

Serial to Network Data Conversion

Conversion of NMEA-0183 serial data into network data, for distribution and aggregation

Summary

Typically, navigation data onboard a ship may come from a number of different sensors and may be required simultaneously by several different clients. For example, heading data might be supplied by a gyro system whereas position will generally come from a GPS system.

Navigation data is often supplied from the source device via a serial port. This can be inconvenient if data from this device is required by multiple systems or if a system requires input from multiple devices.

It is often therefore desirable to convert the serial data into network packet data, which can easily be shared by many systems. Data from multiple sources can also be sent onto the same network address, meaning that a system can easily receive data from various sources time-stamped and aggregated onto one convenient network socket.

Conversion from a serial port stream of NMEA-0183 data to network packets can be implemented in software or by dedicated third-party hardware.

This application note describes Cambridge Pixel's software option and also explains how the hardware option should be configured, in order to create network packets that the SPx software can use.

Overview

Conversion of serial NMEA-0183 data to UDP network messages is a reasonably straightforward process. The converter must accept an input stream of serial data, split it into the individual NMEA-0183 sentences and then output each of these sentences within its own, self-contained UDP message.

NMEA-0183 sentences should be terminated with a carriage return <CR> and a line feed <LF> character. The line feed (ASCII code 0xA) can therefore be used to delimit individual sentences within the serial data stream. Each time the converter reaches a delimiting character and has a complete data sentence it must output that data sentence, as a UDP message, onto the network. This process is depicted in Figure 1.

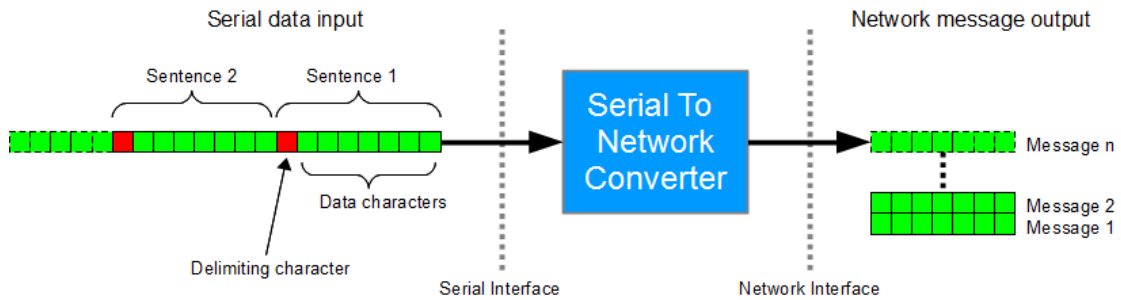


Figure 1: Serial to Network Conversion

The serial to network conversion can be a software process, taking its input from the host’s serial port, or it can be a dedicated piece of third-party hardware, with a serial port connector for input and an Ethernet output port.

The output of multiple converters may be directed at a single network address, allowing a receiving application to access data from multiple sources on a single socket.

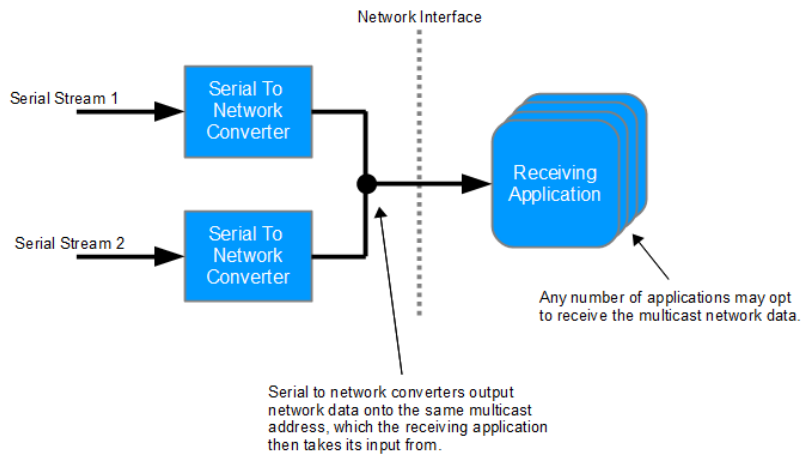


Figure 2: Many-to-Many Serial Distribution

receiving applications to access all of the data simultaneously. This situation is summarised in Figure 2.

