**Advantages of RS485**

RS485 adopts differential signal negative logic. +0.2 V - +6 V is indicated as ‘0’ and -6 V - -0.2 V is indicated as ‘1’. There are two-wire and four-wire connection modes. In most circumstances, when connecting RS485 communication links, only a twisted pair is used to connect the ‘A’ and ‘B’ terminals of each interface. In theory, the maximum communication distance of RS485 can extend up to 1,200 meters. The line drivers in an RS-485 network are designed to drive thirty-two nodes.

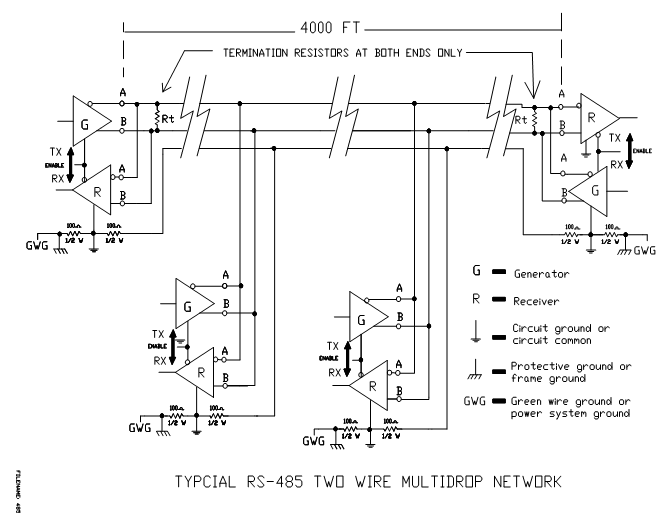
The main advantages of RS485 are:

Logic ‘0’ and ‘1’ are indicated by the voltage differences between the two wires +0.2-6 V and -0.2-6 V respectively. The interface signal level is lower than that of RS-232-C, so the interface circuit chip is not easily damaged. Also the level is compatible with TTL level, which facilitates the connection with a TTL circuit   
Maximum data transmission speed is 10 Mbps.   
The RS485 interface adopts the role of balancing driver and differential receiver so that the common mode disturbance immunity is improved, i.e. it has good noise immunity.   
Maximum 32 transceivers are allowed to connect to the bus. With this multi-station capability, a device network can be conveniently established via a single RS485 interface   
Normally only two wires, a shielded twisted pair (STP), are needed in the half duplex network formed by the RS485 interface.

The main disadvantages are:

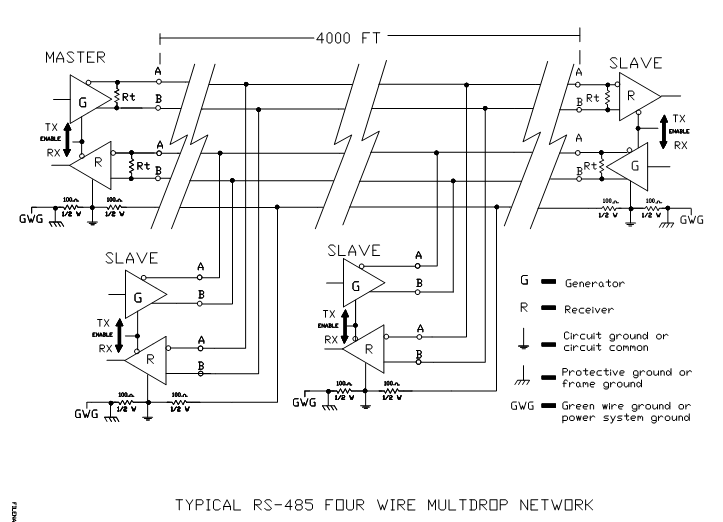
In the whole network only one node can transmit data to the bus at any one moment and all other nodes must be in receiving status. If there are two or more nodes transmitting data to the bus at the same time, all transmission fails.   
A special communication cable is needed. The maximum communication distance depends on communication cable type.

**Two Wire System**



The RS-485 Standard permits a balanced transmission line to be shared in a party line mode. As many as 32 driver/receiver pairs can share a two-wire party line network. Many characteristics of the drivers and receivers are the same as RS-422. The range of the common mode voltage Vcm that the driver and receiver can tolerate is expanded to +12 to -7 volts. Since the driver can be disconnected or tristated from the line, it must withstand this common mode voltage range while in the tristate condition. Some RS-422 drivers, even with tristate capability, will not withstand the full voltage range of +12 to -7 volts. Figure shows a typical two-wire multidrop or party line network. Note that the transmission line is terminated on both ends of the line but not at drop points in the middle of the line. The signal ground line is also recommended in an RS-485 system to keep the common mode voltage that the receiver must accept within the -7 to +12 volt range. This system serves half-duplex communication.

