



Streaming GPS Data using a DNC/CNC System.

**(Configuration of Quatech's Serial Device Server Model
DSE-100D to run rtnt and sharc)**

Version 1.0

Last Modified: 02-11-2008

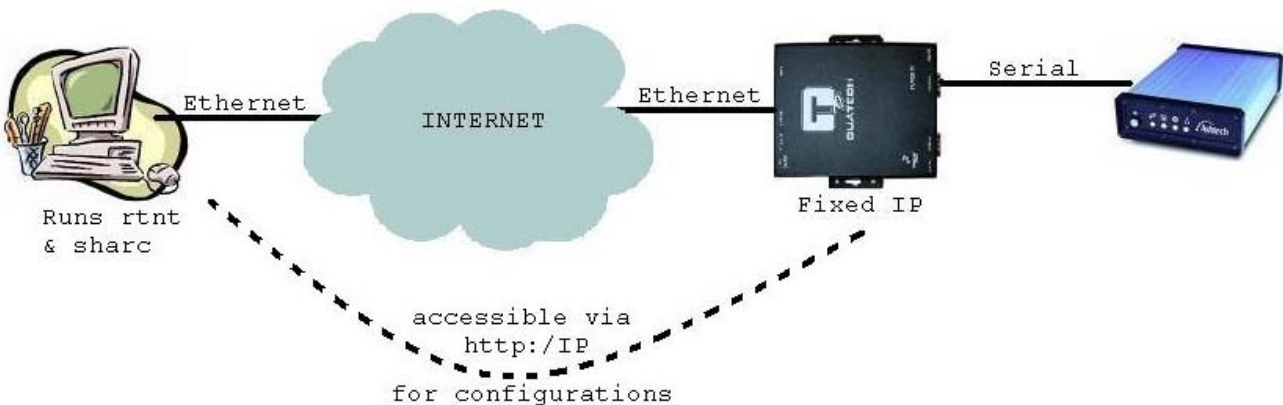
Author: Nic Flores

1.0 Abstract

Several of NASA's Global GPS Network (GGN) sites use Ashtech MicroZ GPS receivers produced by Thales for GPS data collection. Thales has recently started to produce internet ready MicroZ's (iCGR's) but the only way to communicate with CGR's (non internet ready MicroZ's) is through their serial ports. GGN sites equipped with CGR's need to have a computer attached to the MicroZ in order to communicate with it remotely. Quatech.com offers Serial Device Servers which can expand the reach of any serial device by network-enabling them for remote data collection, access and control. In this report we present how Quatech's Serial Device Server Model number DSE-100D can be configured to run **sharc** and **rtnt** remotely on a linux based computer.

2.0 System Diagram

Below we give the connection diagram of the equipment we used for this testing the Serial Device Server.



2.1 This report will describe:

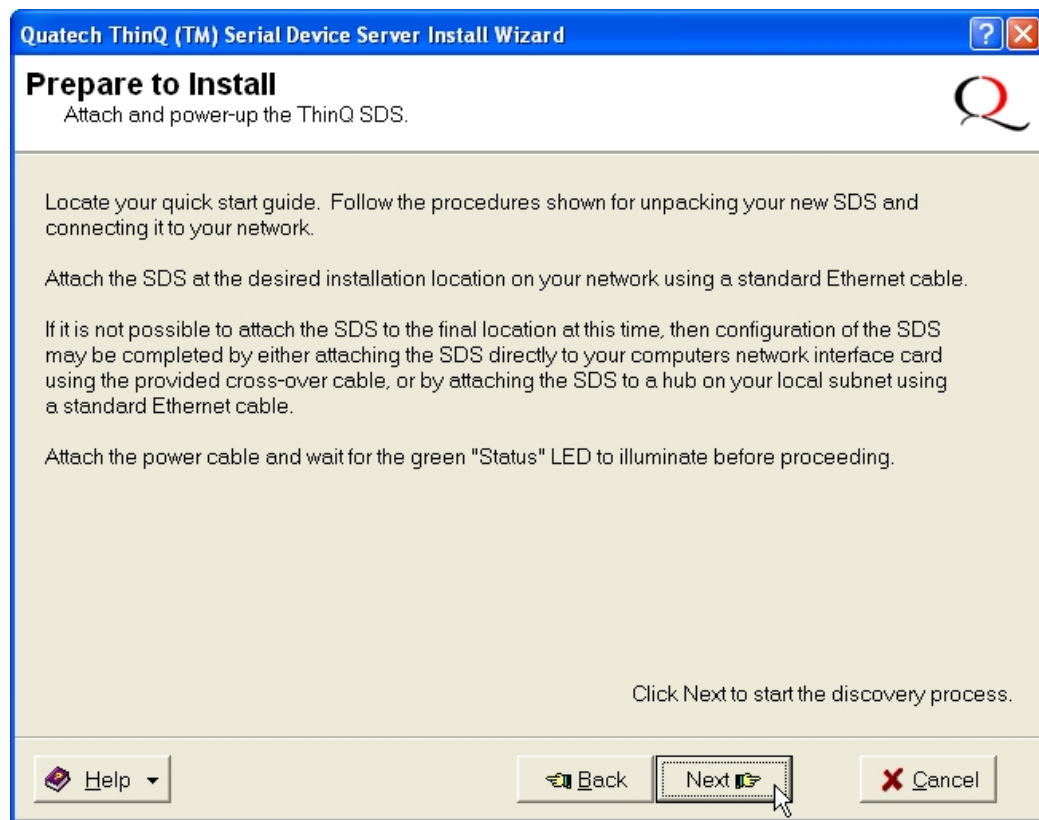
1. How to configure the Serial Server
 - a) with a fix IP
 - b) port settings
2. The installation of software needed on your linux pc to communicate with the receiver over the internet and run **sharc** and **rtnt**.
3. The power consumptions analysis of the Serial Device Server.
4. Future ideas to consider with Serial Device Servers.