Software Conversion

Cambridge Pixel supplies a utility called “SPxSerialToNet” with its SPx software products. SPxSerialToNet takes serial data directly from the host PC’s serial port (e.g. “COM1” on Windows or “/dev/ttyS0” on Linux) and outputs it onto the designated network address and port.

SPxSerialToNet supports a number of operating modes, which define different message types for the serial input. Mode number 1 tells the application to expect to receive NMEA-0183 data from the serial port and therefore that individual sentences within the stream begin with a “\$” character and end with a line feed <LF> character.

A typical SPxSerialToNet command is given below:

```
spxserialtonet -m1 -sCOM3 -b38400 -h -a239.192.50.89 -p5089
```

This command will start the SPxSerialToNet application in mode 1 (NMEA-0183), taking input from serial port "COM3", with a baud rate of 38,400 and outputting the packetised NMEA-0183 sentences onto multicast address 239.192.50.89, port number 5089.

The "-h" option in the command above prevents the SPxSerialToNet application from prepending the NMEA-0183 data with an SPx message header. This means that the messages generated on the network could also be interpreted by non-SPx applications.

Hardware Conversion

There are several commercial serial-to-network converters available, from manufacturers such as Moxa.

Just as with the software conversion process, these devices need to be configured appropriately so that each output data message contains a single, complete NMEA-0183 sentence. In order to achieve this, the device should be set to output UDP messages and a delimiting character may also have to be set explicitly. For NMEA-0183 data this delimiting character should be the line feed <LF> character.

The image below in Figure 3 shows an example of the configuration of a Moxa device, configured to output self-contained NMEA-0183 UDP messages. Note that "0a" is the hexadecimal ASCII code for the line feed character.

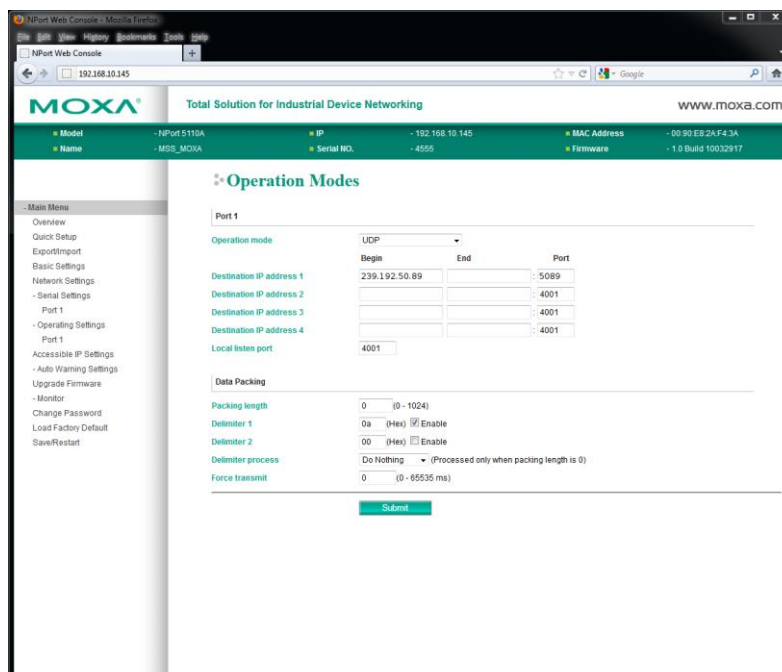


Figure 3: Example Moxa Serial to Network Device Configuration

As well as delimiting individual NMEA-0183 sentences correctly, it is important to ensure that the device does not perform any buffering that may delay the output of UDP messages unnecessarily. Some devices provide the option of buffering incoming data and only outputting network messages after a certain amount of data has been received, or a certain period of time has elapsed. This can result in the device outputting messages in bursts, rather than each time a full NMEA-0183 sentence has been received. The consequence of this is that the receiving application does not have an accurate knowledge of the platform motion at any given time.

Obviously, the controls for configuring devices will vary between manufacturers and device models but the principles are the same:

- Configure the device to output UDP messages
- Delimit NMEA-0183 sentences using the line feed <LF> character (ASCII code 0xA)
- Disable any buffering on the device that may hold up the output of complete messages

Conclusion

Serial messages can be converted to network data via software or hardware. It is important to ensure messages are delimited correctly and sent promptly without buffering. The information above gives details on this, but please contact Cambridge Pixel if more information is required.

< End of document >