**Four Wire System**



An RS-485 network can also be connected in a four-wire mode. In a four-wire network it is necessary that one node be a master node and all other be slaves. The network is connected so that the master node communicates to all slave nodes. All slave nodes communicate only with the master node. This network has some advantages with equipment with mixed protocol communications. Since the slave nodes never listen to another slave response to the master, a slave node cannot reply incorrectly to another slave node. This system serves full-duplex communication.

**485 interface circuit with Auto Switching Link**

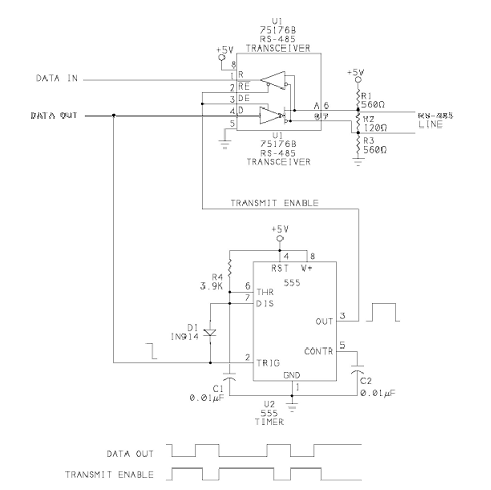
One of challenges in designing an RS-485 link is controlling the driver- enable lines. Because all of the nodes share a data path, only one driver can be enabled at a time. Before transmitting, a driver must be sure that the previous driver has been disabled.

Many RS-485 networks use a command/response protocol, where one node sends commands, and the node being addressed returns a response.

The UART in the node being addressed detects the final Stop bit in the middle of the bit width, or slightly sooner or later if the sender's clock doesn't match exactly. A very fast node may be ready to send a reply within a few microseconds after detecting the Stop bit. To prevent the need for a delay before responding, the sending node's driver should be disabled as soon as possible after the leading edge of its final Stop bit. In most systems, the transmitting driver is enabled on the leading edge of the Start bit and remains enabled for the entire transmission, then is disabled as soon as possible after the final Stop bit. In the delays between transmissions, biasing holds the line in an idle state.

There are various ways that the transmitting node can determine when a transmission has finished and it's safe to disable the driver. The node may read back what it sent, or it may use a hardware or software timer to estimate the time needed to transmit.

This circuit shows a completely automatic way to control the enable line so that the driver is disabled as quickly as possible, soon after the leading edge of the Stop bit:



With this circuit, the program code doesn't have to toggle a signal to enable and disable the driver, and a transmitting driver doesn't need to allow extra time to be sure that the previous driver has been disabled. Unlike other methods of automatic control, there are no jumpers to set for a particular bit rate.

Instead of keeping the transmitter enabled for the entire transmission, the circuit above enables the driver on the leading edge of the Start bit or any logic low at the driver's input, and it disables the driver about 40 microseconds after the leading edge of the Stop bit or any logic high at the driver's input. When the driver is disabled, biasing resistors ensure that the receiver's output is a logic high.

The delay is generated by a 555 timer configured as a monostable (one-shot). The enable inputs of the driver and receiver are tied together, so the receiver is disabled when the driver transmits.

The timer's output controls the transceiver's enable inputs. A falling edge at Data Out indicates a Start bit and triggers the timer. The timer's output

goes high, enabling the driver and bringing line B more positive than line A. Diode feedback to the Trig input holds the timer's output high for as long as Trig remains low.

When Data Out goes high, the RS-485 line switches, bringing line A more positive than line B. The same logic high also causes the timer to begin timing out. About 40 microseconds after the rising edge, the timer's output goes low, disabling the driver. The delay ensures that the driver's RS-485 output switches without delay, while the driver is enabled. When the driver is disabled, the biasing components continue to hold A more positive than B.

In a similar way, any falling edges in the transmitted data enable the driver, and any rising edges disable the driver after the delay. On the final Stop bit, the driver is disabled no later than 40 microseconds after the Stop bit's leading edge.

At rates of 9600 bps or less, the bit width is greater than 100 microseconds, which means that the driver is disabled at around the middle of the bit width. At faster bit rates, the driver will still be disabled no more than 40 microseconds after the Stop bit's leading edge. For networks that need very fast response time at faster bit rates, decrease R4 for a shorter delay.

A downside to this circuit is that the final voltage for logic zeros is the biasing voltage, which is usually less than the differential voltage when the driver is enabled. But since the biasing voltage needs to be great enough to prevent errors due to noise on an idle line, it should do the job for active logic states as well.