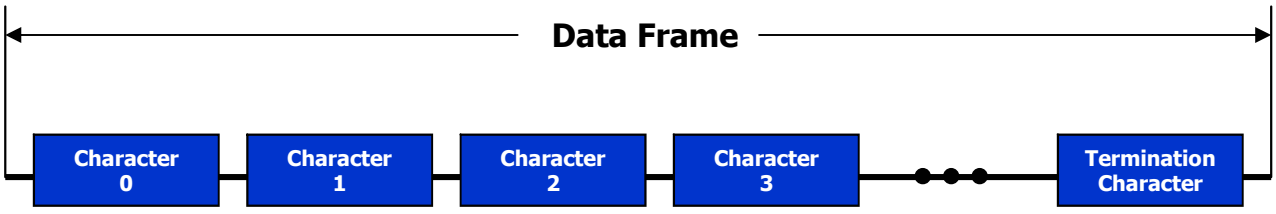
RS-232 provides two single-ended signals for point-to-point, full-duplex communication (simultaneous transmitted and received data). The standard does not specify character encoding, data framing, or protocols. It was designed for short-distance, low-speed serial data communication. Although the maximum cable length is not specified, a distance of less than 15 meters is recommended. The maximum data rate is not also specified, but rates <20 kb/s are recommended.

RS-232 data transmission is asynchronous, meaning that the clock is not transmitted and must be programmed in advance at both the transmitter and the receiver. Each character begins with a start bit, a high value which equates to a logic "0". The character is comprised of 7 or 8 data bits, which must also be programmed. The data bits are transmitted in least-significant to most-significant bit order. The optional Parity bit is next. If not used, the bit is ignored. If used, the polarity must be programmed, and provides simple error detection by indicating whether there are an odd or even number of "1s" in the data word. Finally, the character is usually terminated in one to two stop bits.

Start	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	opt. Data 7	opt. Parity	Stop	opt. Stop
1-bit	1-bit	1-bit	1-bit	1-bit	1-bit	1-bits	1-bit	1-bit	1-bit	1-bit	1-bit

Each RS-232 character can be encoded in various formats, but ASCII format is most commonly used. ASCII, short for **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange, is a 7-bit code (a range from 0 to 127) which is used to represent characters. Of the 128 possible codes, 95 (numbered 32 to 126) represent printable characters. Many of the remaining non-printing characters are control characters which control how text is processed. (Examples of control characters include backspace, tab, carriage return, and line feed.) Since most computer memories are based on 8-bit bytes, the eighth bit of the stored ASCII character can be used for parity, a simple error-detection scheme. (An ASCII conversion chart is included later in this lab document.)

The RS-232 standard does not specify how data content is framed or grouped, but a common technique is to end a data frame with a pre-determined termination character such as carriage return, line feed, or null.



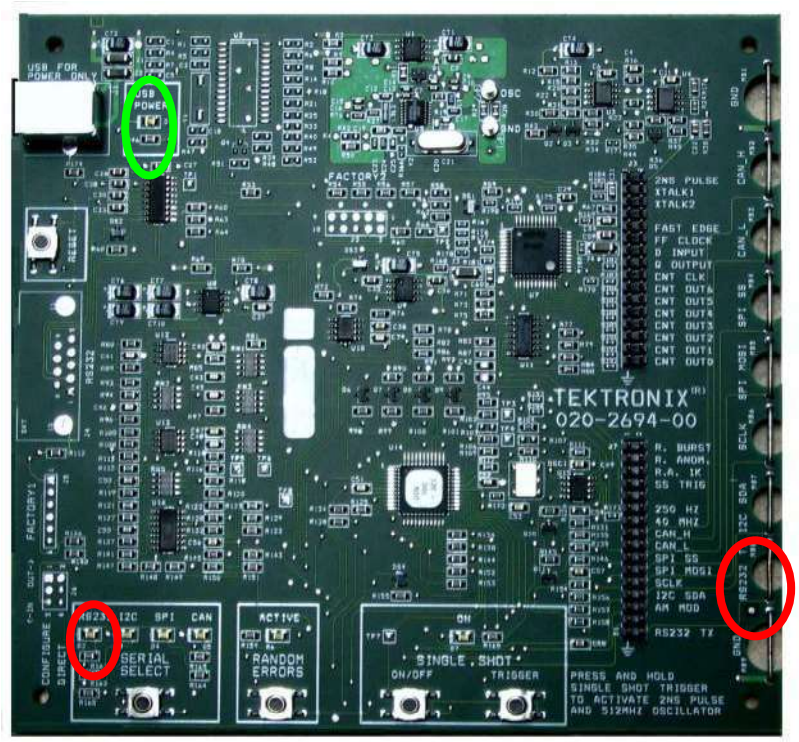
Lab Objectives

- Obtain a basic understanding of the RS-232 serial bus.
- Learn how customers use oscilloscopes to measure and decode RS-232 signals.
- Learn how to setup an RS-232 serial bus display and trigger and search on RS-232 packet content with an MSO/DPO2000 Series oscilloscope.

