

APPLICATION NOTE 723

Selecting and Using RS-232, RS-422, and RS-485 Serial Data Standards

Three common serial data standards, RS-232, RS-422, and RS-485, are described by specification and electrical interface. Cable termination techniques, use of multiple loads, daisy-chaining of RS-232, conversion of RS-232 to RS-485, conversion of RS-485 to RS-232, and RS-232 port-powered RS-485 conversions are described.

Introduction

"The great thing about standards is there are so many to choose from." This statement was made at a recent conference on fiber optics, and it holds true for electrical-interface standards as well. As serial-data standards tend to evolve separately within particular industries, we thus have more standards than we should.

Perhaps the most successful serial-data standard for PC and telecom applications is the RS-232. Similarly, the RS-485 and RS-422 are among the most successful standards for industrial applications. These standards are not directly compatible. For control and instrumentation applications, however, it is often necessary to communicate between the standards. This article discusses the different standards (electrical physical-layer specifications), explains how to convert from one standard to another standard, and demonstrates how to combine different standards within the same application.

RS-232 Electrical Specifications and a Typical Connection

The RS-232 link was initially intended to support modem and printer applications on IBM PCs, however, it now enables a variety of peripherals to communicate with PCs. The RS-232 standard was defined as a single-ended standard for increasing serial-communication distances at low baud rates (<20kbps). Over the years the standard changed to accommodate faster drivers like the MAX3225E, which offers 1Mbps data-rate capability. For RS-232 compliance, a transceiver such as the MAX3225E must meet the electrical specifications listed in **Table 1**. A typical connection (**Figure 1**) shows the use of hardware handshaking to control the flow of data.

Table 1. RS-232 Summary of Major Electrical Specifications

Parameter	Conditions	Min	Max	Units
Driver Output Voltage, Open Circuit			25	V
Driver Output Voltage, Loaded	$3k\Omega < R_L < 7k\Omega$	± 5	± 15	V
Driver Output Resistance, Power Off	$-2V < V < 2V$		300	
Slew Rate		4	30	V/ μ s
Maximum Load Capacitance			2500	pF
Receiver Input Resistance		3	7	k Ω
Receiver Input Threshold:				
Output = Mark (Logic 1)		-3		V

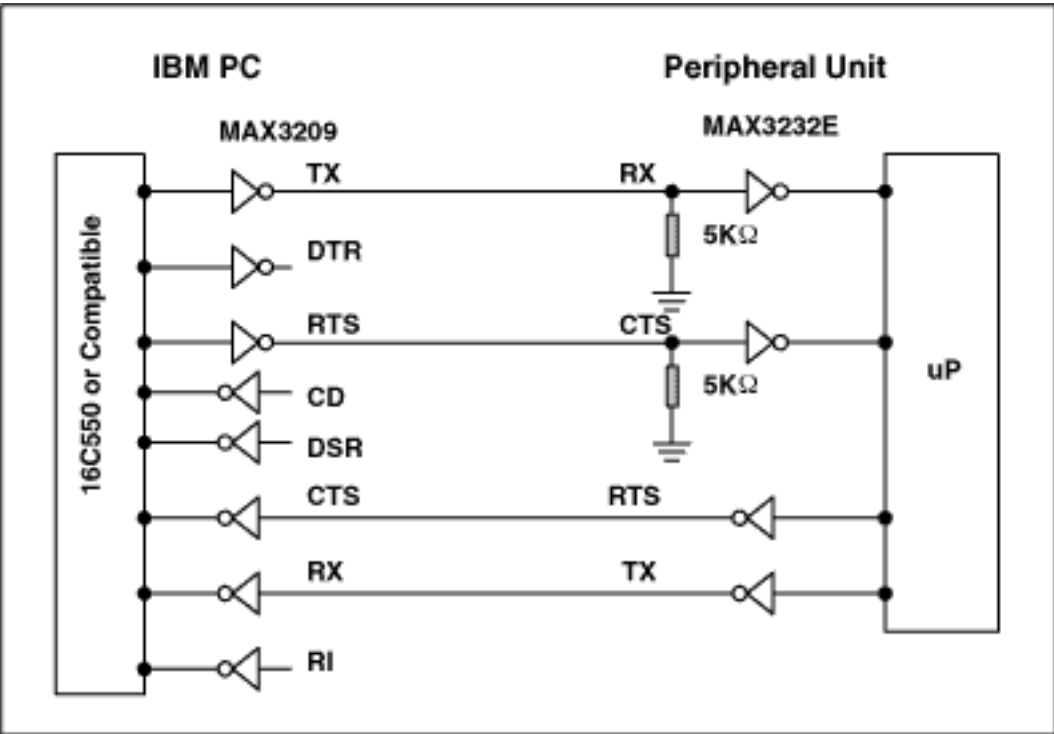


Figure 1. Typical RS-232 connection.

