# Eliminate Risk of Contention and Data Corruption in RS-485 Communications

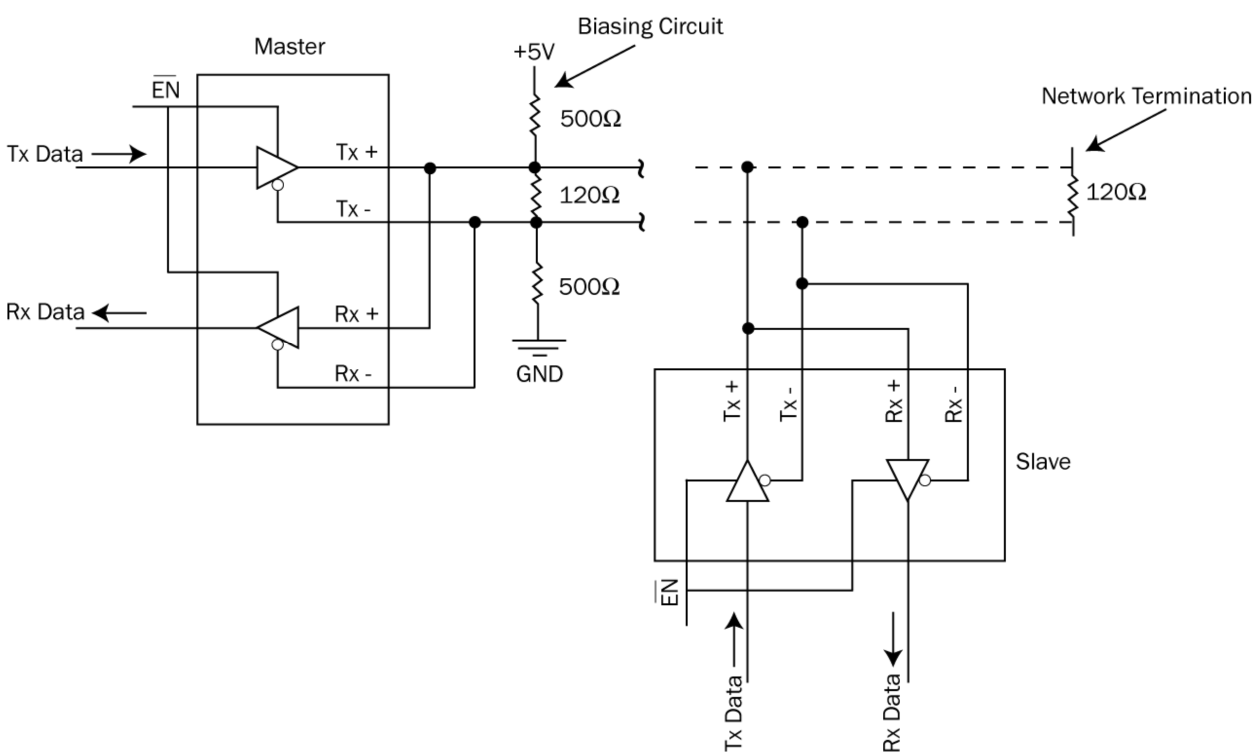
Earle Foster, Jeff Hunter
Sealevel Systems

## I. Background and Objective

The RS-485 communications standard was introduced in 1983 by the Electronic Industries Alliance (EIA) as a two-wire, half-duplex, multi-drop alternative to the point-to-point RS-232 interface. (Note: RS-485 can also be wired using 4-wires to enable full-duplex communication). RS-485 uses differential signaling to allow up to 32 devices to communicate peer-to-peer at distances up to 4000 feet. In most RS-485 configurations two signals, a positive and a negative, are run as a twisted pair. Since these signals are compared to one another rather than a ground reference like RS-232, RS-485 communication is less susceptible to external noise injected on the wiring and performs more reliably in industrial environments.

The following figure shows a typical RS-485 circuit.

**Typical RS-485 Network**



Bi-directional, half-duplex communication on a single twisted pair wire requires management of the data flow. If more than one device attempts to transmit at the same time, collisions will occur that result in data corruption. RS-485 transceivers allow the transmitter to be disabled, or “tri-stated”, so they are not driving the transmission line unless sending data. Bias resistors in the circuit force the signals to the idle state when the lines are not driven. When an RS-485 device sends data the transmitter must be enabled during transmission and disabled when transmission is complete to relinquish control of the RS-485 network.

The EIA RS-485 specification does not stipulate the method for controlling the driver, therefore a number of alternatives have evolved. The objective of this paper is to outline the most common approaches and compare their effectiveness in minimizing risks of communication errors.

The risk of communication error is directly related to the time required for a device to relinquish control of the RS-485 network once communication is complete so that other devices may send data without contention and data corruption.