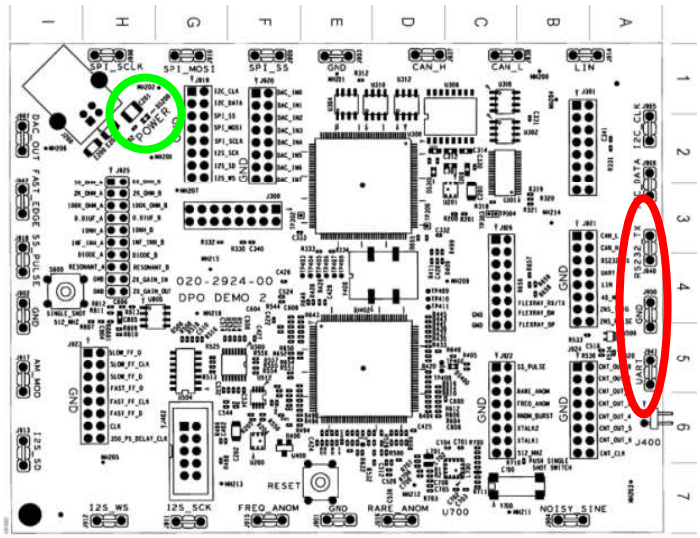
MSO/DPO2000 Series Lab Setup	
Key Take Away Points • RS-232 and related serial buses are industry standards and can be found in many of our customers' embedded designs today. • Traditional manual decoding methods being used by your customers to decode these buses on their oscilloscope are time-consuming. • With the DPO2COMP application module installed, the MSO/DPO2000 Series oscilloscope can trigger, decode and search RS-232, RS-422, RS-485, and UART serial bus traffic. It can be installed in either of the two application module slots on the front panel of the instrument.	**MSO/DPO2000 Series Procedure:** ❑ With the oscilloscope power off, install the **DPO2COMP** application module into one of the slots on the front panel of the instrument. ❑ Power up the oscilloscope. ❑ Press the front panel **Utility** button. ❑ Press the **About** bottom bezel button until the menu is displayed. ❑ Verify that the **DPO2COMP Computer Serial Triggering and Analysis** application module is detected. ❑ Verify that the instrument firmware version is at least 1.03.

Key Take Away Points

- The DPO Demo board (020-2694-XX) has an RS-232 signal which we can use for this lab:



- The DPO Demo 2 board (020-2924-XX) also has an RS-232 and a UART signal which we can use for this lab:



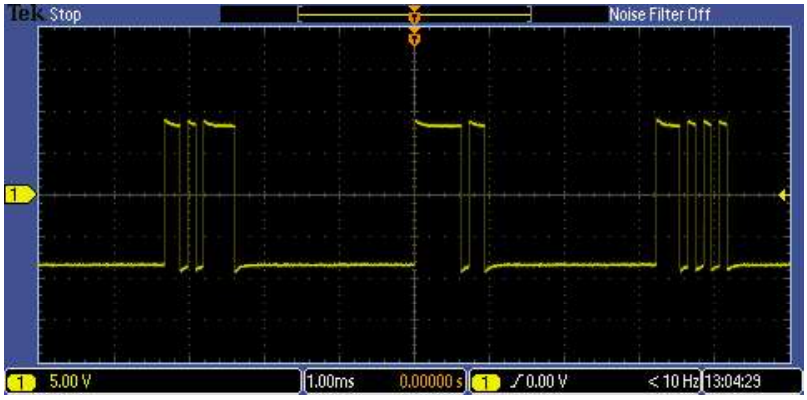
DPO Demo Board Procedure:

- ❑ Verify the USB cable is plugged into the MSO/DPO2000 Series oscilloscope and the DPO demo board.
- ❑ Verify the green **POWER** LED is lit.
- ❑ If using the 020-2694-XX board, verify the red **RS232** LED is illuminated on the DPO demo board. If not, press the **SERIAL SELECT** button until the **RS232** LED is lit.
- ❑ Attach a P2221 probe to the Channel 1 input of the oscilloscope. Then connect the probe ground to **GND** and connect the probe to the **RS232 TX** or **RS232_TX** test point.

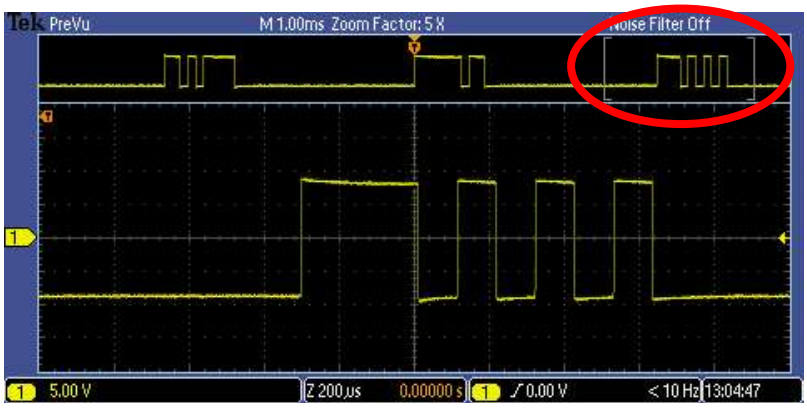
Manual RS-232 Bus Decoding

Key Take Away Points

- Notice that RS-232 signals are generally large signals, going positive and negative, with amplitudes from 3V to 15V peak. These large amplitudes provide a very simple immunity from noise.
- **Warning:** Because the signals may be larger than the P6516 digital probe can handle, only the analog channels should be used to acquire RS-232 signals.
- The display should look about like this:



- Zoom in on one of the bursts of activity. This is a single RS-232 character. The display should look something like this:

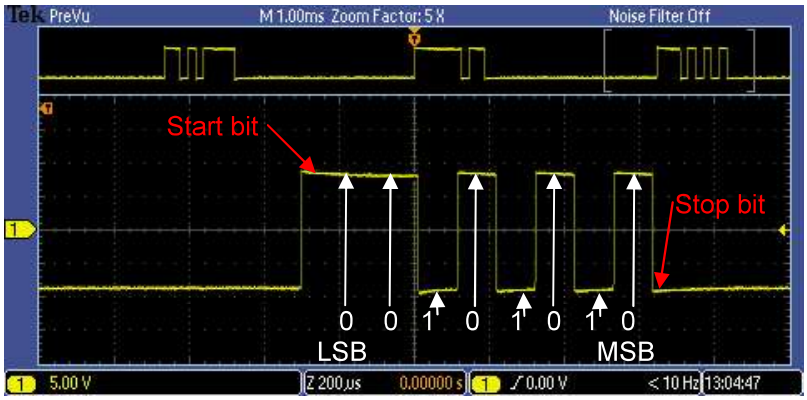


MSO/DPO2000 Series Procedure:

- ❑ Press the front panel **Default Setup** button to get the oscilloscope into a known state.
- ❑ Set the channel **1** Vertical **Scale** to **5V/div**.
- ❑ Use the Trigger **Level** knob to adjust the trigger level to the center of the waveform, about **0V**.
- ❑ Set the Horizontal **Scale** to **1ms/div**. This setting should allow a few bursts (RS-232 characters) to be displayed on screen.
- ❑ Press the front panel **Acquire** button.
- ❑ Press the **Record Length** bottom bezel button.
- ❑ Using the side bezel buttons, change the Record Length to **125k points**.
- ❑ Press the front panel **Menu Off** button twice.
- ❑ Press the front panel **Single** button.
- ❑ Using the **Wave Inspector** pan and zoom controls, zoom in on one whole burst of signal.

Key Take Away Points

- Notice the relatively long idle periods between signal bursts where the signal is low.
- The RS-232 character begins with a positive pulse after the idle period.
- The data bits come next, starting with the least-significant bit and ending with the most-significant bit. (In this case, we know that there are 8 data bits and no parity bit.) Low signals are digital 1s and high signals are digital 0s.



- The binary message above, written in most- to least-significant bit order is 01010100. This can also be written as 54 (hex) or ASCII "T", as shown below.
- Most engineers would prefer to use hexadecimal or "hex" notation, rather than binary. You can use the following chart to translate each group of 4 binary bits to a hex character:

Binary	Hex		Binary	Hex
0000	0		1000	8
0001	1		1001	9
0010	2		1010	A
0011	3		1011	B
0100	4		1100	C
0101	5		1101	D
0110	6		1110	E
0111	7		1111	F

- Since RS-232 is often used to transmit text characters, an even more popular code is ASCII. You can use the following chart to translate each hex character to a text character:

MSO/DPO2000 Series Procedure:

- ❑ Identify the start bit of the RS-232 character.
- ❑ Visually divide the rest of the character into 8 equal sections.
- ❑ If the signal is low in any of the 8 sections, write a "1" in the corresponding space below. If the signal is high in any of the 8 sections, write a "0" in the corresponding space below. (See example at the left.)

LSB MSB

____ ____ ____ ____ ____ ____ ____ ____

- ❑ Now, swap the order of the bits you wrote down, and divide the 8 bits into two 4-bit groups ("nibbles"). This is the binary representation of the RS-232 character. Write the binary character in the reverse order:

MSB LSB

____ ____ ____ ____ ____ ____ ____ ____

- ❑ Now, using the table at the left, translate the character to hex and write the hex value below.

____ ____

- ❑ Finally, using the hex-to-ASCII table, translate the character to hex and write the value below.

____

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

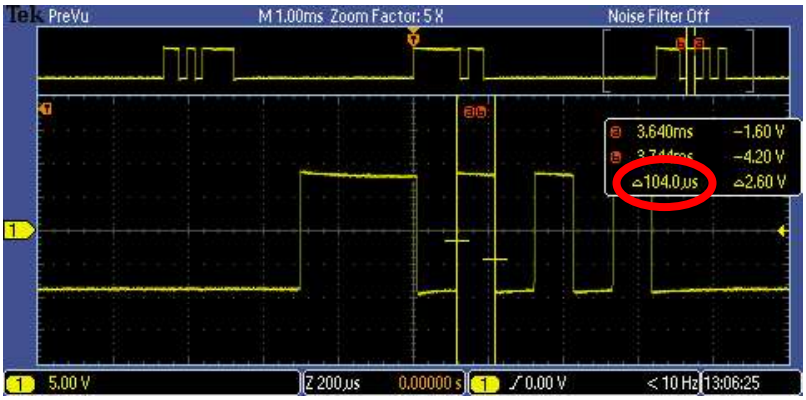
Conclusion: Manually decoding RS-232 data is a time-consuming process. Engineers are looking for a better and faster way to do this.

MSO/DPO2000 Series RS-232 Decoding Setup

Key Take Away Points
(optional) Determining RS-232 Bus Data Rate

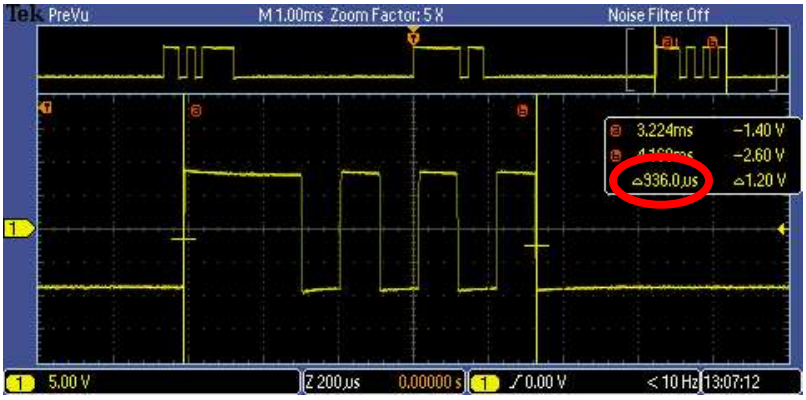
- The data rate of the RS-232 signal on the DPO demo board is 9600 bits/second (also sometimes called "9600 baud"). However, in general, you may need to determine the data rate using this procedure.

- The display should look about like this:



- Since the minimum pulse width is 104 µs, the data rate is verified to be 9600 bits/second.

- The display should look about like this:



- This signal uses 8 data bits with no parity.