A typical RS-232 signal (**Figure 2**, CH1) swings positive and negative. Note the relative location of the 0V trace markers on the left axis. Although the RS-232 data is inverted, an overall translation from TTL/CMOS to RS-232 and back to TTL/CMOS restores the data's original polarity. Typical RS-232 transmissions seldom exceed 100 feet for two reasons. Firstly, the difference between transmitted levels ($\pm 5V$) and receive levels ($\pm 3V$) allows only 2V of common-mode rejection. Secondly, the distributed capacitance of a longer cable can degrade slew rates by exceeding the maximum specified load (2500pF). Because the RS-232 was designed as a point-to-point rather than multidrop interface, its drivers are specified for single loads from $3k\Omega$ to $7k\Omega$. Therefore, a daisy-chain scheme is typically implemented for multidrop interface applications (**Figure 3**).



Figure 2. An RS-232 receiver accepts the bipolar input signal (top trace, CH1) and outputs an inverted TTL/

CMOS signal (bottom trace, CH2).

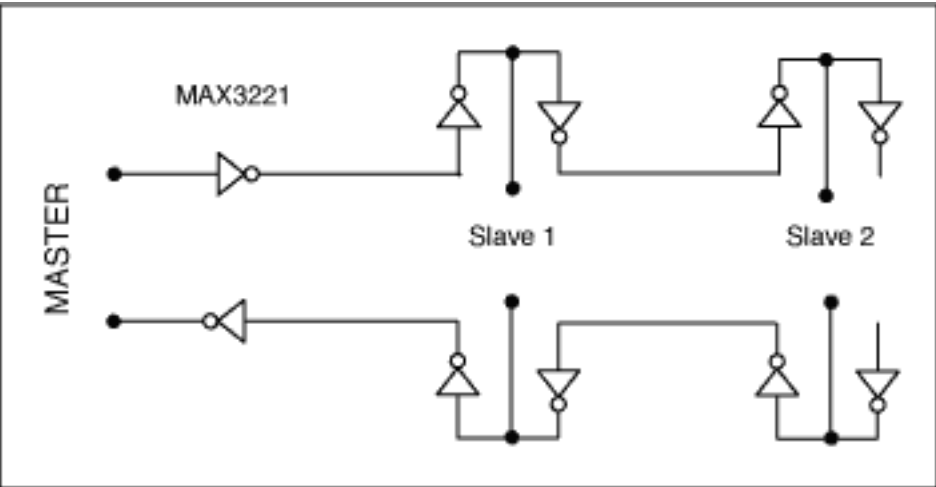


Figure 3. Daisy-chaining allows multiple slave transceivers on a single RS-232 line.

Daisy-Chain Devices and Their Limitations

In a daisy-chain configuration, the RS-232 signal enters through one receiver and is looped through to a transmitter. This configuration is repeated for subsequent devices along the data transmission line. Cable breaks are a major problem with this technique. A break between slave 1 and slave 2 prevents all downstream devices from transmitting or receiving data. Other multidrop RS-232 techniques involve prebuffering or boosting the RS-232 output drive (enabling it to drive multiple 5kΩ inputs in parallel).

To eliminate the problems associated daisy-chain networks, Maxim developed the MAX3322E/MAX3323E, specifically designed to be configured in multidrop applications. These unique devices employ a logically switched input resistance of 5k . When a device is not selected, its input resistance remains in a high impedance state allowing communication to proceed with other devices along the shared bus. Another solution to the daisy-chain network problem is to convert the RS-232 Rx and Tx signals to RS-422 signals (see **Table 2**). RS-422 is a differential standard that allows transmission over much greater distances. The higher input resistance of RS-422 inputs, combined with their higher drive capability, allows a connection of up to ten nodes (**Figure 4**). Another RS-422 advantage is the separate transmit and receive paths for which no direction control is needed. Any necessary handshaking between devices can be performed with either software (XON/OFF handshaking) or hardware (a separate set of twisted pairs). The MAX3162 provides an economical way to translate between RS-232 and RS-422 signals. For more detail about this process, refer to the section below entitled *RS-232/RS-485 Protocol Translators*.

