Keywords: clock accuracy, UART, crystal resonator, crystal oscillator, ceramic resonator, econo-oscillator, econoscillator, UARTs, RS232, RS-232, RS485, RS-485, asynchronous, EIA, ITU, ISO, TIA, EIA/TIA-232-F

Aug 07, 2003

APPLICATION NOTE 2141

Determining Clock Accuracy Requirements for UART Communications

Abstract: This application note discusses the timing requirements for the commonly used serial asynchronous communications protocol implemented in UARTs. The article shows how to determine the tolerance for the UART clock source at both ends of an asynchronous link.

Background

The RS-232 specification dates back to 1962, when it was first released by the Electronic Industries Alliance (EIA). The specification has changed over time, incorporating higher data rates and closing the compatibility gaps between the Telecommunications Industry Association (TIA), International Telecommunication Union (ITU), and International Standards Organization (ISO) requirements. The current version of the RS-232 specification is EIA/TIA-232-F, issued in October 1997.

RS-232 benefited from the availability of MSI ICs from the late 1970s, which had the complexity to handle the specification at a reasonable cost. These ICs are universal asynchronous receive transmitters (UARTs). Many large-scale integration (LSI) ICs (including microcontrollers) now include the functionality.

As is often the case, the availability of UARTs drove the industry to use the RS-232 serial protocol in non-RS-232 ways. Common examples are RS-485 transmissions, optoisolated transmissions, and transmissions using a single-ended physical layer (that is, 0 to 3.3V instead of ±5V or ±10V). This article covers the general timing aspects of the serial interface, not application nuances of handshaking or the physical layer. Therefore, the application note is applicable to all generalized UART applications.

UART Timing

A typical UART frame is shown in Figure 1. It comprises a START bit, 8 data bits, and a STOP bit. Other variants are also possible in RS-232 applications. For example, the data packet could be 5, 6, or 7 bits long, there could be 2 STOP bits, or a parity bit could be inserted between the data packet and the STOP bit for rudimentary error detection. Figure 1 shows the signaling as seen at a UART's transmit data (TXD) or receive data (RXD) pins. RS-232 bus drivers invert as well as level shift, so a logic 1 is a negative voltage on the bus and a logic 0 is a positive voltage.

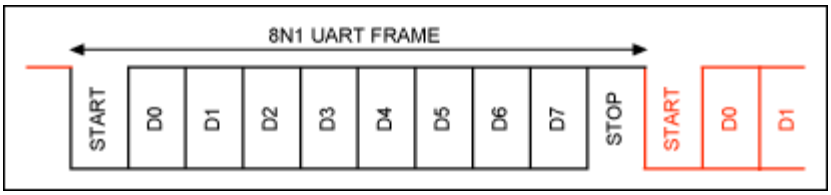

Figure 1. A typical UART data frame.

When two UARTs communicate, it is a given that both transmitter and receiver know the signaling speed. The receiver does not know when a packet will be sent (no receiver clock); hence, the protocol is termed "asynchronous." The receiver circuitry is correspondingly more complex than that of the transmitter. The transmitter simply has to output a frame of data at a defined bit rate. Contrastingly, the receiver has to recognize the start of the frame to synchronize itself, and therefore determine the best data-sampling point for the bit stream.

Figure 2 shows a common method used by a UART receiver to synchronize itself to a received frame. The receive UART uses a clock that is 16 times the data rate. A new frame is recognized by the falling edge at the beginning of the active-low START bit. This occurs when the signal changes from the active-high STOP bit or bus idle condition. The receive UART resets its counters on this falling edge, expects the mid-START bit to occur after 8 clock cycles, and anticipates the midpoint of each subsequent bit to appear every 16 clock cycles thereafter. The START bit is typically sampled at the middle of bit time to check that the level is still low and ensure that the detected falling edge was a START bit, not a noise spike. Another improvement is to sample the START bit three

times (clock counts 7, 8, and 9, out of 16) instead of sampling it only at the midbit position (clock count 8 out of 16).

