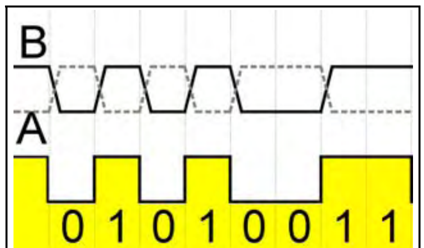
Example Applications: RS-485 Two-Wire

RobustDC Application Note #3

Quick Index:

• BRIEF OVERVIEW OF EIA/RS-485

RS-485 is the most common, open standard for multi-drop industrial data communications today. Isolated RS-485 repeaters can be used to: 1) extend the over-all distance, 2) increase the RS-485 device count, 3) increase the robustness, and 4) convert to a star or tree topology (wiring layout). Robust DataComm Pte Ltd has developed the world's most effective isolated RS-485 repeaters which gives you all this and greatly reduces your system down-time.



ASCII "I"

EIA/RS-485 is called a "balanced differential" signal. It uses twisted wire pairs to transmit data by a differential voltage signal. ***The two wires in a pair are not a loop*** -- both are '+' signals sourcing current to a third "virtual" ground conductor. For example, here is the differential signal for an ASCII character 'I'. EIA labels the signals A and B, while many vendors label them '+' and '--'. Data is represented by the relative voltage of A to B. When $V_A < V_B$, then the data is a binary 1 (or Mark or Off). When $V_A > V_B$, then the data is a binary 0 (or Space or On). An idle asynchronous line without data will normally be in the binary 1 state.

RS-485 gains it's noise immunity from the nature of electrical noise on a wire pair. A noise spike picked up by one wire will induce an equal noise on the other wire. Since an RS-485 receiver is just comparing the relative voltage polarity (ie: which is more positive) even during the noise spike the transmitted relationship will hold true. For example, if A is 2vdc and B is 4vdc, then a common noise spike of 3 volts will make them 5 and 7vdc respectively - but the relationship A < B still holds true.

• BASIC TWO-WIRE MULTI-DROP NETWORK

Figure 1 below shows a basic five device, 2-wire RS-485 network. It consists of 1 Master device (MD) and 4 Slave Devices (SA, SB, SC, & SD). It also includes 3 galvanically isolated RS-232 to RS-485 converters (MDiC, SAiC, & SDiC). Slave devices SB and SC connect directly to the 2-wire RS-485 network. Since generally RS-485 networks include only devices from 1 vendor, this example is not very realistic, but it shows the various configurations possible.

- **THE 2-WIRE RS-485 BUS**

The RS-485 bus runs from MDiC to SDiC. Notice the 120ohm termination resister installed at each end. The ISO-8482 standard (ISO version of RS-485) defines limitations for RS-485 multipoint buses. The overall bus length is limited to 500m at a speed of 1Mbps. MD and SD define the ends of the bus. SA, SB, and SC are connected to the bus by *stubs*. Stubs are limited to 15m. Any surge protection on the bus should clamp beyond ±25vdc. Current limiting fuses should be 250mA.