3.0 Configuring the Serial Server Model#: DSE-100D

Out of the box the Serial Server needs to be given a fixed IP address. Once this it has its own IP address the rest of the configuration can easily be done via its own webpage.

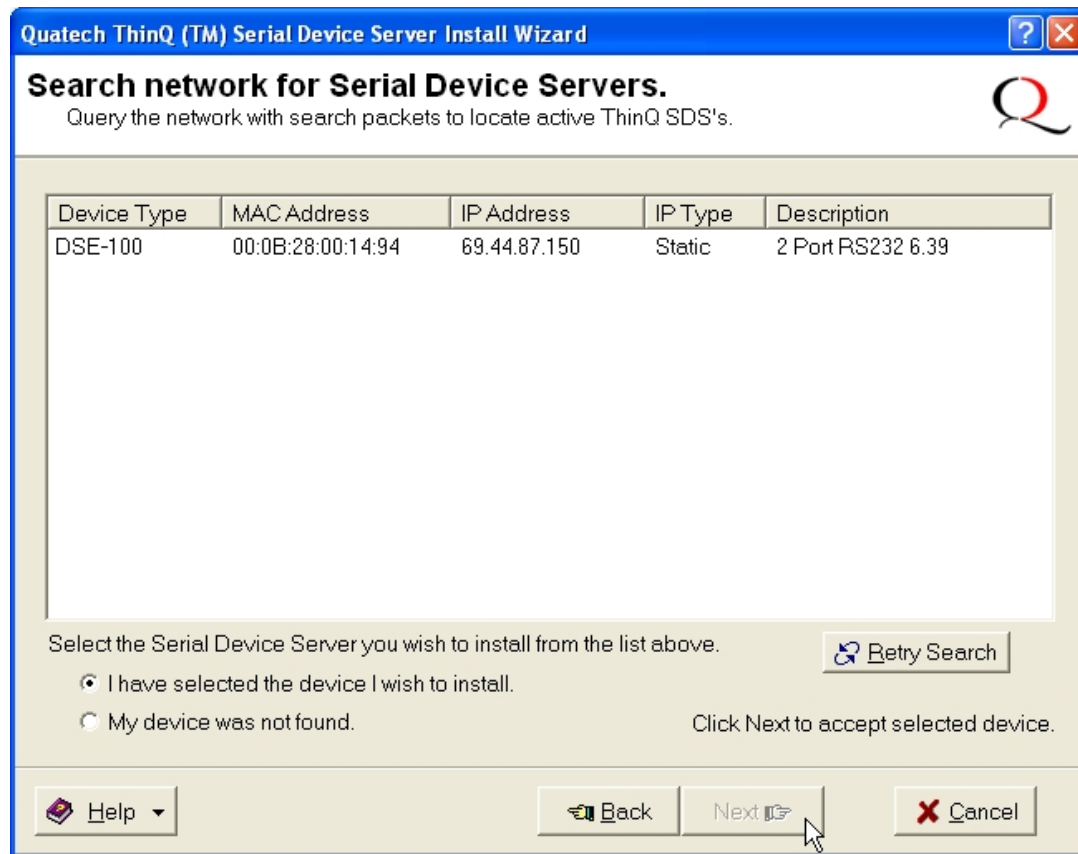
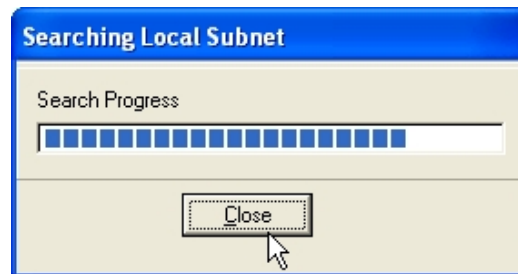
3.1 Giving the Serial Server its own IP address