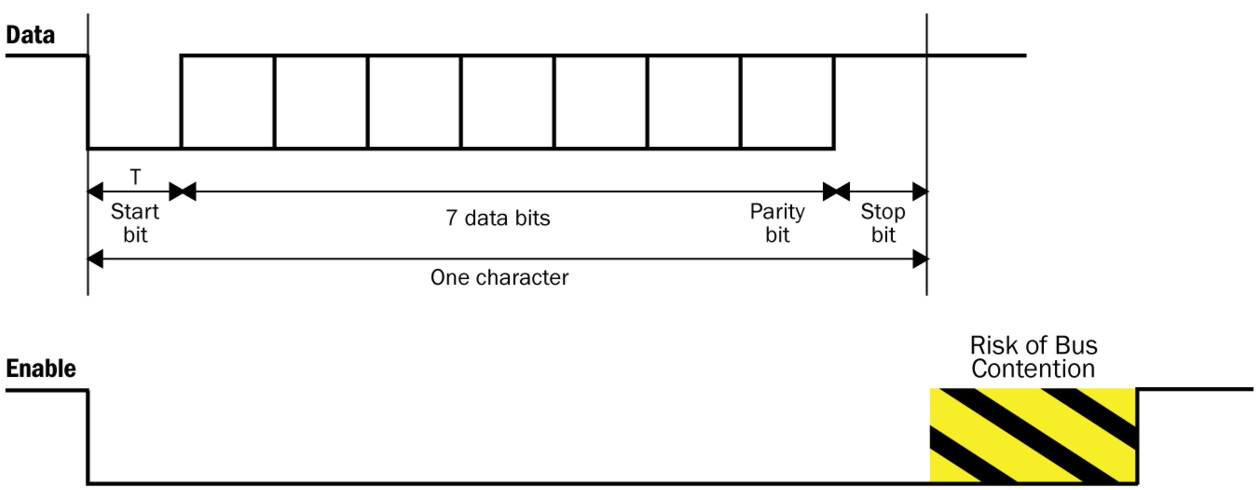
**II. Common Methods Used to Control RS-485 Device CommunicationsRS-485 Transmitter Enable Control Methods**

**A. Application Program Control (Under Windows)**

Rating:

The earliest method of RS-485 driver control was via the application program. In the DOS era this method was reasonably effective and commonly implemented. The application program would simply write to an I/O location (usually the RTS signal) when transmission from that device was required. After sending the data the application would again write to the same address to turn off the driver. However, with today's protected mode operating systems (Windows), the application program can no longer directly access I/O locations. Instead, the application must call a device driver to handle the request through the operating system. The time required for the application to call the driver, the driver to execute the action, and then return control to the application is indeterminate but it can be relatively long. The longer this time is to relinquish, the higher the risk of bus contention from other devices.

- User application program controls RS-485 enable
- Windows requires driver level interface
- Timing through Windows is indeterminate and could result in bus contention