Table 2. RS-422 Summary of Key Specifications

Parameter	Conditions	Min	Max	Units
Driver Output Voltage, Open Circuit			±10	V
Driver Output Voltage, Loaded	R _L = 100Ω	2 -2		V
Driver Output Resistance	A to B		100	Ω
Driver Output Short-Circuit Current	Per output to common		150	mA
Driver Output Rise Time	R _L = 100Ω		10	% of bit width
Driver Common-Mode Voltage	R _L = 100Ω		±3	V
Receiver Sensitivity	V _{CM} ≤ ±7V		±200	mV
Receiver Common-Mode Voltage Range		-7	7	V

Receiver Input Resistance		4		k Ω
Differential Receiver Voltage	Operational		± 10	V
	Withstand		± 12	V

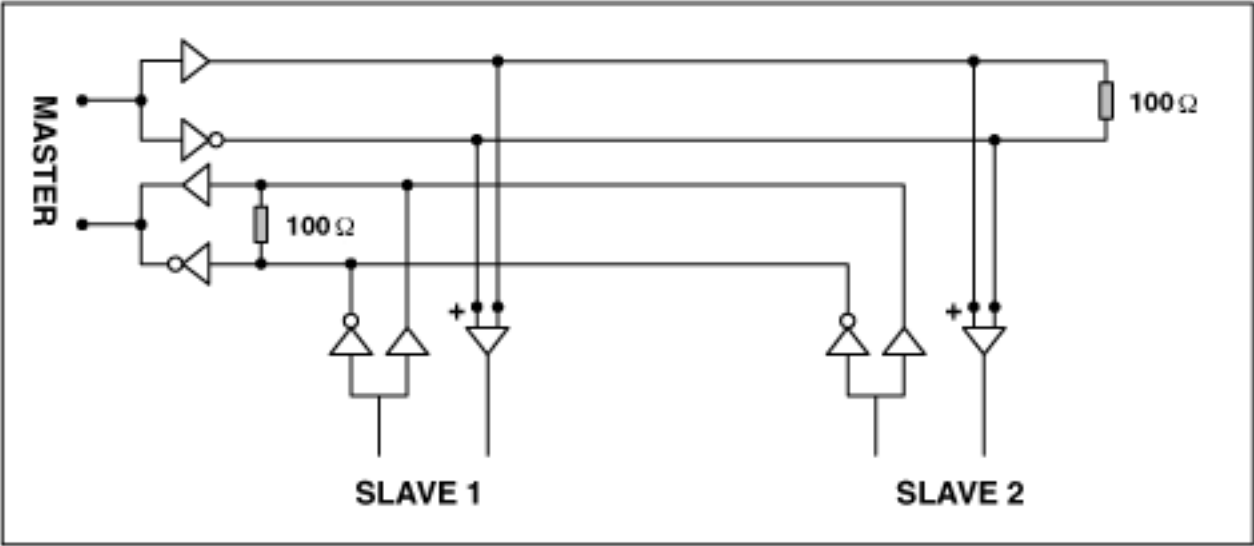


Figure 4. A typical RS-422 system allows as many as ten slave transceivers on the differential transmission line.

Differences Between RS-485 and RS-422 and Their Use in Applications

RS-422 and RS-485 transceivers are often confused with each other; one is assumed to be a full-duplex version of the other. The electrical differences, however, in their common-mode ranges and receiver-input resistances make these standards suitable for different applications. As RS-485 meets all the RS-422 specifications (**Table 3**), RS-485 drivers can be used in RS-422 applications. The opposite, however, is not true. The common-mode output range for RS-485 drivers is -7V to +12V, whereas the common-mode range for RS-422 drivers is only ± 3 V. The minimum receiver-input resistance is 4k Ω for RS-422 drivers and 12k Ω for RS-485 drivers.