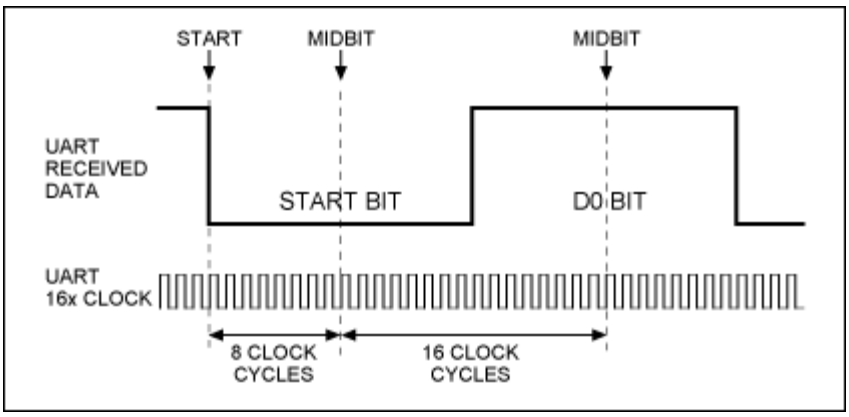

Figure 2. UART receive frame synchronization and data sampling points.

Timing Accuracy

How accurate must the receive UART clock be to receive data correctly? Since the absolute clock rate is unimportant for the purposes of accurate reception, a better question would be to ask how different the transmit and receive UART clocks can be. The first point to understand is that because the UART receiver synchronizes itself to the start of each and every frame, we only care about accurate data sampling during one frame. There is no buildup of error beyond a frame's STOP bit, which simplifies analysis because we only have to consider one frame for the worst-case scenario.

When do we get a timing error due to transmit-receive clock mismatch? Our goal is to sample each bit at the midpoint (Figure 2). If we sample one-half a bit-period too early or too late, we will be sampling at the bit transition and have problems (Figure 3).

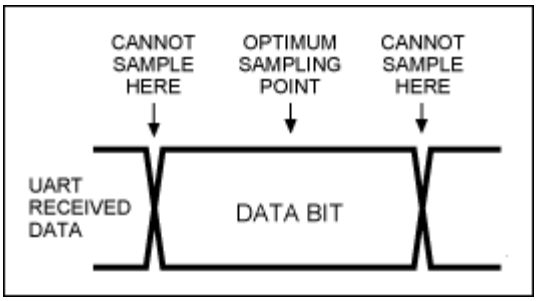

Figure 3. UART receive sampling range.

In reality, we cannot sample close to the bit-transition point reliably. The primary reason for this is the finite (and typically slow) transmission rise and fall times. These times become even slower if overly capacitive cabling is used. A long bus incurs high attenuation, which reduces noise margin and makes it more important to sample when the bit level has settled.

It is difficult to quantitatively assess a worst-case acceptable sampling range across a bit's period. EIA/TIA-232-F does specify a 4% of bit-period maximum slew time for a transmission, but this is difficult to achieve for long runs at 192kbps. But for the purpose of this application note, let us define two data path scenarios. Consider a "nasty" scenario, which can only be sampled reliably within the middle 50% of the bit time (Figure 4). This could equate to a long capacitive RS-232 run. The "normal" scenario can be sampled within the middle 75% of the bit time (Figure 5). This equates to a relatively benign bus (such as a meter-length bus with buffered CMOS logic levels or an RS-485 differential pair) within an equipment chassis.

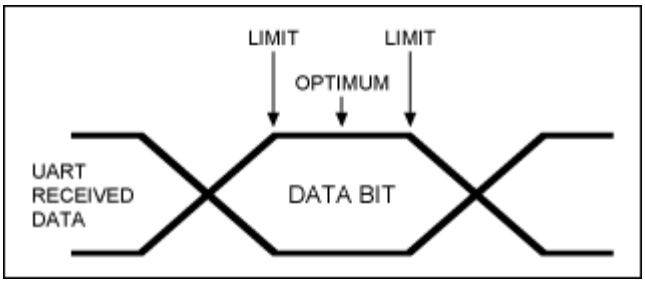

Figure 4. UART "nasty link" is sampled reliably within 50% of bit time.

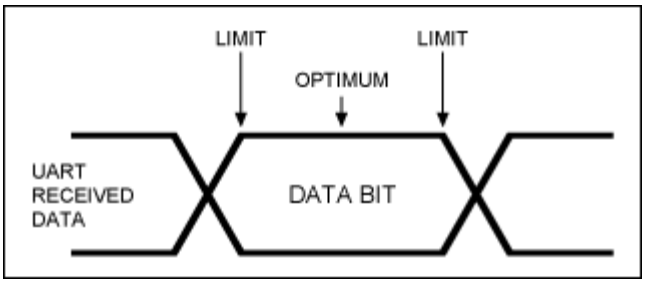

Figure 5. UART "normal link" is sampled reliably within 75% of bit time.