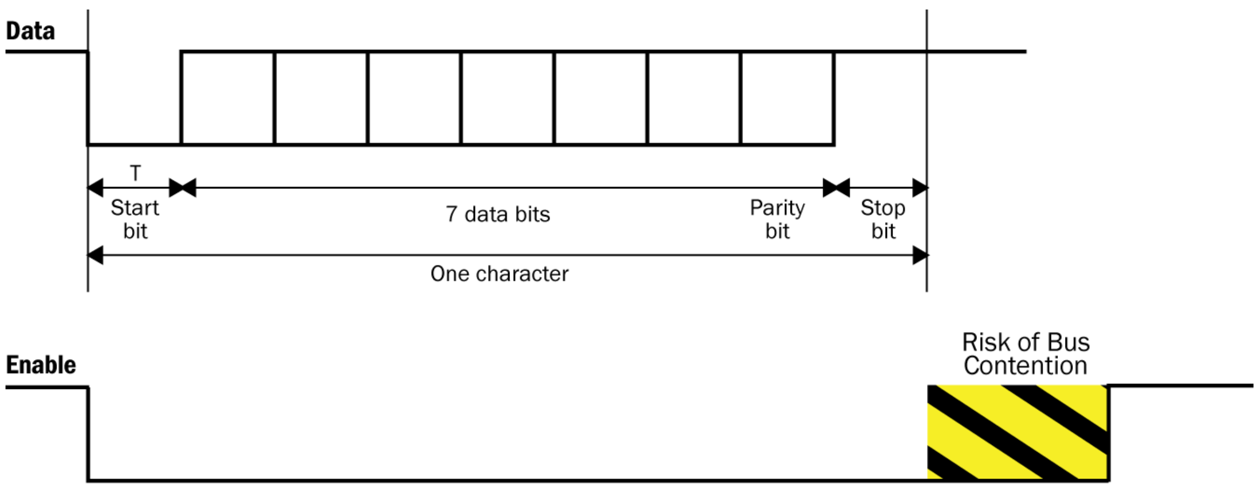


## B. One Shot

Rating: ★☆☆☆☆

Some RS-485 products use a technique called a “one-shot” to control the RS-485 enable signal. In a “one-shot”, a circuit is used to activate the enable signal upon transmission and then deactivate the signal using an RC circuit as a timer. The deactivation time for the enable depends on the “R” and “C” values used. Because the RC values are constant at all data rates these values must be set (sometimes using a jumper selection) to allow the enable to stay active for the slowest supported baud rate. This predetermined delay in deactivating the RS-485 enable could cause data error due to bus contention.

- Enable activated when data is transmitted
- RC values determine time before transmitter is disabled
- Predetermined RC values can result in data errors due to transmitter turning off too quickly or remaining enabled too long

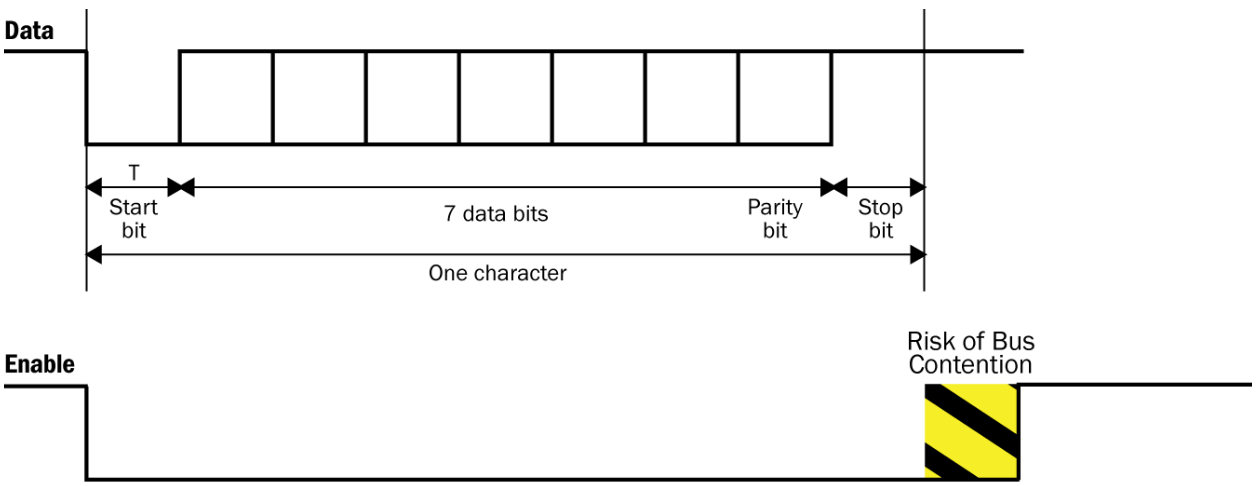


## C. Enable Driven when Data Present

Rating: ★★★☆☆

With this hardware approach the transmitter data line is connected to the transmitter enable pin. This method relies on the RS-485 bias resistors to provide the "high" (logic 1) data signal. Since the enable signal may be disabled during part of a message (whenever a logic "1" is present on the data line) this method does not properly adhere to the RS-485 specification requiring that the enable signal be active only for the duration of message transmission. (Some RS-232 to RS-485 converters utilize this approach so determine their communication methodology before purchasing these devices.)

- Bias resistors provide "high" data signal
- "Low" data signal enables transmitter
- Doesn't adhere to RS-485 specifications