Table 3. RS-485 Summary of Key Specifications

Parameter	Conditions	Min	Max	Units
Driver Output Voltage, Open Circuit		1.5	6	V
		-1.5	-6	V
Driver Output Voltage, Loaded	$R_L = 100\Omega$	1.5	5	V
		-1.5	-5	V
Driver Output Short-Circuit Current	Per output to common		± 250	mA
Driver Output Rise Time	$R_L = 54\Omega$ $C_L = 50\text{pF}$		30	% of bit width
Driver Common-Mode Voltage	$R_L = 54\Omega$		± 3	V
Receiver Sensitivity	$-7\text{V} < V_{CM} < 12\text{V}$		± 200	mV
Receiver Common-Mode Voltage Range		-7	12	V
Receiver Input Resistance		12		k Ω

To reduce wiring expense and achieve longer line lengths, RS-485 transceivers have become a popular standard for use in point-of-sale, industrial, and telecom applications. The wider common-mode range of RS-485 also enables longer line lengths and a higher input resistance per node, allowing more nodes to be connected to the bus (**Figure 5**).

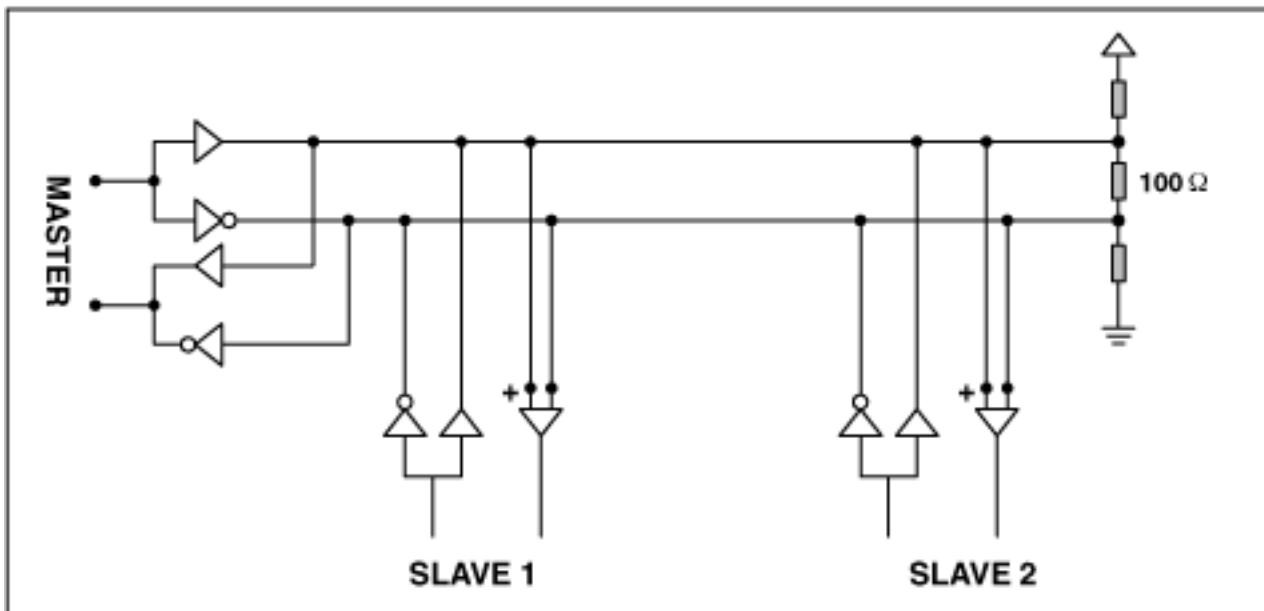


Figure 5. Compared with RS-422, the higher input impedance and wider common-mode range of an RS-485 connection enables longer line lengths.

Differential RS-485 transmissions (**Figure 6**) produce opposing currents and magnetic fields along each segment (wire) of a twisted-pair cable, thus minimizing the emitted electromagnetic interference (EMI) by cross-canceling the opposing fields around each wire. For transmissions through a long cable or at high data rates, the cable appears as a transmission line and should be terminated with the cable's characteristic impedance. This aspect of the RS-485 connection causes confusion. Does the line need to be terminated, and if so, how should it be terminated? If the designer is not the end user, should these questions be left for the installer to resolve? For most RS-485 transceivers, the data sheet indicates a simple choice between no termination and a simple point-to-point termination when the cable acts as a transmission line (**Figure 7**). A termination resistor across the A-B terminals is harmless. By default, the transmission line should be terminated at the last transceiver on the line (bus).