1. Take the Serial Server out of its box. Power it up using the included power adapter and using a straight Cat 5 Ethernet cable plug it into an Ethernet port on the back of your Windows Desktop.
2. Included in the Serial Server box there should be a Quatech SDS Installation CD-ROM you can insert into your CD-Rom drive. The CD should automatically start up and take you to an installation menu. Click on **INSTALL SDS** and then click **Next** at the Welcome Window this will drop you into the following window:



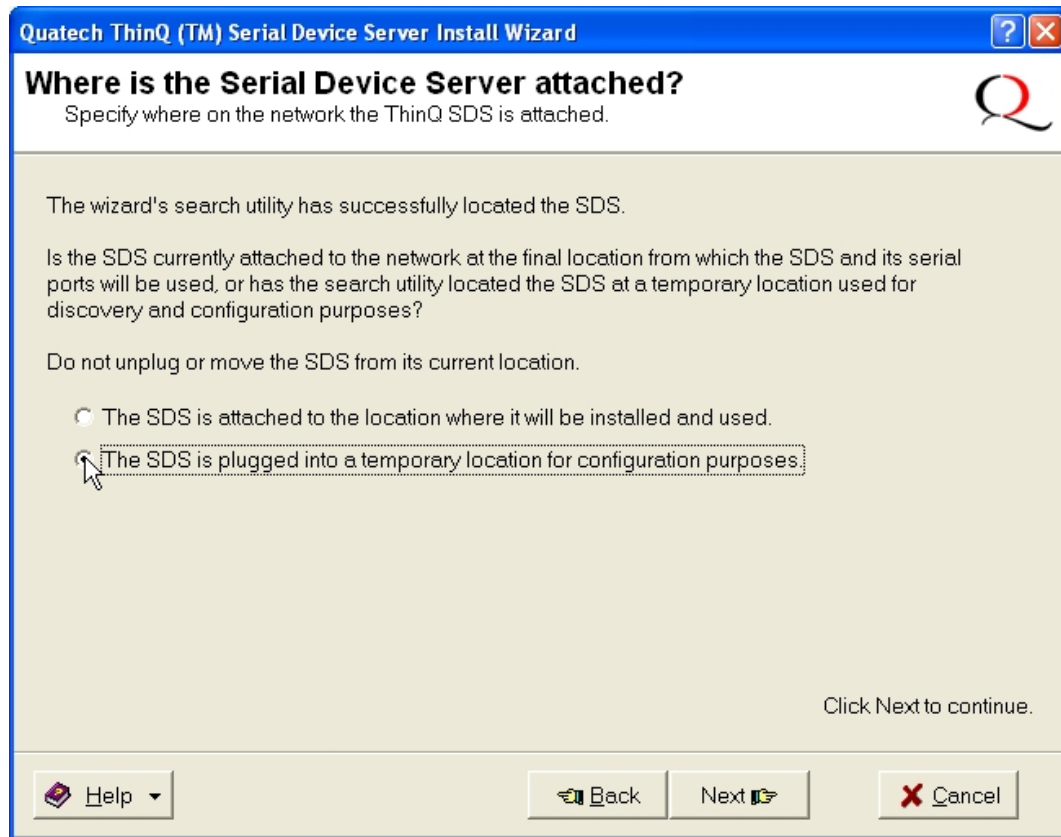
Click **Next**.