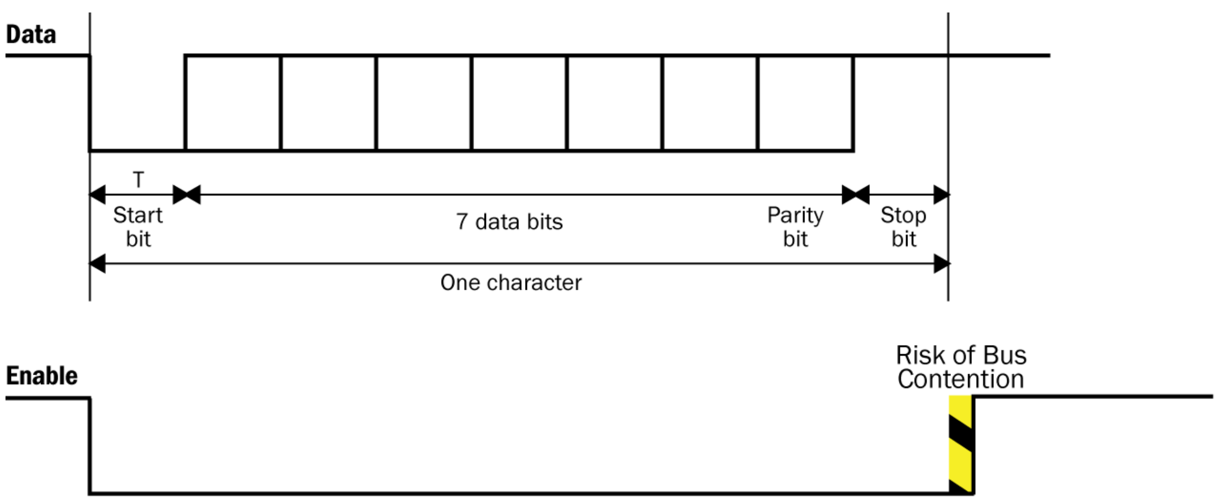


## D. Automatic Driver Control

Rating: ★★★★☆

This advance in RS-485 transmission control uses asynchronous serial driver software that automatically recognizes commands to transmit data from an RS-485 port, enables the transmitter and immediately disables the transmitter at the end of the message. This method is more efficient than control from the application program and does not require the user to design the enable control into the application program.

- Does not require control from the user program
- Much faster timing through Windows
- Bus contention is unlikely



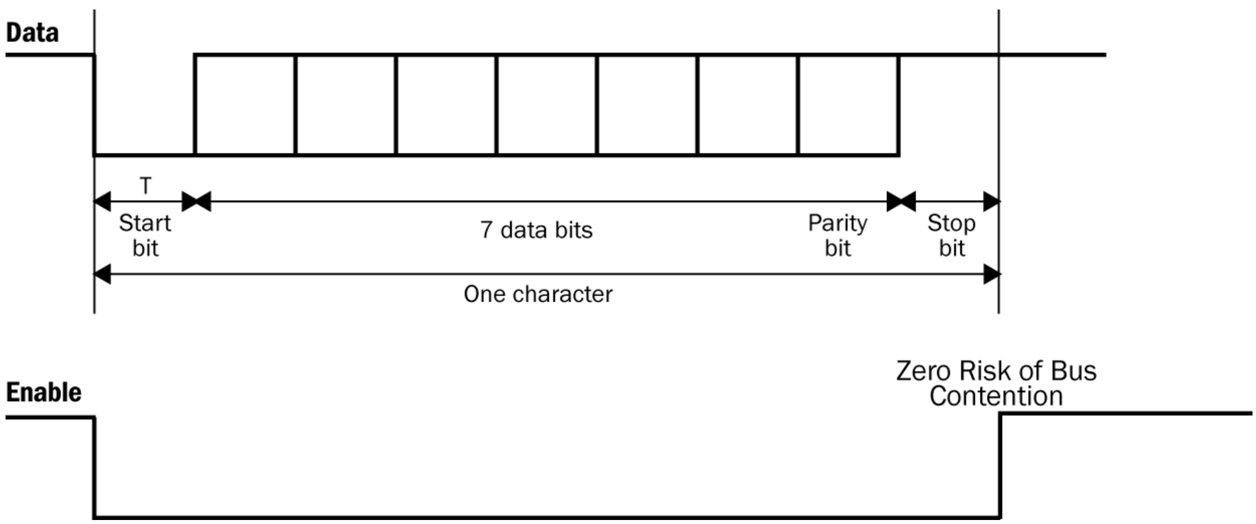

This new approach in RS-485 transmission control was developed by Sealevel Systems and uses the company's SeaCOM asynchronous serial driver software.

## E. Automatic RS-485 Hardware Enable

Rating: ★★★★★

This latest approach is by far the easiest to use and eliminates any risk of bus contention and data corruption. This method uses hardware devices that are auto-enabled to control the RS-485 communications. Products with the auto-enable circuit do not require software control of the transceivers enable pin. Instead, hardware on the board enables the transmission line and then immediately disables the signal after the message is sent. Another method of hardware automatic driver enable is implemented through the use of certain advanced UARTs (e.g., 16C850) that have this function built right into the chip. These UARTs must be properly initialized to enable this feature.

- Application software control not required
- Software driver support is eliminated
- Auto-enable circuit keeps RS-485 signal active for the minimum required time
- Eliminates risk of bus contention




This latest advancement is another innovation from Sealevel Systems.

## III. Conclusion

System Designers need to be aware of the risks created by the various methods of controlling RS-485 communications. There is a higher risk associated with products that use outdated communication controls. (Often these are the lower priced products because they employ less sophisticated technology.)

The latest RS-485 communication control technology pioneered by Sealevel Systems uses auto-enabled hardware that eliminates the risk of bus contention and data corruption.