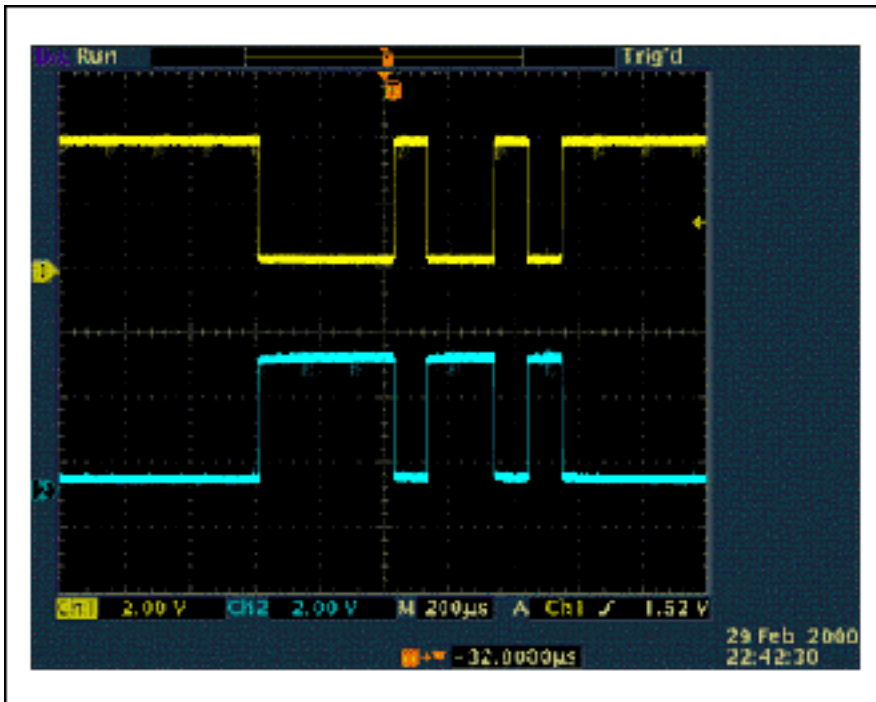


Figure 6. The opposite polarity signals on an RS-485 line minimize EMI by cross-canceling each other's respective magnetic fields. The GND references on the above scope photo have been shifted (offset) to clearly show the inverted polarities of the RS-485 output signals.

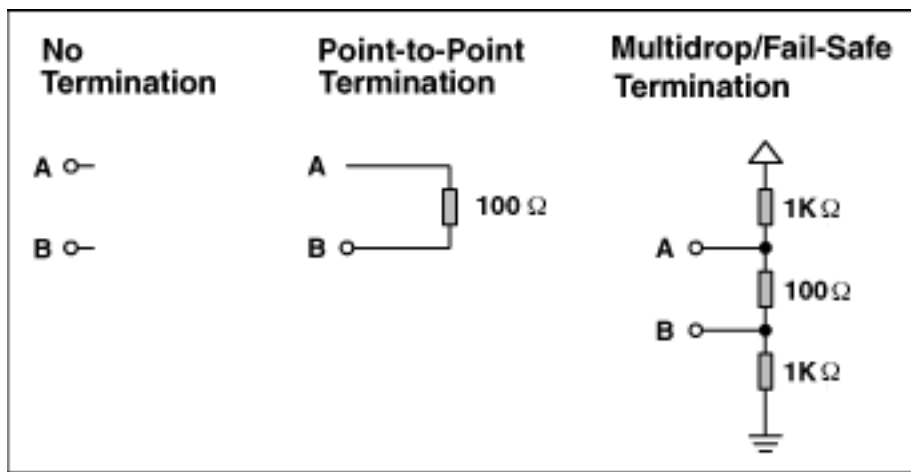


Figure 7. The choice of termination resistors for a transmission line depends on the application.