MSO/DPO2000 Series Procedure:

- ❑ Using the **Wave Inspector** controls, zoom in on the narrowest pulse on the display.
- ❑ Press the front panel **Cursor** button once.
- ❑ Using the multipurpose knobs, measure the width of the narrowest pulse by positioning the cursor cross-hairs near the center of the rising and falling edges of the pulse.
- ❑ Read the pulse width (Δ time in the cursor readout) and write the value here: _____
- ❑ Look up the data rate in the table below:

Minimum Pulse Width	Data Rate (bits / sec)
26 µs	38400
52 µs	19200
104 µs	9600
139 µs	7200
208 µs	4800
417 µs	2400
500 µs	2000
556 µs	1800
833 µs	1200
1.7 ms	600
3.3 ms	300
6.7 ms	150
9.1 ms	110
13.3 ms	75

- ❑ Using the **Wave Inspector** controls, zoom in on one whole burst of signal.
- ❑ Read the burst width (Δ time in the cursor readout) and write the value here: _____
- ❑ Divide this value by the pulse width value you recorded earlier, and subtract one (for the Start Bit). Assuming that there is no Parity bit being used, the result is the number of data bits (7 or 8).

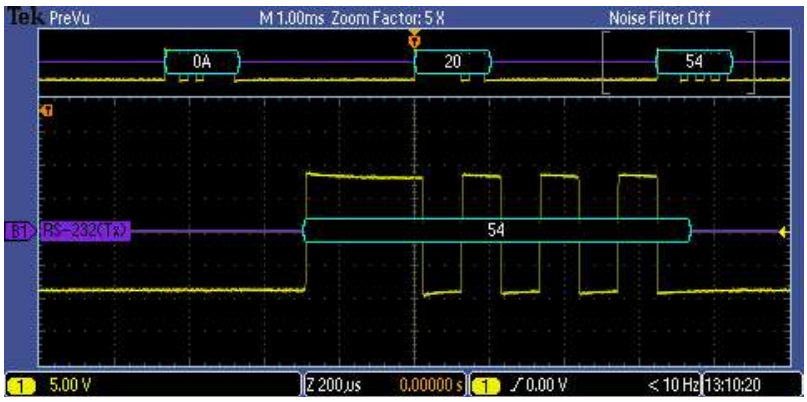
_____

- ❑ Press the front panel **Cursor** button until cursors are turned off.

MSO/DPO2000 Series RS-232 Bus Decoding

Key Take Away Points

- As you personally experienced in the last part of the lab, manually decoding RS-232 signals can be a time-consuming process. In this section, you will learn how to use the MSO/DPO2000 Series to automatically decode RS-232 packet content.
- Setting up a basic RS-232 bus waveform display takes only a few simple steps with the MSO/DPO2000 Series. After turning on a bus with a front panel button, the menus guide you through the setup in the left-to-right order across the bottom bezel buttons.
- Notice that any of the analog input signals can be used as a source for the RS-232 bus.
- The default RS-232 bus values in the MSO/DPO2000 Series were chosen to match the signal on the DPO demo board, so many of the steps to the right have already been done for you.
- The display should now look about like this, with the hexadecimal decoded value shown in the bus waveform:



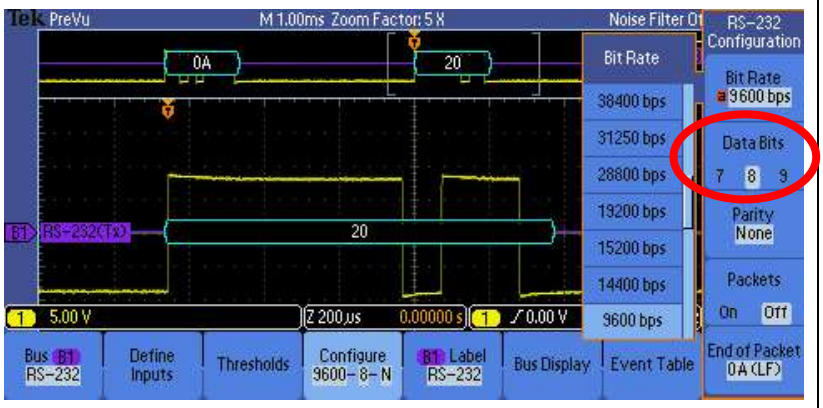
- A lot simpler than manually decoding the data!

MSO/DPO2000 Series Procedure:

- ❑ Press the front panel Bus **B1** button.
- ❑ Press the **Bus B1** bottom bezel button.
- ❑ Using the multipurpose **a** knob, select the **RS-232** Bus Type.
- ❑ Press the **Define Inputs** bottom bezel button.
- ❑ Using the multipurpose **a** knob, select channel **1** as the **Tx** Input.
- ❑ For RS-232 transmitted signals, press the **Polarity Normal (High = 0)** side bezel button. For RS-422, RS-485, and UART signal, press the **Polarity Inverted (High = 1)** side bezel button.
- ❑ Press the **Thresholds** bottom bezel button.
- ❑ Using the multipurpose **a** knob, set the Channel **1** Threshold value to the center of the signal, usually about **0 V** for RS-232 signals.
- ❑ Press the **Configure** bottom bezel button.
- ❑ Using the multipurpose **a** knob, set the Bit Rate to **9600** bps.
- ❑ Using the **Data Bits** side bezel button, select **8** bits.
- ❑ Using the **Parity** side bezel button, select **None**.
- ❑ If desired, you can edit the label on the bus. Press the **B1 Label** bottom bezel button and enter the desired label.
- ❑ Press the **Bus Display** bottom bezel button.
- ❑ Verify that the **Hex** side bezel button has been selected to enable hexadecimal decoding.
- ❑ Press **Menu Off** once to remove the side menu.