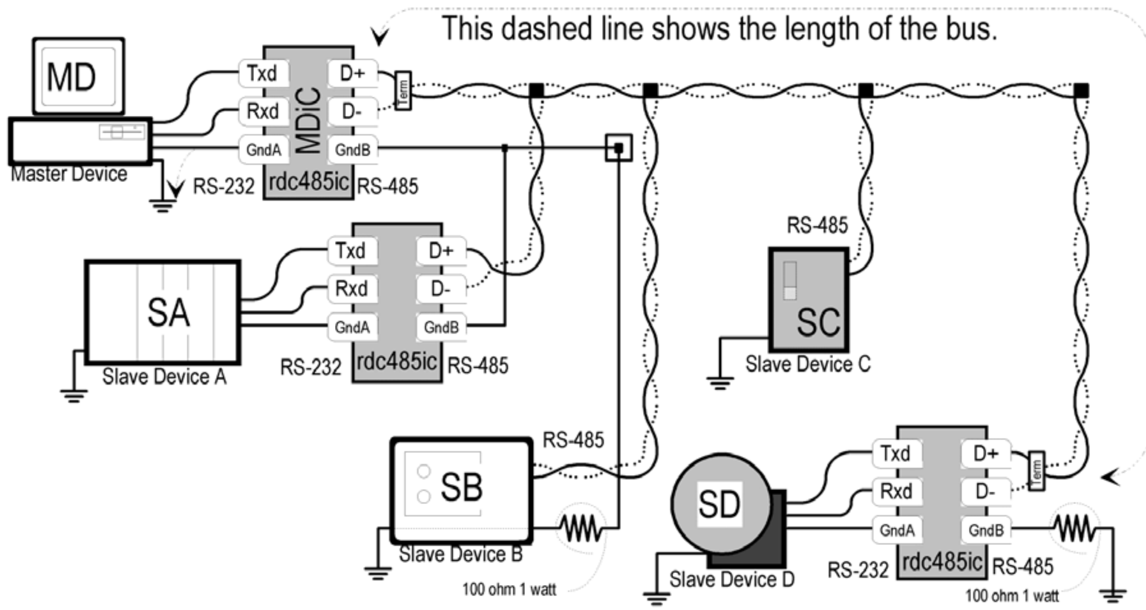


Figure 1: Example 2-wire RS-485 Network

Standards by nature are conservative. While ISO says only 500m, RS-485 product vendors claims distances from 1km up to 13km. One reason for the discrepancy is the effect of speed (baud rate or bps) on the maximum bus length. Slower data speeds allow longer workable bus lengths. The most common RS-485 claim is 1000m. Longer distances can be covered if you are willing to lower baud rates if problems develop on site.

- MASTER DEVICE (MD)

MD is a standard office-grade PC with an RS-232 port. It connects to the RS-485 bus by an isolated RS-232 to RS-485 converter (MDiC). The RS-485 bus direction control is automatic, so MD will *receive* from the bus until it has something to transmit. All slave devices will also be receiving from the bus. Since no device is transmitting, the bus will be "floating" -- and floating wires are bad. They are very susceptible to noise, which may cause false communication interrupts on devices. Therefore, to pull the bus into a known state bias resisters in MDiC are used. When idle, the voltage on wire D+ is greater than the voltage on D- (D+ > D-). Voltages vary, but maybe D+ = 2.6v and D- = 2.2v.

MD is grounded per office equipment standards. Its RS-232 signal ground is tied directly to both the chassis ground and the internal digital ground -- a very good reason to galvanically isolate MD from an industrial plant! But don't be fooled by the term "industrial grade" -- most industrial PCs follow the same design. The rdc485ic is an isolated RS-232 to RS-485 2-wire converter with 2Kv galvanic isolation. An rdc485ic can be configured to support both 2 or 4-wire RS-485. RS-485 direction control is automatic.

- SLAVE DEVICE A (SA)

SA also has an RS-232 port and uses an isolated RS-232 to RS-485 converter (SAiC) to connect to the RS-485 bus. Like MDiC, it has automatic direction control. The rdc485ic is an isolated RS-232 to RS-485 2-wire converter with 2Kv galvanic isolation. The rdc485ic can be configured to support both 2 or 4-wire RS-485.

- SLAVE DEVICE B (SB)

SB has a direct RS-485 port without galvanic isolation. In this example network, the entire RS-485 network is now grounded directly through SB due to the direct ground connection. SB controls it's RS-485 direction internally, probably directly through an internal RTS signal.

This example shows an optional 100 ohm, 1 watt resister. The RS-485 standard suggests installing this resister any time a potential ground loop is formed. The two RS-485 signal lines (D+ and D-) have a high enough impedance to permit normal operation with a ±7vdc ground potential difference (or common mode voltage). However, ground wires do not and a direct ground connection could cause damaging current flows. In figure 1 as drawn, this resister is not required since both MDiC and SAiC are galvanically isolated with floating grounds and no such current would flow. However, if future maintenance replaces the isolated MDiC with a non-isolated converter, you now would have a direct ground path through your small data communication cable. Even a small 1v ground potential difference could result in a significant current flow between the MD site and SB site.

- SLAVE DEVICE C (SC)

SC also has a direct RS-485 port without galvanic isolation. Unlike SB, SC provides no ground terminal -- therefore none is connected. SC will assume that the RS-485 ground is referenced to its own signal ground ± 7 vdc. If this is not known, install a galvanic isolator.