- The next window will search for the Serial Device Server attached to the Ethernet port on the back of your PC. The search starts automatically with a progress window showing up almost immediately:

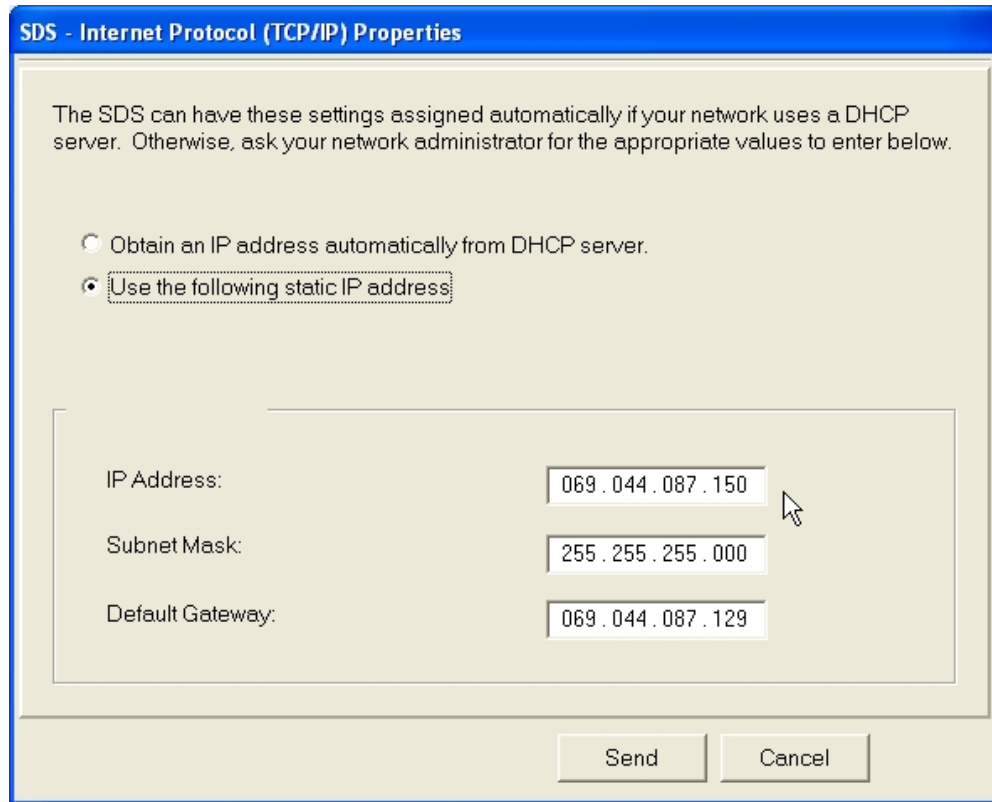


Once the search is complete you will see the Serial Device Server information listed in the window. Select the device found then click **Next**.

4. The next window needs you to specify if the Serial Device Server is at the final location that it will operate from. Since it is not select the second radio button and click **Next**.



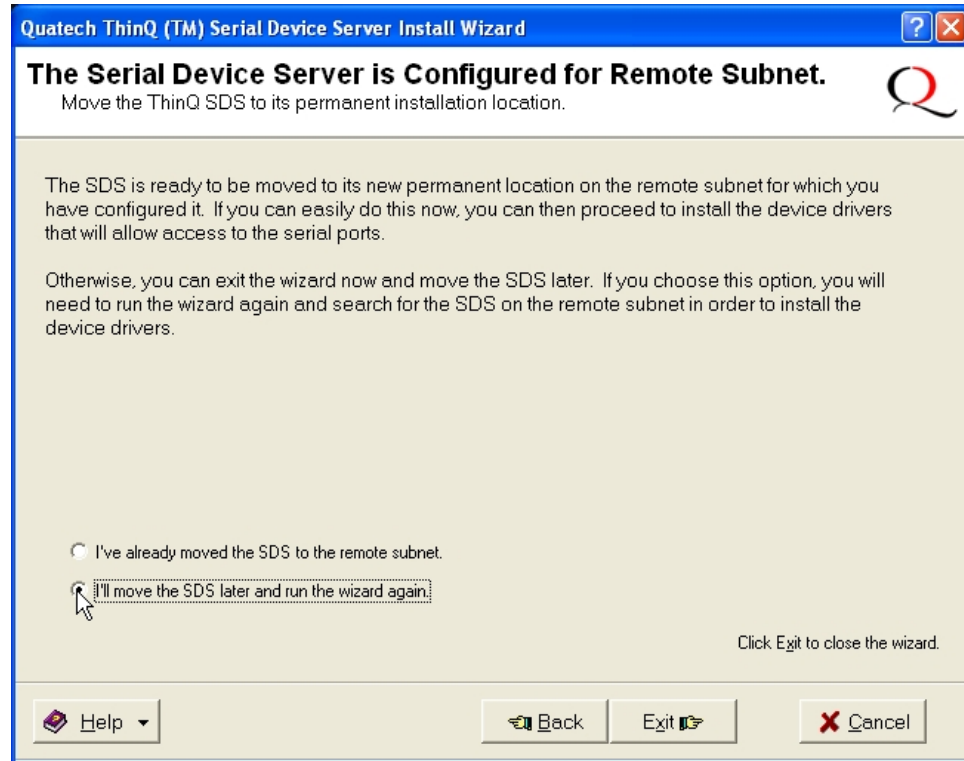
5. The next window allows you to set the IP type, IP address, Subnet Mask, and the Gateway. Click on **Reconfigure SDS** and enter the IP information:



After you have entered the correct IP information click **Send** and you will be asked if you want to restart the Serial Device Server. Click **OK** and wait for a few seconds.



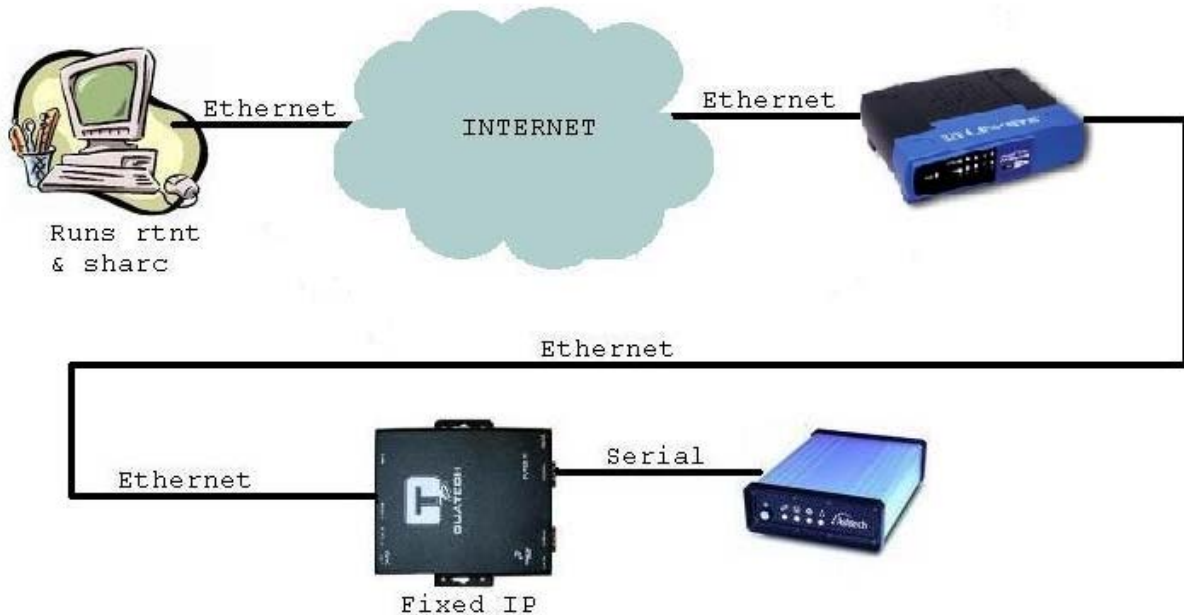
6. After clicking **OK** in the previous step the next window you get asks if you want to run a wizard that will allow you to talk to the Serial Device Server. You do not need to do this. Select the second radio button and click **Exit** to finish the configuration.



Now the Serial Device Server is configured and ready for testing.

4.0 Installing interceptty-0.6

To begin testing the Serial Device Server at the Unavco R&D Lab setup the following diagram:



4.1 Virtual Serial Ports

In order to communicate with the MircoZ while sitting at your office linux box you need to create a *virtual serial port*. An easy way to do this is to install and run interceptty-0.6. This program is not distributed by Quatech, although Quatech has a similar piece of software but interceptty seems to be more user friendly. Simply untar interceptty-0.6.tar.gz in /usr. Then `cd /interceptty-0.6` and run `./configure`, then `make`, then `make install`. These commands should run without any errors if the source tree is good. Once you have installed interceptty we are ready to create a virtual serial port.