Key Take Away Points
Enhanced Flexibility in RS-232 Bus Setup

- We have had numerous customer requests for enhancements, especially higher baud rates, user-defined baud rates, and 9-bit-no-parity operation for some UART communication. These features can be seen in the menu below:



MSO/DPO2000 Series Procedure

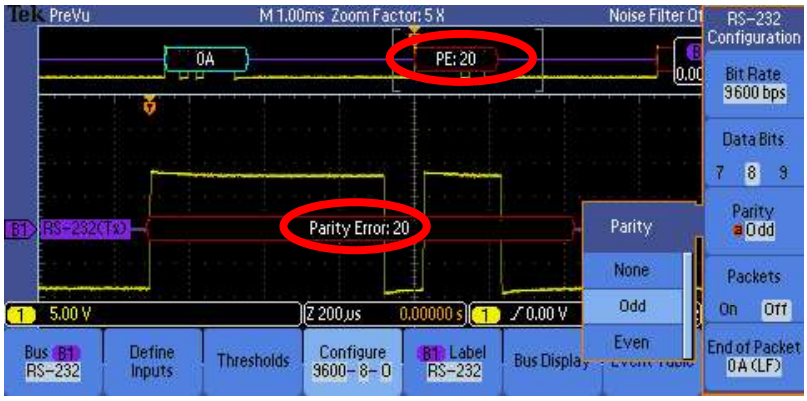
- ❑ Press the **Configure** bottom bezel button and notice the new baud rates, including **Custom**. Custom allows you to select baud rates from 50 bps to 10 Mbps. You will want to turn off the front panel Fine control if you are making a large change in Custom bit rates.
- ❑ Also notice that the **Data Bits** control allows for 9-bit data communication between UARTs.

Key Take Away Points

- With the early DPO/MSO4000 firmware versions, if the RS-232 decoder found a parity error, it replaced the decoded value with the text "Error", as shown below:

- We have had some customer requests to mark the parity errors but to also show the decoded value.

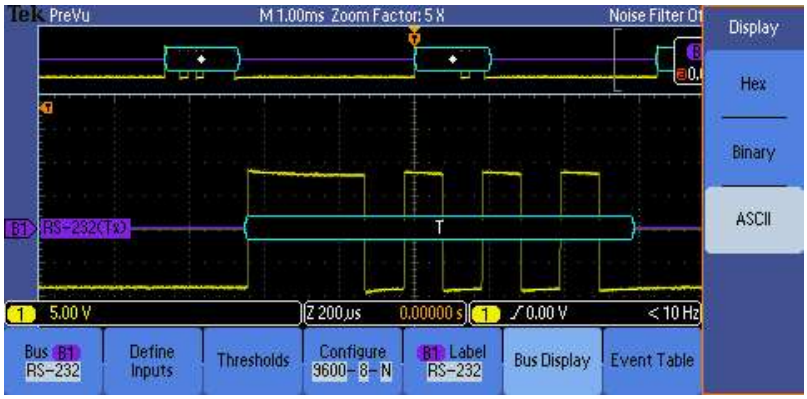
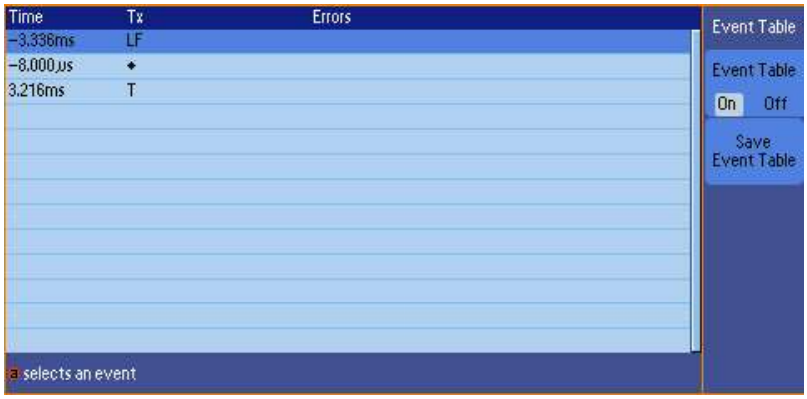
- Notice the MSO/DPO2000 (and current MSO/DPO4000 and DPO3000) firmware clearly indicates that an error has been detected, but it also displays the decoded value.



- This decoded data and parity error information is also shown in the Event Table display.

MSO/DPO2000 Series Procedure

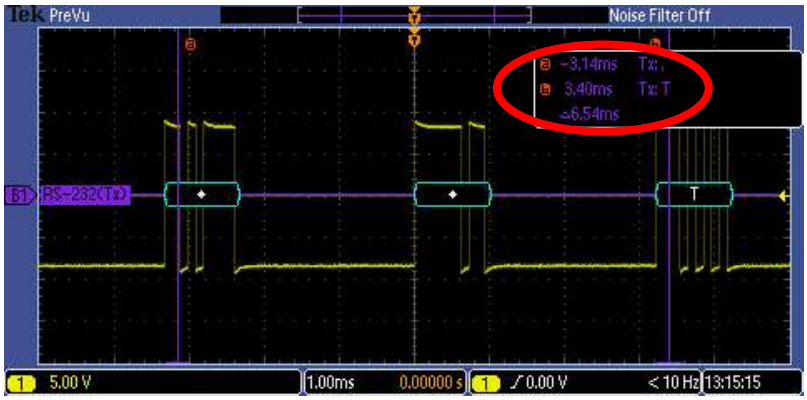
- ❑ The DPO demo 2 board does not produce RS-232 signals with parity errors, but we can simulate them with the following steps:
  - ❑ While in the RS-232 Configure menu, press the **Parity** side bezel button until **Odd** is selected. Since the hex number **51** has an odd number of '1' digits, this produces a parity error.
  - ❑ When you are done examining parity errors, return the **Parity** control to **None**.