For Figures 4 and 5, we can determine that the error budget is ±25% and ±37.5% from the optimal bit-center sampling point for the nasty and normal scenarios, respectively. This error is equivalent to ±4 or ±6 periods of the 16x UART receive clock. Another error to include in this budget is the synchronization error when the falling edge of the START bit is detected. The UART will most likely start on the next rising edge of its 16x clock after detecting the START-bit. Since the 16x clock and the received data stream are asynchronous, the falling edge of the START bit could occur just after a 16x clock rising edge. The falling edge could, alternatively, occur just before the clock rising edge, but not with enough setup time to use it. This means that the UART has a ±1 bit error built in at the synchronization point. So our error budget reduces from ±4 or ±6 clock periods to ±3 or ±5 periods.

We will presume that short-term clock errors (essentially jitter) are very small, and therefore we are only considering midterm and long-term errors. These errors point to a mismatch in the transmit UART and receive UART timing that is consistent during a frame. Since the timing is synchronized at the falling edge of the START bit, the worst-case timing error will be at the last data sampling point, which is the STOP bit1. The optimum sampling point for the STOP bit is its bit center, which is calculated as:

> (16 internal clock cycles per bit) × (1 start bit + 8 data bits + ½ a stop bit) = (16) × (9.5) = 152 UART receive clocks after the original falling edge of the START bit.

Now we can calculate our allowable error as a percentage. For the normal scenario, the clock mismatch error can be ±5/152 = ±3.3%. For the "nasty" scenario it can be ±3/152 = ±2%. As hinted earlier, although the problem will materialize at the receive end of the link, clock mismatch is actually a tolerance issue shared between the transmit and receive UARTs. So presuming that both UARTs are attempting to communicate at exactly the same bit rate (baud), the allowable error can be shared, in any proportion, between the two UARTs.

Making use of the allowable error budget is helpful in systems where both ends of the link are being designed at the same time. This is partly because the tolerance of both ends will be known, and partly because trade-offs and cost savings can be made. In general, a standard low-cost, ceramic resonator with ±0.5% accuracy and a further ±0.5% drift over temperature and life can be used for the clock source at both ends of the link. This meets the 2% "nasty" scenario discussed earlier. If the system uses a master controller (typically a microcontroller or a PC) with a standard 100ppm crystal oscillator for the UART clock source, the link error can be cut approximately in half. Be careful with microcontrollers that synthesize baud frequencies for their internal UARTs. Depending on the choice of microcontroller clock, the baud rates may not be exact. If the error can be determined, it can be easily included in the link error budget.

1. It may appear odd that the STOP bit is sampled, but it is. If the STOP bit is detected as a low level instead of the expected high level, UARTs typically report a frame error.

Related Parts

MAX14830	Quad Serial UART with 128-Word FIFOs and Internal Oscillator	
MAX200	+5V, RS-232 Transceivers with 0.1µF External Capacitors	-- Free samples
MAX201	+5V, RS-232 Transceivers with 0.1µF External Capacitors	-- Free samples
MAX202	+5V, RS-232 Transceivers with 0.1µF External Capacitors	-- Free samples
MAX202E	±15kV ESD-Protected, +5V RS-232 Transceivers	-- Free samples
MAX203	+5V, RS-232 Transceivers with 0.1µF External Capacitors	-- Free samples
MAX203E	±15kV ESD-Protected, +5V RS-232 Transceivers	-- Free samples
MAX204	+5V, RS-232 Transceivers with 0.1µF External Capacitors	-- Free samples
MAX205	+5V, RS-232 Transceivers with 0.1µF External Capacitors	-- Free samples
MAX205E	±15kV ESD-Protected, +5V RS-232 Transceivers	-- Free samples