Fail-Safe

Deciding whether you need a termination resistor or not is only part of the problem in implementing an RS-485 system. Normally, an RS-485 receiver output is "1" if $A > B$ by +200mV or more, and "0" if $B > A$ by 200mV or more. In a half-duplex RS-485 network, the master transceiver tri-states the bus after transmitting a message to the slaves. Then, with no signal driving the bus, the receiver's output state is undefined, as the difference between A and B tends towards 0V. If the receiver output, RO, is "0," the slaves interpret it as a new start bit and attempt to read the following byte. The result is a framing error because the stop bit never occurs. The bus goes unclaimed, and the network stalls.

Unfortunately, different runs of chips can produce different output signals on RO for a 0V differential input. The prototype can work perfectly, however, certain nodes will fail in a later production run. To solve this problem, bias the bus as shown in Figure 7 under Multidrop/Fail-Safe Termination. Biasing the bus ensures that the receiver output remains "1" when the bus is tri-stated. Alternatively, you can use "true fail-safe" receivers like those of the MAX3080 (5V) and MAX3070 (3V) families. These devices ensure an RO output of "1" in response to a 0V differential input by changing the receiver's threshold to -50mV.

RS-232/RS-485 Protocol Translators

The MAX3162 is an unique device for it contains both RS-232 and RS-485 receivers and transmitters. This wide range of communication devices contained within a single IC enables an individual to convert bidirectionally between RS-232 and RS-485 signals. The circuit in **Figure 8** illustrates the MAX3162 configured to bidirectionally convert RS-232 and RS-485 signals in a point-to-point application.