The name we would like to give our virtual serial port has the form `ttyS#` where # is the number of your last serial port plus 1. To determine how many `ttyS`'s you have, as root, run `cat /proc/tty/driver/serial`. This command will spit out something like:

```
serinfo:1.0 driver revision:
0: uart:16550A port:000003F8 irq:4 tx:0 rx:0
1: uart:16550A port:000002F8 irq:3 tx:0 rx:0
```


This means we have two serial ports, ttyS0 and ttyS1. So the name virtual serial port we create will be ttyS2. To create the virtual serial port run the following command as root:

```
interceptty -q '@<serial_server_ip_address>:<port>' /dev/ttyS2
```

The “-q” option tells interceptty to run in quiet mode. During testing I suggest you use the “-v” option, this tells interceptty to run in verbose mode. The “-v” option will help with troubleshooting in case things don’t work.

The <serial_server_ip_address> should be clear. It’s the IP address you gave the Serial Device Server during section 3 above.

The <port> option needs some explanation. Quatech Serial Device Servers model number DSE-100D come with two serial connections and each has a port number assigned. For the DSE-100D model the port numbers are assigned as follows:



Now, suppose the IP address to the Serial Server is 69.44.87.150 and you have a MicroZ attached to Serial 1, then to create a virtual serial port on your pc to communicate with the MicroZ you would run

```
interceptty -q '@69.44.87.150:5000' /dev/ttyS2
```

assuming that you don’t already have a ttyS2 on your linux box.

4.2 Virtual Serial Port Permissions

After running the command in section 4.1 a symbolic link is created in `/dev`. It looks like:

```
lrwxrwxrwx 1 root root  date-time ttyS# -> /dev/pts/<some_number>
```

If you now `cd /dev/pts` and do an `ls -al` you will see the following:

```
crw- -w- - - - 1 gpsops tty 136, 0          date 0
crw- -w- - - - 1 gpsops tty 136, 1          date 1
crw- -w- - - - 1 gpsops tty 136, 2          date 2
crw- -w- - - - 1 root tty 136, <some_number> date <some_number>
crw- -w- - - - 1 gpsops tty 136, 5          date 5
```

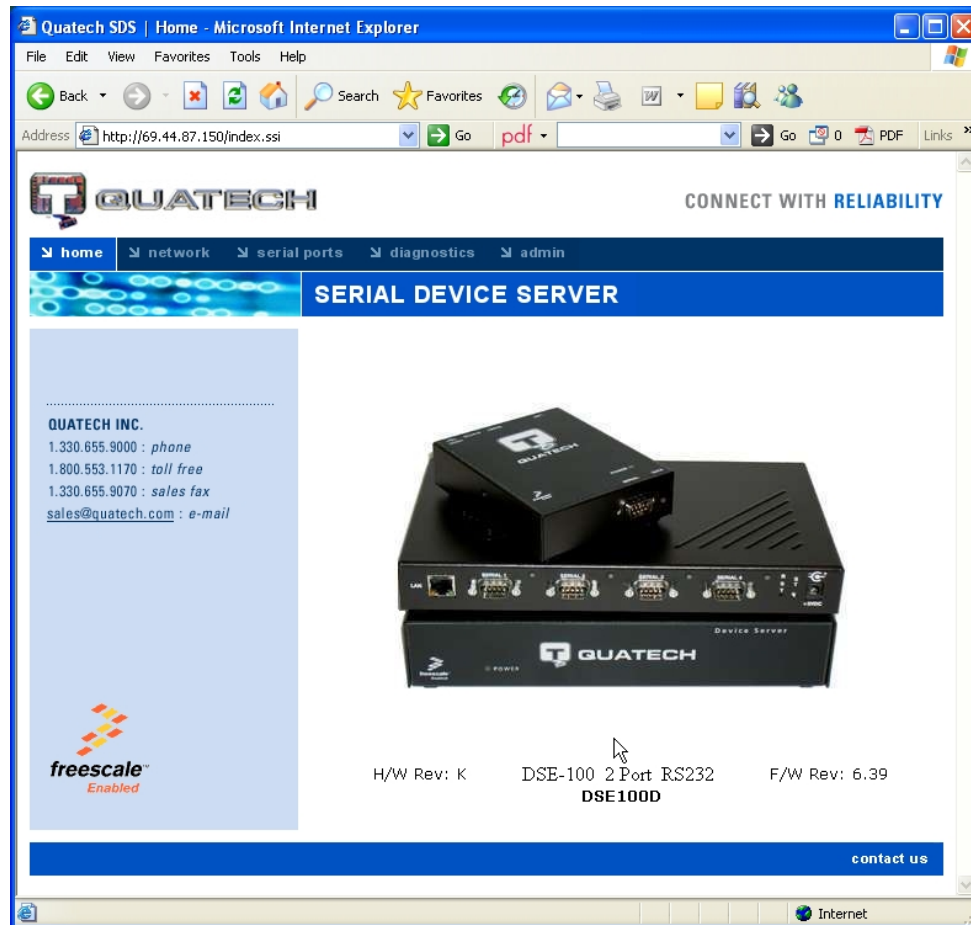
Notice the owner of line 4, it is owned by root since we had to run the command in section 4.1 as root. Since, `rtnt` and `sharc` are meant to be run as the user `gpsops` we need to change the owner of the file in line 4. To do this run the following command as root:

```
chown gpsops <some_number>
```