- SLAVE DEVICE D (SD)

SD has an RS-232 port and connects to the RS-485 bus much like MD. Since SDiC is technically the end of the bus, it also has a bus terminating resister installed. You could enable the SDiC bias resisters to act as a back-up to those in MDiC (ie: if MDiC is powered off, the bus is still pulled to a known idle state.

SDiC is not grounded to the other RS-485 network devices. It's floating ground is tied to the local signal ground, so that SDiC also assumes that the RS-485 bus ground is referenced to its own signal ground ± 7 vdc. If this is not known, then connect SDiC's ground to the bus ground from MDiC, SAiC, and SB

- **RS-485 GROUNDING**

Data communication systems involve connecting multiple "systems" together and therefore careful thought must be given to grounding. It is a common misconception that RS-485 requires only "two wires". ***This is never true.*** RS-485 always requires at least three conductors: 2 signal wires and 1 signal return path. The EIA/RS-485 standard states:

> *"Proper operation of the generator and receiver circuits requires the presence of a signal return path between the circuit grounds of the equipment at each end of the interconnection. The circuit reference may be established by a third conductor connecting the common leads of devices, **or** it may be established by connections in each using equipment to an earth reference."*

Looking at figure 1, you will see that MDiC and SAiC follow the "third conductor" rule, while SC and SDiC follow the "earth reference" rule. SB follows both rules -- in part to ensure that the third conductor is referenced to earth reference to support SC and SDiC.

- **CABLING FOR 2-WIRE RS-485**

If you chose the earth reference grounding rule, then a single twisted pair cable is required.

If you chose the third conductor grounding rule, then 2-wire RS-485 would then require one of the following: 1) 2-pair cable using 1 pair for ground, 2) 1-pair cable with a separate, external ground wire, or 3) 1-pair cable, with the signal reference run down the shield drain wire. While the first two schemes are understandable, the third suggestion will upset some people. Yet some reputable RS-485 vendors do actually suggest using the shield drain wire. Why? Even without a shield, RS-485 is quite robust. The small current involved in a floating data communication signal reference will have little impact on the effectiveness of the shield. Of course, this option *should not be used* if both devices at each end of the cable are locally grounded. Just for reference, here are example cables specs from Belden:

Belden P/N	Pairs	AWG	mm	Shield/drain wire	Imp	Cap
9841	1	24	0.6	Yes, 24AWG	120ohm	42pf/m
9842	2	24	0.6	Yes, 24AWG	120ohm	42pf/m
8132	2	28	0.4	Yes, 28AWG	120ohm	36pf/m

- **CONTROL OF 2-WIRE HALF-DUPLEX TRANSMISSION**

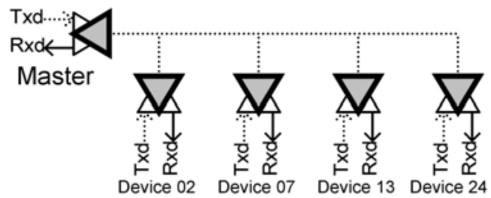
Here is a example of an RS-485 two-wire network in action. Since RS-485 is half-duplex, data can only be transmitted in one direction at a time. Devices must be explicitly *receiving from* OR *transmitting to* the network at any one time, but never both. This change from transmit to receive mode must be activated electronically. The most robust method is to use a direct signal from the device to control the mode. This is often called a Ready To Send (RTS) control signal. Unfortunately, the device's control software must also be programmed to activate and deactivate RTS when appropriate -- many commercial developers plan for full-duplex RS-232 and overlook this. However, if you find yourself without an RTS signal all is not lost. RS-485 converters with automatic duplex control are available.

These detect a start-bit condition on the asynchronous transmit line from the device and automatically switch from receive to transmit mode.

- **PASSIVE DUPLEX CONTROL (AUTOMATIC)**

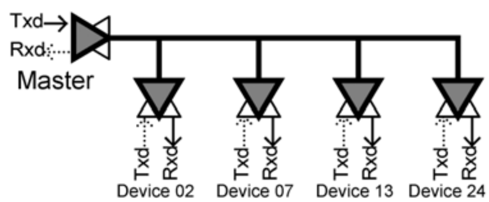
Here is another example of RS-485 two-wire in action. However the devices below use passive duplex control -- also called automatic bus direction control or transmit control. The interface normally receives from the network and when it detects data being sent from the attached device, it automatically "turns around" and transmits the data.