Key Take Away Points	MSO/DPO2000 Series Procedure:
• Although the bus waveform display is very familiar to most hardware engineers, there re other display formats that other engineers, such as software engineers, may find more useful.	
• The first is the ASCII display, since RS-232 is often used to transmit text messages. The display should now look something like this, with the ASCII decoded value shown in the bus waveform:	❑ Press the **Bus Display** bottom bezel button. ❑ Press the **ASCII** side bezel button.
	
• For large quantities of data, and for those who are not comfortable with waveform displays, you can also use the Event Table to display the RS-232 data (in the selected format) along with the time stamps. The display should now look something like this:	❑ Press the **Event Table** bottom bezel button. ❑ Using the **Event Table** side bezel button, select **On**.
	
• Notice that, from the Event Table menu, you can also save the Event Table data in .CSV format with a single button press.	❑ Press the **Save Event Table** side bezel button to save the Event Table in CSV format to the pre-selected mass storage device. ❑ Using the **Event Table** side bezel button, select **Off**.

MSO/DPO2000 Series RS232 Cursors

Key Take Away Points

- The RS-232 serial cursors provide another tool to bridge between the serial waveform and the decoded values, showing absolute timing relative to the trigger, bus values, and relative timing between bus events.
- For this example, we can measure the time from the start of one character to the start of the next:



- Even when you are zoomed out to where you can't read the decoded values on the display, cursor measurements enable you to make timing measurements on buses.

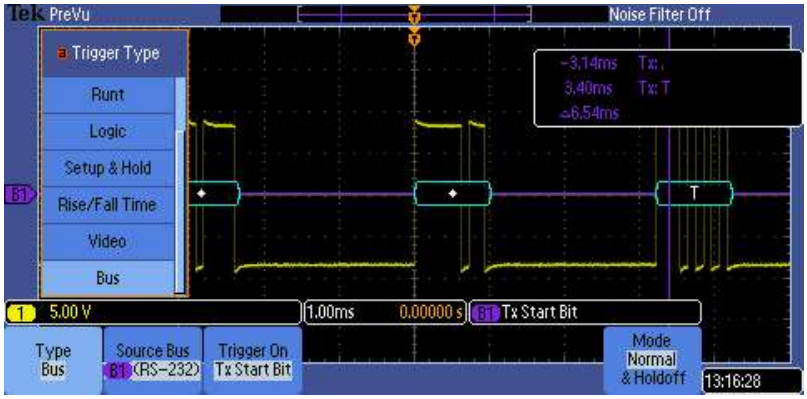
MSO/DPO2000 Series Procedure:

- ❑ Using the Wave Inspector knob, zoom in on one of the RS-232 messages.
- ❑ Press the **Cursors** front panel button once to turn on vertical bar cursors.
- ❑ Using the multipurpose **a** knob, place the **a** cursor on an event of interest, such as the start of the first character.
- ❑ Notice the **a** cursor time and bus value readout in the upper right corner of the display.
- ❑ Using the multipurpose **b** knob, place the **b** cursor on an event of interest, such as the start of the second character.
- ❑ Notice the **b** cursor time and bus value readout in the upper right corner of the display.
- ❑ Also notice that we are displaying the time delay between the first and second bus events of interest.
- ❑ Press the **Cursors** front panel button once to turn off cursors.

MSO/DPO2000 Series RS-232 Bus Triggering

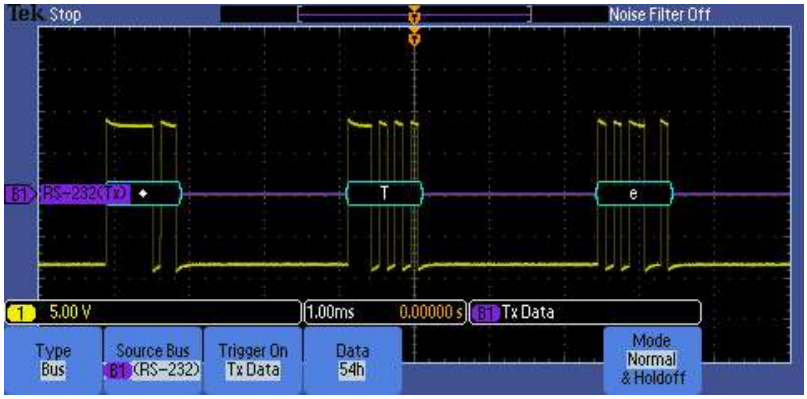
Key Take Away Points

- When debugging a system, you often want to capture the state of some key signals when a certain event occurs. One key event may be the transmission of specific content over the RS-232 serial bus.



- The MSO/DPO2000 Series can trigger on the Start Bit, End of Packet, or Data value for either the transmitted (Tx) or received (Rx) serial signal.

- By following this simple procedure, you can easily trigger the scope on a specified serial pattern, capturing each occurrence.



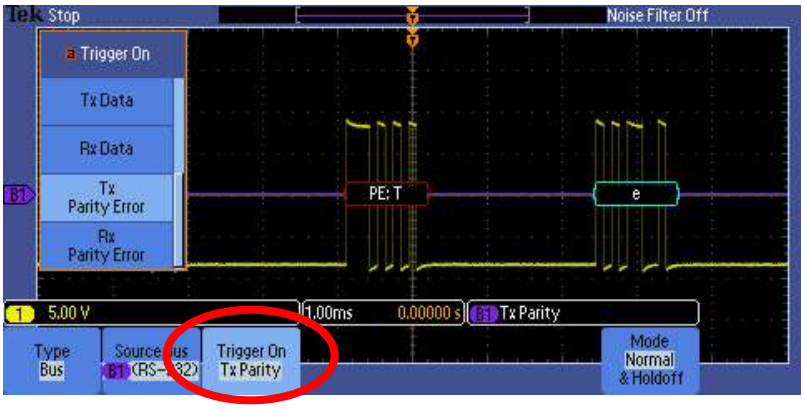
MSO/DPO2000 Series Procedure:

- ❑ Press the front panel Trigger **Menu** button.
- ❑ Press the **Type** bottom bezel button.
- ❑ Using the multipurpose **a** knob, select the **Bus** Trigger Type.

