



## TCP/IP Developers Kit

The comm. protocol and TCP/IP Development interface provide developers with the ability to interface with the Kestrel autopilot through the Virtual Cockpit ground control application.

No License Fee

### Communication Protocol (serial)

The Kestrel Autopilot and Virtual Cockpit talk through a proprietary communications protocol by sending data packets serially over a communications link. Developers may choose to replace some or all of the functionality of the Virtual Cockpit by understanding the published communication protocol specification provided in the Developers Kit. Doing so requires a programmer capable of opening up a computer serial port and sending data packets to the Commbus. A disadvantage of working with the pure serial protocol is that only one application can talk to the Commbus at a time so developers would have to replace all of the required capabilities that are already offered in the Virtual Cockpit (i.e. Mapping Tools, PID tuning and configuration). However, this disadvantage is overcome by using the TCP/IP development interface built into the Virtual Cockpit.

### TCP/IP Development Interface Protocol

Built into the Virtual Cockpit is a TCP/IP development interface socket server. This server accepts connections from outside applications and depending on the data received, the Virtual Cockpit will react to the commands. Using the TCP/IP interface simplifies development time because the Virtual Cockpit will take care of the underlying low level serial communication and packet generation. This also allows developers to continue to use all the inherent features provided by the Virtual Cockpit while still allowing developers to add their own special modifications needed for custom projects.

To help developers get familiar with the TCP/IP development interface, an open source Visual C++ 6.0 MFC application is provided in the Development Kit. This application shows how to interface to the Virtual Cockpit in a simple modular manner and issue some of the common commands such as pass-through packets that will be sent directly through the Virtual Cockpit to the Kestrel autopilot.

An example of using the TCP/IP interface and Developers Kit might entail a developer adding a new sensor to their UAV. Code could be written on the autopilot to talk to the sensor and gather critical information. The information could potentially be requested by the developer's custom written application that sends a request packet through the Virtual Cockpit to the autopilot. The autopilot would respond with the data to the Virtual Cockpit and the Virtual Cockpit would pass it through to the TCP/IP interface.