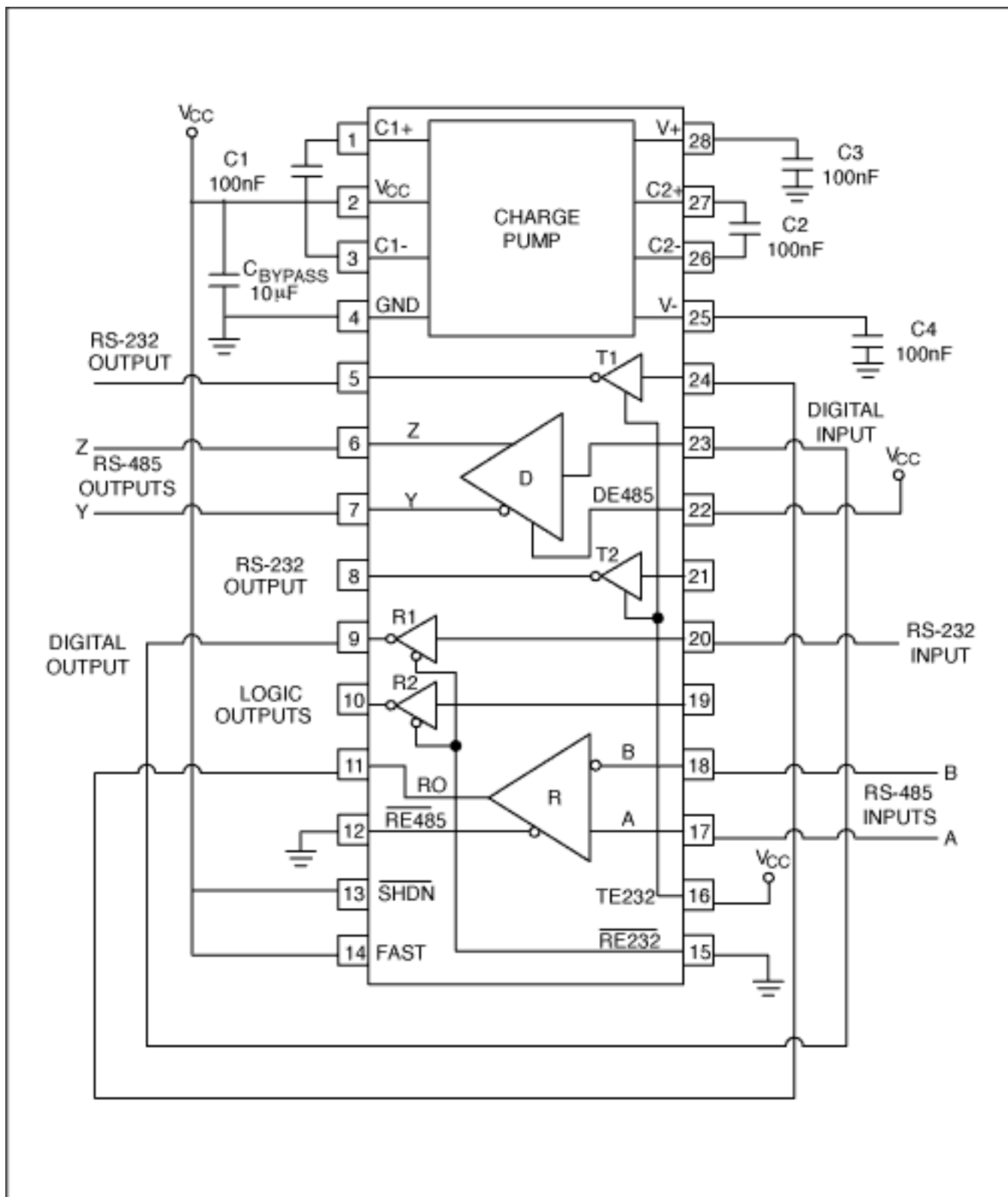


Figure 8. The MAX3162 converts bidirectionally between RS-232 and RS-485 signals in a point-to-point application.

Figure 9 shows the MAX3162 configured as an RS-232/RS-485 multipoint protocol translator. The direction of translation is controlled through the RTS signal, R1IN. The single-ended RS-232 receiver-input signal is translated to a differential RS-485 transmitter output. Similarly, a differential RS-485 receiver-input signal is translated to a single-ended RS-232 transmitter output. RS-232 data received on R2IN is transmitted as an RS-485 signal on Z and Y. RS-485 signals received on A and B are transmitted as an RS-232 signal on T1OUT.

The RTS line is a common means for controlling bus direction in circuits that convert from RS-232 to RS-485.

This line on the RS-232 port controls whether the RS-485 transceiver acts as a transmitter or a receiver (Figure 9). Note that the system cannot be sure that a byte of data in the UART's transmit buffer has been transmitted unless the system monitors the RS-485 driver input, DI. That is, the system must either allow a fixed time delay or actively monitor the DI input before using the DE input to change the bus direction. Other direction-control techniques include using a microcontroller and driving the DE input with data while pulling the A-B lines apart (connecting a pull-up resistor from A to 5V and connecting a pull-down resistor from B to ground). The value of these resistors varies with cable capacitance, but is typically $1k\Omega$.

