Weeder Technologies

AN100

90-A Beal Pkwy NW, Fort Walton Beach, FL 32548 www.weedtech.com 850-863-5723

Stackable RS-232 Communications

Author: Terry J. Weeder

INTRODUCTION

The stackable RS-232 data modules from Weeder Technologies incorporate a special communications port that allows them to be plugged end to end on a normal RS-232 serial cable. Since RS-232 is not meant for multi-drop access, certain key functions necessary for networking had to be performed external from the bus. The stackable data modules incorporate a separate transmitter and receiver for each RS-232 data line so that the bus can be converted to a multi-drop configuration in-between. The MCU can then manipulate the bus in a fashion common to multi-drop networks.

PHYSICAL LAYER

A standard RS-232 communications link uses two separate lines for two-way communications. On a DB9 RS-232 connector, pin 3 carries data from the DTE port of a host to the DCE port of a peripheral. Pin 2 carries data from the peripheral back to the host. Notice the discrepancies of the two types of ports. A DTE port is meant to plug into a DCE port and vice versa. You can not plug two of the same type of ports together or both devices will be transmitting on the same line head to head, as well as trying to receive on the same line. A null modem adapter must be used if wishing to connect a DTE to another DTE, or a DCE to another DCE port.

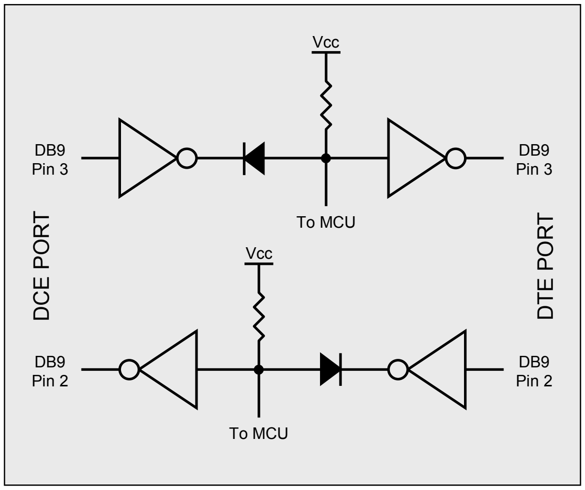
The RS-232 standard uses a +3 to +25 volt level to represent a logic 0, and a -3 to -25 volt level to represent a logic 1. Both these logic states are defined as dominant. In other words, a voltage level is held firm by the output driver sourcing it, and should not be acted upon by any other driver.

In order to use an RS-232 data link in multi-drop mode, it is necessary to convert the hardware configuration over to a dominant/recessive logic state backbone as used in common multi-drop data busses such as I^2C, CBUS, CAN, and other open-drain types. To do this, the data is passed through an RS-232 receiver/transmitter pair as shown in Figure 1 which extracts the data for bus hardware re-formatting. Pull-up resistors and diodes are used to create the tri-state style buffer in the data path which is needed for the recessive state of the bus as seen by the MCU. This buffer provides the means for bus sharing and collision contention throughout the network.

COLLISION CONTENTION

The utilization of the communications line can be thought of more as a single, bi-directional, data bus, operated in a multi-drop mode rather then a normal RS-232 data link. A transmission from a data module travels in both directions, upstream to the host, and downstream to signal other modules that it has seized the line. Before transmitting, a module will listen to the communications line and wait for quiescence. After a silent period equal to the length of one byte, the waiting module will send its data packet.

FIGURE 1: HARDWARE INTERFACE



BUS ARBITRATION

The stackable data modules use a communications stratagem based on the Carrier Sense Multiple Access (CSMA/CD) protocol. Carrier Sense (CS) is the monitoring of the data bus for a period of inactivity before a module is allowed to begin its own transmission. Multiple Access (MA) means that once the bus is free, every module in the network has an equal opportunity to transmit a frame. And Collision Detection (CD) is used to assure the integrity of a data frame when two or more modules try to transmit at the same time.

When sending a frame, arbitration occurs during the transmission of the header character. The stackable data modules use a non-destructive bit wise arbitration that allows messages to remain intact even if a collision is detected. Furthermore, there is no delay introduced to the message that wins arbitration because its data has not been effected by the collision, and thus can continue transmitting without interruption.

For example, during a transmission, a logic 0 is generated by pulling the data lines to their opposite state (dominant bit). A logic 1, by releasing them (recessive bit). Because a logic 1 requires no forced output, a module can listen during this time for a collision. In cases when two modules begin transmitting at the exact same time, the first module to exclusively send a logic 1, will detect the logic 0 of the second module and immediately surrender arbitration to it without corrupting its data. Once the communications line is again free, the surrendered module will re-transmit.

PROPAGATION DELAYS

Each module in the network introduces a small amount of propagation delay to the data signals as they pass through in both directions. With a half-dozen or less modules plugged end to end, this propagation delay is negligible. However, as the number of modules in the chain increases, so does the overall propagation delay from one end of the chain to the other. Eventually the delay becomes a factor in the communications integrity. If for instance, there are 20 or 30 modules in the network and two modules at opposite ends start to transmit at the same time, there will be a delay before the down-stream module's start bit reaches the up-stream module. And because the leading edge of the start bit generated by the up-stream module is already on its way to the host, it is imperative that the down-stream module does not win arbitration of the line because its data will not be in sync with this start bit.

Therefore, while transmitting a start bit, a module will release the down-stream data line prematurely in order to listen for a start bit coming from an up-stream module. And thanks to the propagation delay, even though the up-stream module will be stopping to listen also, the time lag allows the down-stream module to still detect the presence of a foreign start bit and in turn surrender the line to it. This all happens before the up-stream module even finishes transmitting the start bit to the host, thus no data is lost or corrupted.

Of course, the data traveling in the opposite direction also inherits this same start-bit misalignment error by the time it reaches the down-stream module. To prevent this data from passing through, which could become corrupted due to the error, the down-stream module will hold the down-stream data line at a logic 0 for the remainder of the first byte. This will alert any subsequent modules further down the line to ignore the data packet.

HOST PC CONSIDERATIONS

Unfortunately, the non-destructive bit wise arbitration mentioned earlier is not supported by the host PC’s communications port. And because the PC’s BIOS does not generate an interrupt until after receiving one full byte, there is a 1 mS window at the beginning of a module’s transmission where the host is blind, and may inadvertently begin its own transmission. Should this occur, the current transmitting module along with those modules downstream from it will miss the data packet sent by the host.

Due to the speed of the communications line, and leanness of the data packets, the probability of this event is rare and should not pose a problem in most applications. Furthermore, since all members of the command set transmitted by the host evoke some form of response from the module it is talking to, it is easy for the host to determine if this has in fact occurred, and thus re-transmit the missed data packet.