- ❑ Press the **Trigger On** bottom bezel button.
- ❑ Using the multipurpose **a** knob, set the oscilloscope to Trigger On **Tx Data**.
- ❑ Press the **Data** bottom bezel button.
- ❑ Press the **Data** side bezel button until the Data selection menu appears.
- ❑ Using the multipurpose knobs, set the Data pattern to **54** hex or an ASCII 'T' to capture the start of the "Tektronix" message.
- ❑ Press **Menu Off**.
- ❑ Press the front panel **Run/Stop** button.
- ❑ Adjust the Wave Inspector **Pan** control as needed.

Key Take Away Points

- The RS-232 decoder also has the ability to trigger on a parity error, as shown below:



MSO/DPO2000 Series Procedure

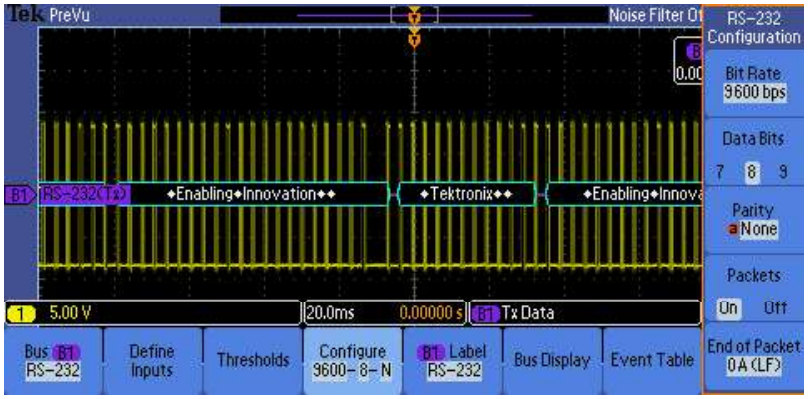
- ❑ Press the front panel **Trigger Menu** button.
- ❑ Press the **Trigger On** bottom bezel button.
- ❑ Using the multipurpose **a** knob, select **Tx Parity Error**.
- ❑ Verify that you have the parity set to Odd on the RS232 bus definition.
- ❑ Press the front panel **Single** button.

MSO/DPO2000 Series RS-232 Packet Mode Decoding

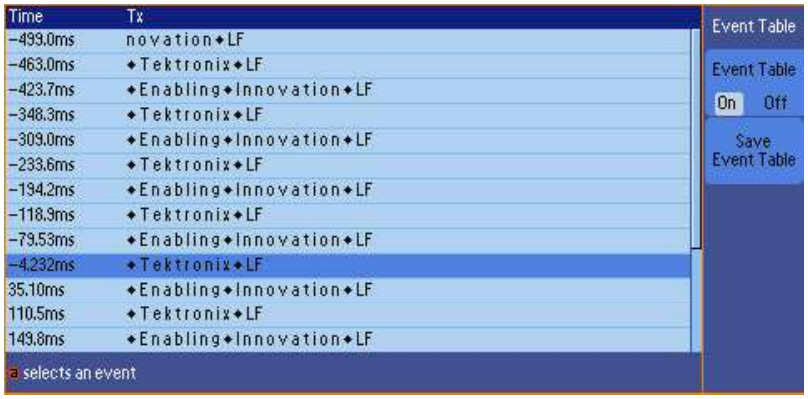
Key Take Away Points

- As you work with large amounts of RS-232 serial data, you may find that you want to focus less on the timing characteristics of the bus waveform and more on the data content. This is the purpose of the Packet mode decoding.

- Notice, in the overview window at the top of the display, the way the data stream has been divided into a few messages (blue packets). Notice how the text characters are displayed in an easily readable format in the zoom window. The display should now look something like this:



- The Event Table also displays the packets of messages, if possible, one per line. The display should now look something like this:



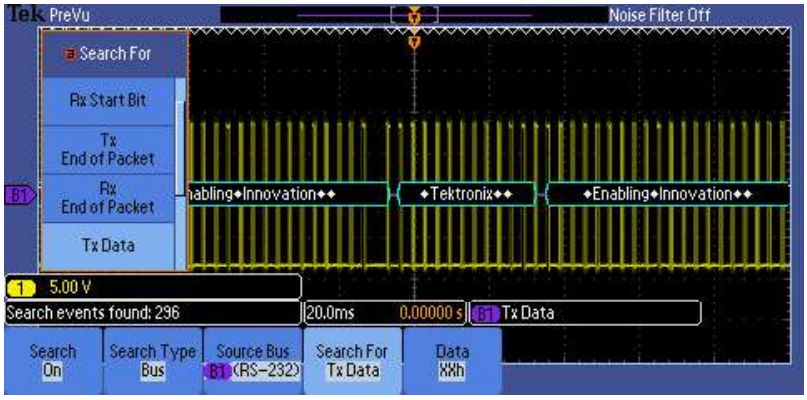
MSO/DPO2000 Series Procedure:

- ❑ Press the front panel zoom button to turn zoom off.
- ❑ Set the Horizontal **Scale** to **20ms/div** to display a few packets of RS-232 characters.
- ❑ Press the front panel **Acquire** button.
- ❑ Press the **Record Length** bottom bezel button.
- ❑ Press the **1.25M points** side bezel button.
- ❑ Press the front panel **Menu Off** button.
- ❑ Press the front panel Bus **B1** button.
- ❑ Press the **Configure** bottom bezel button.
- ❑ Press the **Packets** side bezel button until **On** is selected.
- ❑ Press the **End of Packet** side bezel button until the selection menu is displayed.
- ❑ Each text packet is delimited by a character, such as Null, Line Feed, Carriage Return, Space, or hex FF. In the case of the DPO demo board, each message is terminated with a Carriage Return (CR) and Line Feed (LF).
- ❑ Using the multipurpose **a** knob, select the Line Feed **0A (LF)** End of Packet character.
- ❑ Press the front panel **Single** button.
- ❑ Using the Wave Inspector **Pan** and **Zoom** controls, zoom in so you can read the decoded characters in the main display window.
- ❑ Press the **Event Table** bottom bezel button.
- ❑ Using the **Event Table** side bezel button, select **On**.
- ❑ Using the **Event Table** side bezel button, select **Off**.