This will allow you to run `sharc` and `rtnt` as the user `gpsops`. Before we run these programs we need to configure the ports on the Serial Device Server.

5.0 Configuring the Serial Server over http

Since the Serial Device Server now has its own IP address we can open up a web browser to configure it. The only web browser that seems to work best is Internet Explorer. Assuming the Serial Device Server is connected as in the figure in section 4.0 you can enter `http://<serial_server_ip_address>` and get the following introductory website:



This is the introductory page to the Serial Device Server. This page displays the HW REV and the FW REV which are good values to know in case you need to talk to the people at Quatech about this item.

Clicking on the “network” takes you to following page:

Quatech SDS | Network - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Refresh Print PDF Links

Address http://69.44.87.150/ipaddress.ssi Go

QUATECH CONNECT WITH RELIABILITY

home network serial ports diagnostics admin

SERIAL DEVICE SERVER

Network Setup

☒ **Static**

IP Address 69.44.87.150

Gateway Address 69.44.87.129

Address Mask 255.255.255.0

☐ **DHCP**

MAC Address 000B2800278B

Submit

Network Setup

SNMP Setup

QUATECH INC.

1.330.655.9000 : phone

1.800.553.1170 : toll free

1.330.655.9070 : sales fax

sales@quatech.com : e-mail

freescale
Enabled

contact us

The static IP configuration should be the same as the IP we gave the Serial Device Server in section 3.0. Within this same page you can click on “SNMP Setup” to configure who can access the Serial Device Server.

Clicking on the “serial ports” tab takes you to the following page:

Quatech SDS | Serial Ports - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://69.44.87.150/sports2.ssi> Go pdf Go 0 PDF Links

QUATECH CONNECT WITH RELIABILITY

home network **serial ports** diagnostics admin

SERIAL DEVICE SERVER

Serial Port Setup

Serial Port Setup
Serial Port Status

QUATECH INC.
1.330.655.9000 : phone
1.800.553.1170 : toll free
1.330.655.9070 : sales fax
sales@quatech.com : e-mail

freescale™
Enabled

1. Select Serial Port
Port 1 Port 2

2. Select Operating Mode
☐ Normal ☐ Tunneling ☒ Raw TCP ☐ Auto TCP ☐ Raw UDP

3. Configure Operating Mode
Rate Multiplier Performance Selector Heart Beat Time
Auto ☒ Balanced ☐ Low Latency 0
Baud Rate Parity Data Bits Stop Bits Flow Control
19200 None 8 1 None

4. Save Changes
Save

Internet

If you have a MicroZ attached to the Serial 1 port on the Serial Device Server the above settings for Port 1 will have to be set in order to communicate with MicroZ.

5.1 Running **sharc** and **rtnt**

Once you have configured Port 1 on the Serial Device Server as above and you have installed **interceptty** and fixed the permissions of the symbolic link created now you are ready to run **sharc** and **rtnt** as you normally do.

To run **sharc** you can use the following:

```
sharc --port /dev/<virtual_port> --baud 19200 --terminal
```

To run **rtnt** you need to create the file **install_microz.txt** that contains

```
install_microz -n jplb -i 21 -s S<last_#_of_vitual_port> -cl 1000 -  
donotsend_ip1 -pl 3138 -ip1 127.0.0.1
```

