Getting Started with WatTCP

94-0020

JK microsystems products are distributed with the WatTCP stack. WatTCP is a public domain TCP/IP implementation for DOS written by Erick Engelke. Our example code and simple web server make use of the WatTCP libraries to perform TCP/IP level communication.

This guide outlines the basics in configuring WatTCP and preparing to develop a TCP/IP application. This guide is not intended as a substitute for the WatTCP Programming Manual or an indepth TCP/IP tutorial.

Overview

WatTCP is a library of functions written in Borland C and assembly language. Complete source code is on the development CDROM. TCP/IP applications are written in C or C++ and then linked with the WatTCP library. Due to the fact that TCP/IP is implemented at the program level, not the kernel or OS level, the JK microsystems SBC will not respond to any network activity until an application is running. This means that pinging an SBC that is sitting at the DOS prompt, will result in a timeout message.

Setup

WatTCP communicates to the physical layer via a packet driver. A packet driver must be loaded prior to starting the WatTCP application. Only one packet driver can be addressed with WatTCP. As with all networks, some configuration is necessary to inform the software about the network node. WatTCP makes use of an ASCII text file, named `WATTCP.CFG`, to store this configuration information. The configuration file contains network parameters and may also contain parameters for an application and debugging. A minimal configuration file would look like this:

```
my_ip=10.32.1.1
netmask=255.0.0.0
```

In this example, the application will respond at address IP address 10.32.1.1 within a Class A subnet.

A more complete configuration file might look like this:

```
my_ip=129.97.176.99
netmask=255.255.254.0
nameserver=129.97.128.24
nameserver=129.97.128.196
gateway=129.97.176.1
DEBUG.MODE=ALL
DEBUG.PROTO=ALL
DEBUG.FILE=b:\junk
```

A default configuration file is present on the board and was used in production test. This file will need to be updated before you start testing your network. The file can be edited with a text editor, created with the `CONFIG.EXE` program, or uploaded from the development PC. Be sure there is only one configuration file on your SBC. This file should be in the same directory as your application. In general, the application and configuration file reside in the root of the B: drive. If multiple configuration files are present, incorrect network parameters may be loaded.

Each node on a TCP/IP network must have a unique address within that network. Depending on the design of the network, address may be global. Duplicating addresses will create all kinds of problems both for you and other users.

If you are part of a larger network, you must obtain valid network parameters from your systems administrator. It is not acceptable to pick an IP address (or use the default) and expect the SBC to communicate on your network.

If you are creating an isolated network for your development, there are several addresses that are reserved for used in private networks, i.e. networks that are not connected to the Internet:

Class A: 10.xxx.xxx.xxx (1 network, 33554430 nodes)
Netmask= 255.0.0.0
Class B: 172.16.xxx.xxx through 172.31.xxx.xxx (16 networks, 32766 nodes)
Netmask= 255.255.0.0
Class C: 192.168.0.xxx through 192.168.255.xxx (256 networks, 254 nodes)
Netmask= 255.255.255.0

Use of these network addresses reduces the chances for interfering with other devices.

PING

Ping is a valuable network diagnostic tool. Ping sends a echo request to the specified host and waits for a reply. If a reply is received, network statistics are provided. Once the `WATTCP.CFG` file has been created on the SBC and, if necessary, the parameters on the development PC updated, it is useful to ping your development PC. You may need to upload `PING.EXE` to the SBC.

```
B:\>ping 10.10.11.11
Pinging [10.10.11.11]
sent PING # 1 , PING receipt # 1 : response time 0.00 seconds

Ping Statistics
Sent        : 1
Received    : 1
Success     : 100 %
Average RTT : 0.00 seconds
```

Next, you should try pinging the SBC from the development computer. Remember you need to have a WatTCP application running in order for the SBC to respond to the echo request. Running the webserver or just starting ping with the ‘-S’ option will accomplish this task.
On the SBC:

```
B:\WEBTCP
```

Or

```
B:\PING 10.10.11.11 –S
```

On the development PC:

```
C:\WINDOWS>ping 10.32.1.1

Pinging 10.32.1.1 with 32 bytes of data:

Reply from 10.32.1.1: bytes=32 time=3ms TTL=250
Reply from 10.32.1.1: bytes=32 time=1ms TTL=250
Reply from 10.32.1.1: bytes=32 time=1ms TTL=250
Reply from 10.32.1.1: bytes=32 time=1ms TTL=250

Ping statistics for 10.32.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum =  3ms, Average =  1ms
```

If you can successfully ping in both directions, you have a working development network and can proceed to work on writing you own application.

Note that WatTCP does not implement the loopback (local host) address. Therefore, pinging 127.0.0.1 will timeout.

If you encounter troubles, it is useful to verify the network settings on your development PC. Use the IPCONFIG program (Windows 98) to display the network configuration:

```
C:\WINDOWS>ipconfig

Windows 98 IP Configuration

0 Ethernet adapter :

IP Address. . . . . . . . . : 10.10.11.11
Subnet Mask . . . . . . . . : 255.0.0.0
Default Gateway . . . . . . :
```

Also, verify that the ‘LINK’ LED is illuminated on the SBC, the PC and on the network hub. You should see the activity LED’s flash when the ping packets are sent or when there is other network traffic. If the LED’s are not illuminated, there are network wiring problems that need to be resolved.

Writing a WatTCP Application

WatTCP applications are generally written in Borland C. Examples of network clients and servers are included on the development CDROM and in the WatTCP distribution. Carefully reading these examples and looking at the development environment settings (where applicable) will help getting the application development started.

The most common area of confusion is creating a project in the Borland IDE. The WatTCP library must be included in the project. There are three libraries in the distribution, one for each of the common memory models: `WATTCPSM.LIB` (small model), `WATTCPLG.LIB` (large model) and `WATTCPHG.LIB` (huge model). You must use the library that corresponds to the memory model you are using for your project. Please refer to the document ‘Getting Started with the Borland IDE’ available for download from the JK microsystems website (http://www.jkmicro.com).

Compiler and linker messages frequently do not provide enough information to quickly fix errors. Listed below are several things to check:

1. Is the WatTCP library included in the project?
2. Does the WatTCP library match project memory model?
3. Does the ‘Source Directory-Include’ path include the path to the WatTCP header files?
4. Is the `TCP.H` header included in the application?
5. Is the TCP/IP socket structure (tcp_Socket) declared as a static or global variable, not a local?
6. If the project is written in C++, is the `TCP.H` header included as a C header?

```
        extern “C” {        // wattcp is a C library, we’re using the C++ compile
        #include <tcp.h>
        }
```

Revision	Date	Author	Changes
A	18MAY00	EW	First Issue
B	12DEC01	EW	Add example .h include for use with C++