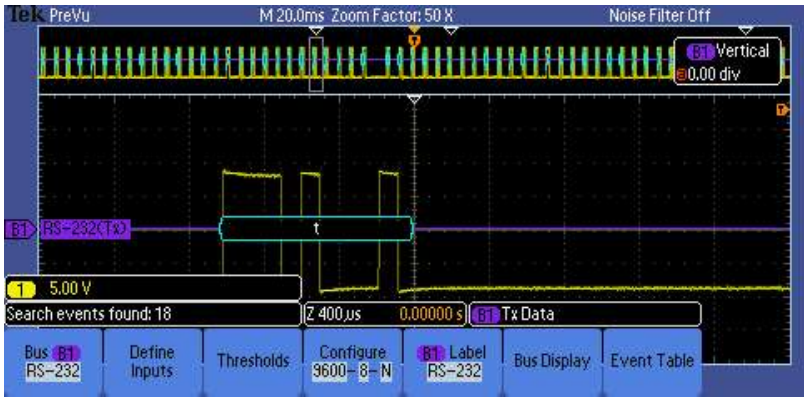
MSO/DPO2000 Series RS-232 Bus Searching

Key Take Away Points

- One of the key reasons for using long-record-length oscilloscopes is to be able to capture long time windows at high timing resolution.
- Capture a long time window. The display should look about like this:



- Wave Inspector search allows you to find all occurrences of a specified event, such as the letter 't'. The display should look something like this:



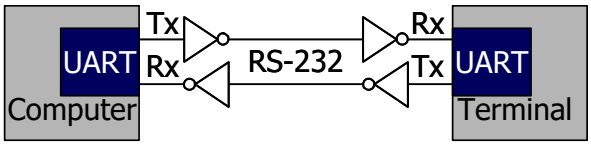
MSO/DPO2000 Series Procedure:

- ❑ Press the **Configure** bottom bezel button.
- ❑ Press the **Packets** side bezel button until **Off** is selected.
- ❑ Press the front panel **Search** button.
- ❑ Press the **Search** bottom bezel button.
- ❑ Press the Search side bezel button until Search is turned **On**.
- ❑ Press the **Search Type** bottom bezel button.
- ❑ Using the multipurpose **a** knob, select the **Bus** Search Type.
- ❑ Press the **Search For** bottom bezel button.
- ❑ Using the multipurpose **a** knob, select Search For **Tx Data**.
- ❑ Press the **Data** bottom bezel button.
- ❑ Press the **Data** side bezel button until the Data selection menu appears.
- ❑ Using the multipurpose knobs, set the Data pattern to **74** hex (ASCII 't', one of the most common letters in the English language).
- ❑ Notice all of the white triangular marks appear at the top of the display.
- ❑ Press **Menu Off**.
- ❑ Navigate between marked occurrences by using the front panel ← and → buttons.

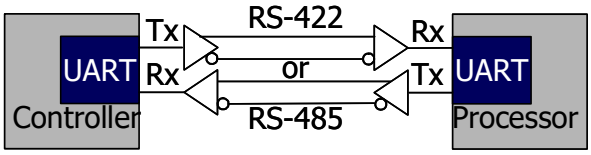
Conclusion: The MSO/DPO2000 Series bus decoding, triggering, and searching capabilities will save engineers valuable time when trying to find that elusive RS-232 bug. No longer will engineers have to manually decode RS-232 data waveforms to find the specific serial data values of interest.

MSO/DPO2000 Series Support for RS-232-related Standards

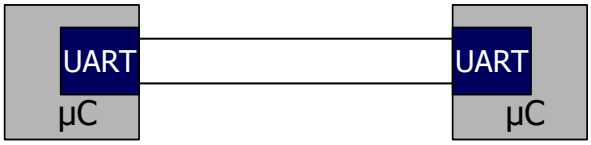
The primary application for RS-232 is where these serial signals are transmitted between modules, over cables, and between products. These RS-232 signals are transmitted single-ended, at relatively high voltage (up to ±15V), and inverted (so a digital 0 is a positive voltage and a 1 is a negative voltage).



RS-232 application example



Audio / video application example



Embedded communication application example

There are two other types of variants on the RS-232 family that are also addressed with the DPO2COMP.

The first is differential signaling, such as with the RS-422 and RS-485 standards. These standards specify transmission of a lower-voltage differential signal which is not inverted. Although one side of these differential signals can be probed with a passive probe, the TDP0500 and TDP1000 differential probes will provide better signal fidelity, especially in noisy environments.

The second variant on RS-232 signals is the transmission of these serial signals between components on a single circuit board. This embedded system application, most commonly communication between a microcontroller and a Universal Asynchronous Receiver / Transmitter (UART) or RS-232 driver/receiver IC. These signals are single-ended and non-inverted, with standard logic levels.

For both of these non-inverting RS-232 variants, you need to select the Polarity Inverted (High = 1) side bezel button in the Bus Define Inputs Menu. Otherwise, the signals are treated like RS-232 signals.