MAX206	+5V, RS-232 Transceivers with 0.1µF External Capacitors	-- Free samples
MAX206E	±15kV ESD-Protected, +5V RS-232 Transceivers	-- Free samples
MAX207	+5V, RS-232 Transceivers with 0.1µF External Capacitors	-- Free samples
MAX207E	±15kV ESD-Protected, +5V RS-232 Transceivers	-- Free samples
MAX208	+5V, RS-232 Transceivers with 0.1µF External Capacitors	-- Free samples
MAX208E	±15kV ESD-Protected, +5V RS-232 Transceivers	-- Free samples
MAX209	+5V, RS-232 Transceivers with 0.1µF External Capacitors	-- Free samples
MAX211	+5V, RS-232 Transceivers with 0.1µF External Capacitors	-- Free samples
MAX211E	±15kV ESD-Protected, +5V RS-232 Transceivers	-- Free samples
MAX212	+3V-Powered, Low-Power, True RS-232 Transceiver	-- Free samples
MAX213	+5V, RS-232 Transceivers with 0.1µF External Capacitors	-- Free samples
MAX213E	±15kV ESD-Protected, +5V RS-232 Transceivers	-- Free samples
MAX214	Programmable DTE/DCE, +5V RS-232 Transceiver	-- Free samples
MAX216	Low-Power, AppleTalk™ Interface Transceiver	-- Free samples
MAX218	1.8V to 4.25V Powered, True RS-232 Dual Transceiver	-- Free samples
MAX220	+5V-Powered, Multichannel RS-232 Drivers/Receivers	-- Free samples
MAX221	+5V, 1µA, Single RS-232 Transceiver with AutoShutdown	-- Free samples
MAX221E	±15kV ESD-Protected, +5V, 1µA, Single RS-232 Transceiver with AutoShutdown	-- Free samples
MAX222	+5V-Powered, Multichannel RS-232 Drivers/Receivers	-- Free samples
MAX223	+5V-Powered, Multichannel RS-232 Drivers/Receivers	-- Free samples
MAX225	+5V-Powered, Multichannel RS-232 Drivers/Receivers	-- Free samples
MAX230	+5V-Powered, Multichannel RS-232 Drivers/Receivers	-- Free samples
MAX231	+5V-Powered, Multichannel RS-232 Drivers/Receivers	-- Free samples
MAX232	+5V-Powered, Multichannel RS-232 Drivers/Receivers	-- Free samples
MAX232A	+5V-Powered, Multichannel RS-232 Drivers/Receivers	-- Free samples
MAX232E	±15kV ESD-Protected, +5V RS-232 Transceivers	-- Free samples
MAX233	+5V-Powered, Multichannel RS-232 Drivers/Receivers	
MAX233A	+5V-Powered, Multichannel RS-232 Drivers/Receivers	
MAX234	+5V-Powered, Multichannel RS-232 Drivers/Receivers	-- Free samples
MAX235	+5V-Powered, Multichannel RS-232 Drivers/Receivers	-- Free samples
MAX236	+5V-Powered, Multichannel RS-232 Drivers/Receivers	-- Free samples
MAX237	+5V-Powered, Multichannel RS-232 Drivers/Receivers	-- Free samples
MAX238	+5V-Powered, Multichannel RS-232 Drivers/Receivers	-- Free samples
MAX239	+5V-Powered, Multichannel RS-232 Drivers/Receivers	-- Free samples
MAX240	+5V-Powered, Multichannel RS-232 Drivers/Receivers	
MAX241	+5V-Powered, Multichannel RS-232 Drivers/Receivers	-- Free samples
MAX241E	±15kV ESD-Protected, +5V RS-232 Transceivers	-- Free samples
MAX242	+5V-Powered, Multichannel RS-232 Drivers/Receivers	-- Free samples
MAX243	+5V-Powered, Multichannel RS-232 Drivers/Receivers	-- Free samples
MAX244	+5V-Powered, Multichannel RS-232 Drivers/Receivers	
MAX245	+5V-Powered, Multichannel RS-232 Drivers/Receivers	
MAX246	+5V-Powered, Multichannel RS-232 Drivers/Receivers	
MAX247	+5V-Powered, Multichannel RS-232 Drivers/Receivers	
MAX248	+5V-Powered, Multichannel RS-232 Drivers/Receivers	
MAX249	+5V-Powered, Multichannel RS-232 Drivers/Receivers	
MAX3100	SPI/MICROWIRE-Compatible UART	-- Free samples
MAX3107	SPI/I²C UART with 128-Word FIFOs and Internal Oscillator	-- Free samples
MAX3110E	SPI/MICROWIRE-Compatible UART and ±15kV ESD-Protected RS-232 Transceivers with Internal Capacitors	-- Free samples
MAX3111E	SPI/MICROWIRE-Compatible UART and ±15kV ESD-Protected RS-232 Transceivers with Internal Capacitors	-- Free samples
MAX3140	SPI/MICROWIRE-Compatible UART with Integrated True Fail Safe RS-485/RS-422 Transceivers	-- Free samples
MAX3188	1Mbps, 1µA RS-232 Transmitters in SOT23-6	
MAX3188E	±15kV ESD-Protected, 1Mbps, 1µA RS-232 Transmitters in SOT23-6	-- Free samples
MAX3189	1Mbps, 1µA RS-232 Transmitters in SOT23-6	
MAX3189E	±15kV ESD-Protected, 1Mbps, 1µA RS-232 Transmitters in SOT23-6	-- Free samples
MAX3190	±15kV ESD-Protected, 460kbps, RS-232 Transmitters in SOT23-6	-- Free samples
MAX3190E	±15kV ESD-Protected, 460kbps, RS-232 Transmitters in SOT23-6	-- Free samples

