3-way Communication for Serial-to-Fiber Converters

Richard Hsia, Business Development Manager

Richard.Hsia@moxa.com

In general, serial-to-fiber converters on the market today only provide one serial port and one fiber port for physical layer media conversion. But as industrial networking applications increase in complexity, users have started to demand additional capabilities and features including easy wiring, system diagnostics, and dual hosts/slaves. This white paper explains how to use 3-way communication on Moxa's ICF-1150 series media converters to achieve these various benefits on a single device.

Let us use the ICF-1150 series media converters to illustrate how 3-way communication works for serial-to-fiber converters. Each ICF-1150 is equipped with one fiber port and two serial ports. The D-sub connector is used for RS-232 communication and the removable terminal block is used for RS-422 or RS-485 communication. When the ICF-1150 series converter receives data through one of its ports, it will send the data back out through the remaining 2 ports. For example, when the ICF-1150 series receives a command from the remote device via the fiber port, it will convert the command and transmit it via the RS-232 port and RS-422/485 port at the same time.

Released on January 7, 2009

Three-way communication brings the following benefits:

1. Easy wiring
2. Easy system diagnostic
3. Dual hosts/clients, redundant solution, and data logging

Easy Wiring

Compared to traditional serial-to-fiber converters that only have one serial port and one fiber port, Moxa's ICF-1150 series provides 3 ports for data communication. Based on our many years of experience, we have selected the most common connectors used in the industry, such as D-sub for RS-232 connections and removable terminal blocks for RS-422/485 connections. Adopting the most commonly used connectors can save users a great deal of time in field installation and maintenance.

Easy System Diagnostic

When connecting PLCs or meters to a fiber converter, users need a clear way to diagnose what is wrong if the send/receive data are blocked, stopped, or an error occurs. In most cases, users will change the wiring to see if the failure was caused by improper wiring. Users may also try using a USB-to-serial converter to connect the converter to a laptop that can run diagnostics. After checking the data transmission, users remove the USB-to-serial converter and reconnect all the wires. However, none of that can guarantee that the new wiring is correct.

Three-way communication can solve this problem and fulfill customer expectations. Let us examine the following 2 scenarios to better understand this concept.

A.   Connecting an RS-485 device to fiber converter.
Since the RS-232 port on the fiber converter is still unused, we can connect the RS-232 port to a laptop and open the COM port by "Hyper Terminal" to see what data is running. After running diagnostics and correcting the wiring (if needed), we can remove the link between the laptop and converter without affecting any running systems.

B.   Connecting an RS-232 device to a fiber converter.
We can use a USB converter such as Moxa's UPort 1130 or UPort 1150 to provide a computer with direct access to RS-485 data. Again, after checking the data and correcting the wiring (if needed), we can remove the link between the laptop and converter without affecting any running systems.

Dual Hosts/Clients, Redundant Solution, and Data Logging

Since the 3 ports are virtually connected, you can only connect a maximum of 2 serial ports to different devices. For example, users can connect 2 master devices, such as PLCs, one at the RS-232 side and the other at the remote side via fiber link. The slave device can be connected to the RS-485 port and receive the command from a local master or remote master.

A computer can also be used to monitor the system and check all the running communication as a solution for data logging.

Limitations

Again, since the 3 ports are virtually connected, the converter will not receive data from more than two devices. Connecting more than two devices will cause the program to halt or disrupt data transmission since two devices sending data at the same time will cause you to receive incorrect and unexpected data.