1. No Data to Send -- Line is Idle



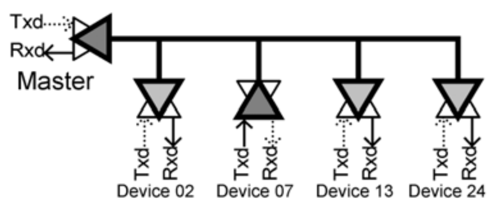
When no message is being sent, the RS-485 two-wire bus is left floating. Special bias resisters must pull the idle wires into the idle state (a binary 1). Each slave waits for a request from the master. No device interface is in transmit mode, all device interfaces are in receive mode.

2. Master Sends a Request



The master device sends a Modbus request; its interface detects the first start-bit and automatically transmits. The slave devices receive the message. When the master device stops sending the request, the interface returns to receive mode. For the moment, all devices are back to receive mode and the bus is floating idle.

3. Device Sends a Response

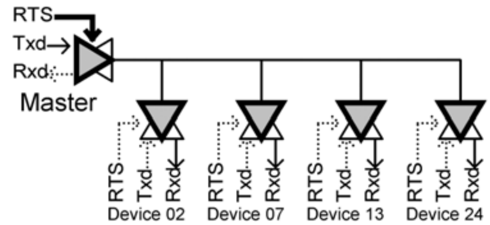


Assuming the addressed device was 07 and no error was detected in the request, then device 07 formats a response and sends it. The interface of device 07 detects the first start-bit and turns to transmit mode. The master device and all other slave devices receive the response. The device 07 interface is in transmit mode, all others are still in receive mode. Once device 07 is finished responding, its interface returns to receive mode and the network is pulled back to idle by the bias resisters.

• ACTIVE DUPLEX CONTROL (WITH RTS)

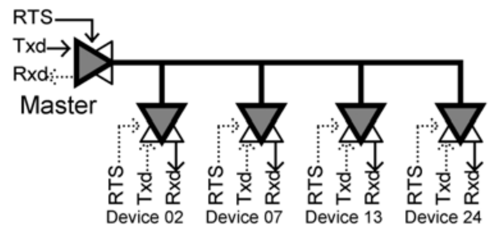
The devices below use active duplex control -- also called direct or software control. For this example the network is using Modbus between a single master device and four slave devices.

1) No Data to Send -- Line is Idle



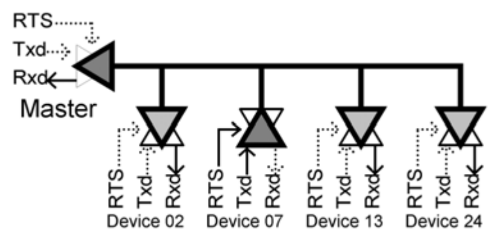
When no message is being sent, the master device activates its RTS control signal, forcing the EIA-485 two-wire bus to the idle state -- a binary 1. Each slave device deactivates its RTS control signal and waits for a request from the master. The master device interface is in transmit mode, all others are in receive mode.

2) Master Sends a Request



The master device sends a Modbus request message. The start-bit of the first byte of the message interrupts all devices on the network, and they start receiving the message byte by byte as it arrives. When completely received, the check sum is verified for possible errors. If an error occurred, the device usually discards the message and sends no response. As soon as the master device completes sending the request, it deactivates its RTS signal. For the moment, all devices are in receive mode.

3) Device Sends a Response



Assuming the addressed device was 07 and no error was detected in the request, then device 07 formats a response. Device 07 activates its RTS control signal and transmits the response. The master device and all other slave devices receive the response -- though other slave devices will likely ignore it. The device 07 interface is in transmit mode, all others are still in receive mode. Once device 07 is finished responding, it deactivates its RTS control signal. Once the master recognizes the end of response, it activates its RTS control and returns its interface to transmit mode to force the network idle.

• FOR MORE INFORMATION

Robust DataComm can truly make your data flow like water - safely, sanely, and silently.

Robust DataComm Pte Ltd
26471 Portola Pkway, Suite 1E-242
Foothill Ranch, CA 92610-1743 U.S.A.
US Sales: (972)248-7466 Fax: (972)248-9533
email : info@robustdc.com

Robust DataComm Pte Ltd
18 Kaki Bukit Road 3 #02-08
Entrepreneur Business Center Singapore 415978
Tel: (65)6487-5624 Fax: (65)6288-3306
web : http//www.robustdc.com