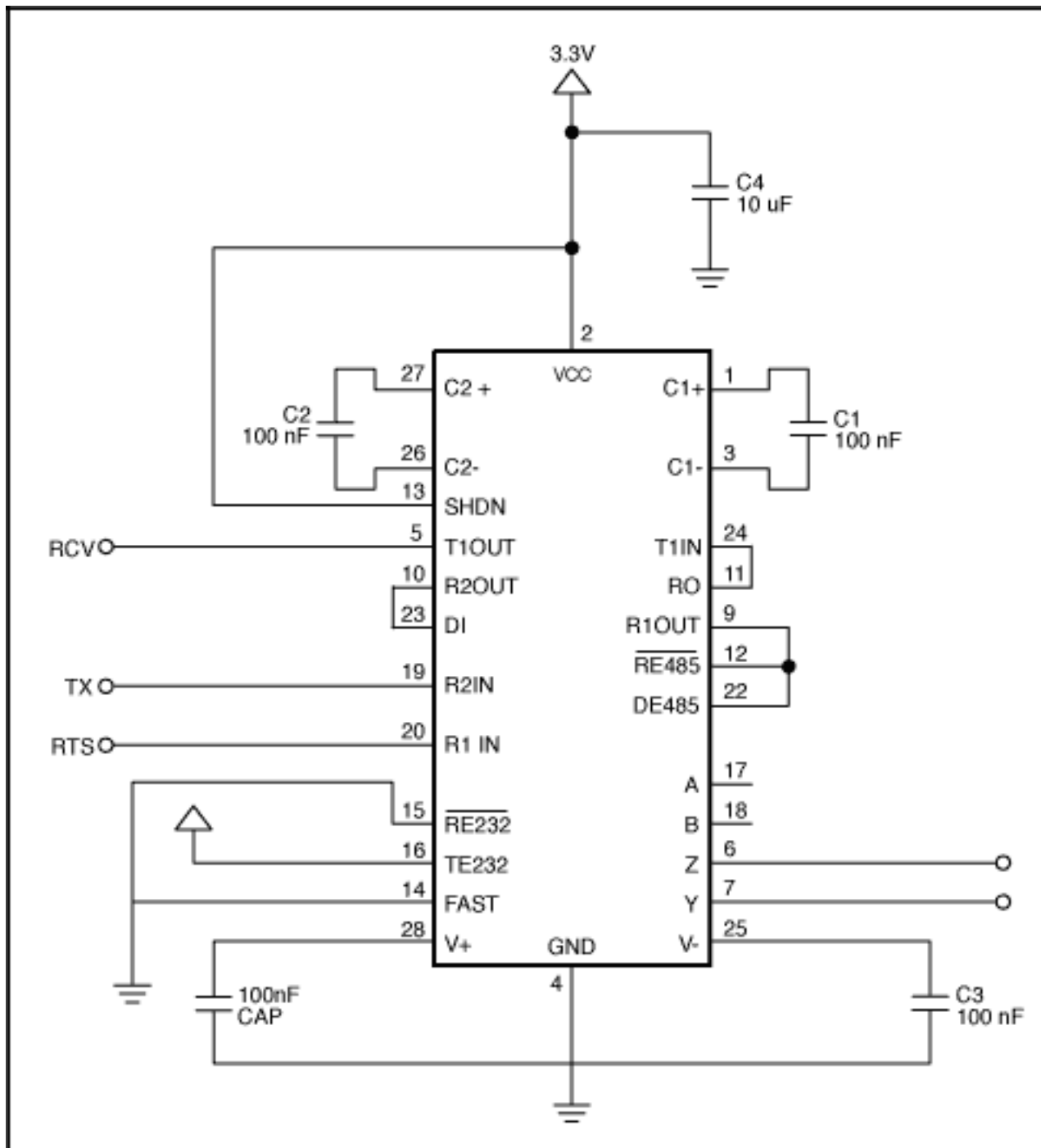


Figure 9. The MAX3162 converts bidirectionally between RS-232 and RS-485 signals in a multipoint application.

Port-Powered Devices

Many converters from RS-232 to RS-485 are "port-powered converters" in which the RS-485 power is derived from the RS-232 RTS line (or sometimes a combination of the RTS and CTS (DTR) lines). Because the power available from an RS-232 port is limited, the RS-485 launch voltages are not achieved when using a port-

powered converter with, for example, one hundred RS-485 terminations. The low receiver threshold (200mV), however, allows a fair margin for error. This technique is acceptable in systems with short line lengths and without termination resistors across the A-B terminals.

Hot-Swap

When circuit boards are inserted into a hot or powered backplane, differential disturbances to the data bus can lead to data errors. Upon initial circuit-board insertion, the data communication processor undergoes its own power-up sequence. During this period, the processor's logic-output drivers are high impedance and unable to drive the DE and /RE\ inputs of the MAX3060E/MAX3080E to a defined logic level. Leakage currents up to $\pm 10\text{mA}$ from the high-impedance state of the processor's logic drivers could cause standard CMOS enable inputs of a transceiver to drift to an incorrect logic level. Additionally, parasitic circuit board capacitance could cause coupling of VCC or GND to the enable inputs. Without the hot-swap capability, these factors could improperly enable a transceiver's driver or receiver.

References

1. RS-422 and RS-485 Application Note, B&B Electronics (This is a great source of information on RS-232, RS-485 and RS-422 standards and their practical realizations. <http://www.bb-elec.com>)
2. Serial Port Complete, Jan Axelson (This definitive reference for serial ports includes considerable useful source code written in Visual Basic. <http://www.lvr.com>)

Application Note 723: <http://www.maxim-ic.com/an723>

More Information

For technical questions and 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>

Related Parts

MAX232: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

MAX3162: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

MAX3232: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

MAX485: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

AN723, AN 723, APP723, Appnote723, Appnote 723

Copyright © 2005 by Maxim Integrated Products

Additional legal notices: <http://www.maxim-ic.com/legal>