MAX3202E	Low-Capacitance, 2/3/4/6-Channel, ±15kV ESD-Protection Arrays for High-Speed Data Interfaces	-- Free samples
MAX3203E	Low-Capacitance, 2/3/4/6-Channel, ±15kV ESD-Protection Arrays for High-Speed Data Interfaces	-- Free samples
MAX3204E	Low-Capacitance, 2/3/4/6-Channel, ±15kV ESD-Protection Arrays for High-Speed Data Interfaces	-- Free samples
MAX3206E	Low-Capacitance, 2/3/4/6-Channel, ±15kV ESD-Protection Arrays for High-Speed Data Interfaces	-- Free samples
MAX3209E	±15kV ESD-Protected, 12V, Dual RS-232 Serial Port with Low-Power Standby for Motherboards/Desktops	-- Free samples
MAX3212	+2.7V to +3.6V Powered, 1µA Supply Current, 3-Driver/5-Receiver, True RS-232 Transceiver	-- Free samples
MAX3218	1µA, 1.8V to 4.25V RS-232 Transceiver with AutoShutdown™	-- Free samples
MAX3221	1µA Supply-Current, True +3V to +5.5V RS-232 Transceivers with AutoShutdown	-- Free samples
MAX3221E	±15kV ESD-Protected, 1µA, 3.0V to 5.5V, 250kbps, RS-232 Transceivers with AutoShutdown	-- Free samples
MAX3222	3.0V to 5.5V, Low-Power, up to 1Mbps, True RS-232 Transceivers Using Four 0.1µF External Capacitors	-- Free samples
MAX3222E	±15kV ESD-Protected, Down to 10nA, 3.0V to 5.5V, Up to 1Mbps, True RS-232 Transceivers	-- Free samples
MAX3223	1µA Supply-Current, True +3V to +5.5V RS-232 Transceivers with AutoShutdown	-- Free samples
MAX3223E	±15kV ESD-Protected, 1µA, 3.0V to 5.5V, 250kbps, RS-232 Transceivers with AutoShutdown	-- Free samples
MAX3224	1µA Supply Current, 1Mbps, 3.0V to 5.5V, RS-232 Transceivers with AutoShutdown Plus	-- Free samples
MAX3224E	±15kV ESD-Protected, 1µA, 1Mbps, 3.0V to 5.5V, RS-232 Transceivers with AutoShutdown Plus	-- Free samples
MAX3225	1µA Supply Current, 1Mbps, 3.0V to 5.5V, RS-232 Transceivers with AutoShutdown Plus	-- Free samples
MAX3225E	±15kV ESD-Protected, 1µA, 1Mbps, 3.0V to 5.5V, RS-232 Transceivers with AutoShutdown Plus	-- Free samples
MAX3226	1µA Supply Current, 1Mbps, 3.0V to 5.5V, RS-232 Transceivers with AutoShutdown Plus	-- Free samples
MAX3226E	±15kV ESD-Protected, 1µA, 1Mbps, 3.0V to 5.5V, RS-232 Transceivers with AutoShutdown Plus	-- Free samples
MAX3227	1µA Supply Current, 1Mbps, 3.0V to 5.5V, RS-232 Transceivers with AutoShutdown Plus	-- Free samples
MAX3227E	±15kV ESD-Protected, 1µA, 1Mbps, 3.0V to 5.5V, RS-232 Transceivers with AutoShutdown Plus	-- Free samples
MAX3228	+2.5V to +5.5V RS-232 Tranceivers in UCSP	
MAX3228E	±15kV ESD-Protected +2.5V to +5.5V RS-232 Transceivers in UCSP and WLP	
MAX3229	+2.5V to +5.5V RS-232 Tranceivers in UCSP	
MAX3229E	±15kV ESD-Protected +2.5V to +5.5V RS-232 Transceivers in UCSP and WLP	
MAX3232	3.0V to 5.5V, Low-Power, up to 1Mbps, True RS-232 Transceivers Using Four 0.1µF External Capacitors	-- Free samples
MAX3232E	±15kV ESD-Protected, Down to 10nA, 3.0V to 5.5V, Up to 1Mbps, True RS-232 Transceivers	-- Free samples
MAX3233E	±15kV ESD-Protected, 1µA, 250kbps, 3.3V/5V, Dual RS-232 Transceivers with Internal Capacitors	
MAX3235E	±15kV ESD-Protected, 1µA, 250kbps, 3.3V/5V, Dual RS-232 Transceivers with Internal Capacitors	-- Free samples
MAX3237	3.0V to 5.5V, Low-Power, up to 1Mbps, True RS-232 Transceivers Using Four 0.1µF External Capacitors	-- Free samples
MAX3237E	±15kV ESD-Protected, Down to 10nA, 3.0V to 5.5V, Up to 1Mbps, True RS-232 Transceivers	-- Free samples
MAX3238	+3.0V to +5.5V, 1µA, Up to 250kbps, True RS-232 Transceiver with AutoShutdown Plus	-- Free samples
MAX3238E	+3.0V to +5.5V, 10nA, 250kbps RS-232 Transceivers with ±15kV ESD-	-- Free samples

