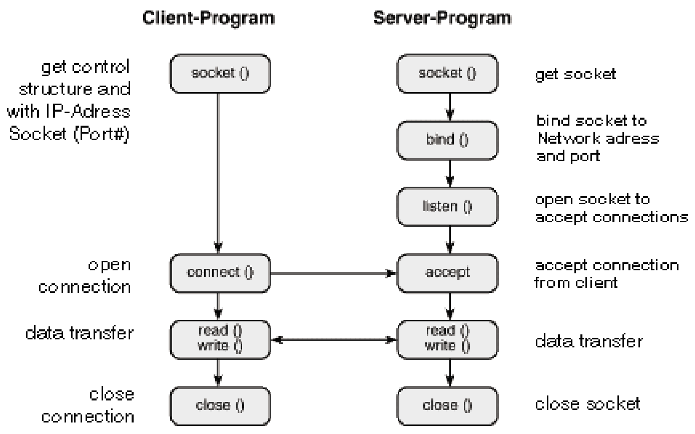
Application for the serial Com-Server:

With 30 lines over the network: TCP/IP sockets as a programming interface

With the socket-API (e.g. Winsock under Windows or Berkely Sockets under UNIX), many different applications can be realized within your existing software. API offers all functionalities for data transfer over the network.

One of the easiest options to establish connections over a network is available with Microsoft Visual Basic, version 5.0 or higher. Even without any expert knowledge on networking, you will be able to transfer data over the network; here you can find our little sample program. You will soon see that 30 lines of source text and our small adapters are sufficient to transfer serial data over the LAN/WAN.



The programming of these APIs is even less complicated than the ordinary function calls for the integration of a COM port, as the entire complexity of the network mechanisms is packed into the socket port.

The diagonal arrows indicate how the client and server programs are working together. All you need to do is to implement one program, depending on whether the Com-Server is to be operated in client or server mode respectively.

Server or client?

As a TCP client, the Com-Server is able to independently establish a connection whenever data needs to be transferred and to disconnect, when the data transfer is completed. While no connection to another server is established, the Com-Server can also act as a server by accepting connection requests from other clients.

No special settings are necessary to operate the Com-Server as a TCP server. In this case, the connection is controlled by your client process, i.e. you both establish and terminate the connection. Once a connection is established, data can be exchanged between the two processes. The Com-Server transfers the data from the network to the serial port or the I/O ports and reads data from the serial port or the I/O ports to make it available to the client process.

TCP or UDP?

TCP is a connection-based protocol; while data is being transmitted there is a fixed, logical connection between client and server. TCP includes all the mechanisms necessary for opening a connection, ensuring error-free data transfer over the network, and then closing the connection. Here the protocol stack creates and administers dedicated buffers for each connection. Note that if many Com-Servers need to be accessed from one computer, you may in extreme cases reach a memory limit.

In contrast, UDP does not incorporate the mechanisms for repeating data packets or checking them for completeness. Data transfer using UDP is recommended, when there is already a host protocol for ensuring error-free transmission between the terminal device on the Com-Server and the application on the TCP/IP station, or when a large number of network stations need to be accessed simultaneously.

The (sometimes important) special cases

For those who make use of socket programming, the Com-Server provides control and service ports allowing for additional functions, which can be activated parallel to the data transfer in the form of server processes. This gives you the option for example of remote configuration and maintenance of the unit. For more information on this topic, please refer to part 6 of the Com-Server manual..

To the practical example

We are available to you in person:

Wiesemann & Theis GmbH
Porschestr. 12
42279 Wuppertal
Phone: +49 202/2680-110 (Mon.-Fri. 8 a.m. to 5 p.m.)
Fax: +49 202/2680-265
info@wut.de