	Protected I/O and Logic Pins	
MAX3241	3.0V to 5.5V, Low-Power, up to 1Mbps, True RS-232 Transceivers Using Four 0.1µF External Capacitors	-- Free samples
MAX3241E	±15kV ESD-Protected, Down to 10nA, 3.0V to 5.5V, Up to 1Mbps, True RS-232 Transceivers	-- Free samples
MAX3243	1µA Supply-Current, True +3V to +5.5V RS-232 Transceivers with AutoShutdown	-- Free samples
MAX3243E	±15kV ESD-Protected, 1µA, 3.0V to 5.5V, 250kbps, RS-232 Transceivers with AutoShutdown	-- Free samples
MAX3244	1µA Supply Current, 1Mbps, 3.0V to 5.5V, RS-232 Transceivers with AutoShutdown Plus	-- Free samples
MAX3244E	±15kV ESD-Protected, 1µA, 1Mbps, 3.0V to 5.5V, RS-232 Transceivers with AutoShutdown Plus	-- Free samples
MAX3245	1µA Supply Current, 1Mbps, 3.0V to 5.5V, RS-232 Transceivers with AutoShutdown Plus	-- Free samples
MAX3245E	±15kV ESD-Protected, 1µA, 1Mbps, 3.0V to 5.5V, RS-232 Transceivers with AutoShutdown Plus	-- Free samples
MAX3246E	±15kV ESD-Protected, Down to 10nA, 3.0V to 5.5V, Up to 1Mbps, True RS-232 Transceivers	
MAX3248E	+3.0V to +5.5V, 10nA, 250kbps RS-232 Transceivers with ±15kV ESD-Protected I/O and Logic Pins	-- Free samples

Automatic Updates

Would you like to be automatically notified when new application notes are published in your areas of interest? Sign up for EE-Mail™.

More Information

For technical support: http://www.maxim-ic.com/support
For samples: http://www.maxim-ic.com/samples
Other questions and comments: http://www.maxim-ic.com/contact

Application note 2141: http://www.maxim-ic.com/an2141
AN2141, AN 2141, APP2141, Appnote2141, Appnote 2141