After this you need to run

```
source install_microz.txt
```

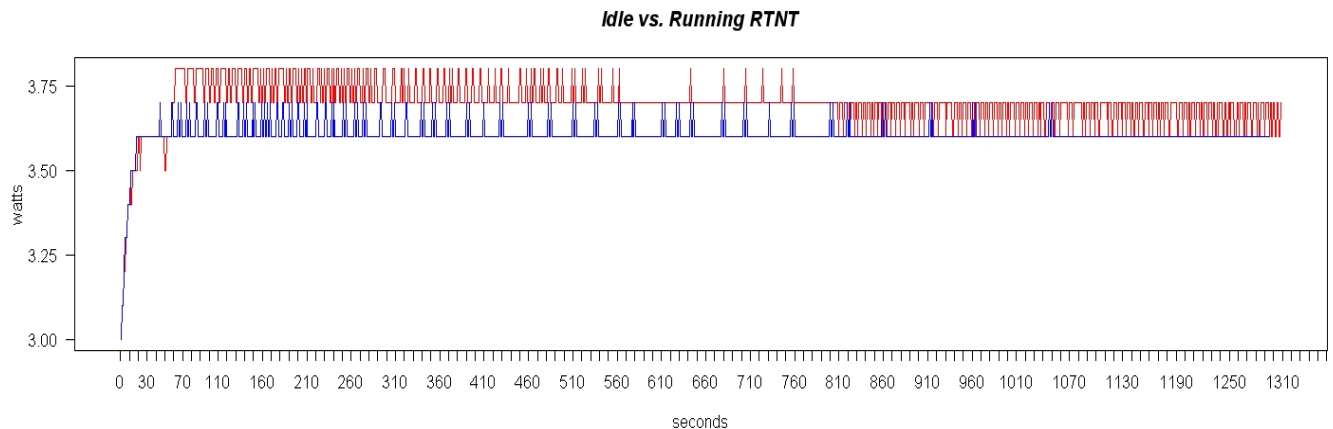
then run

```
run_MICROZ
```

6.0 Power Consumption Analysis of DSE-100D

6.1 Power Consumption of Serial Device Server

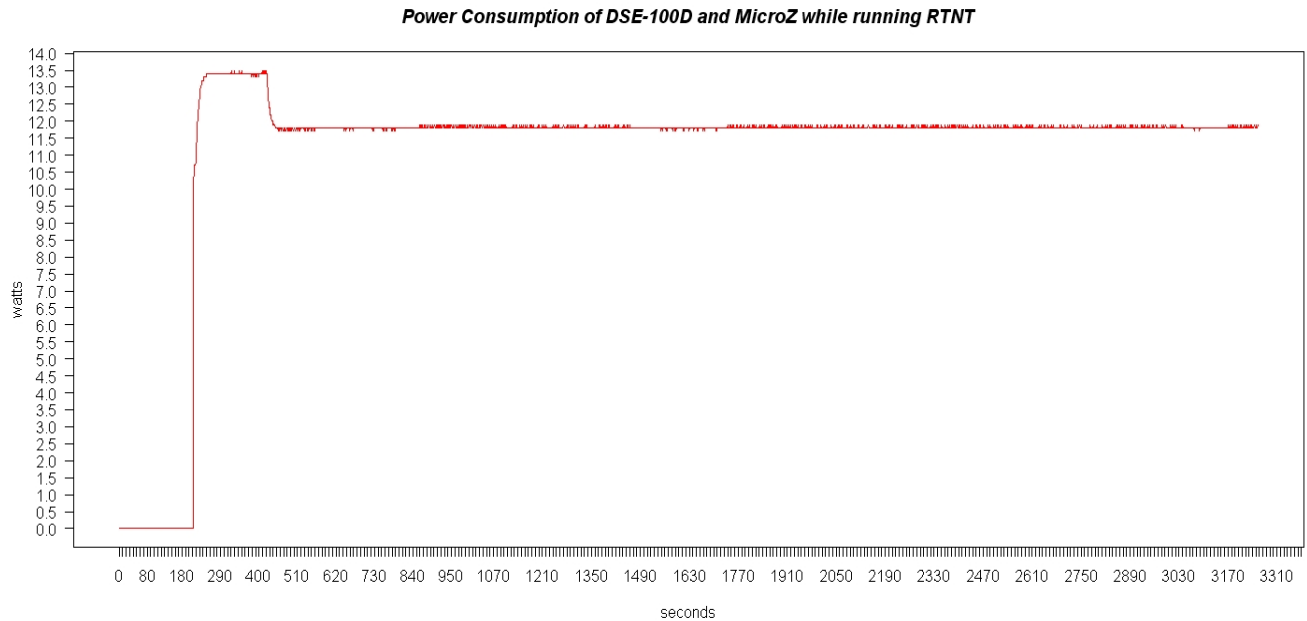
We tested several wattage consumption scenarios using the DSE-100D. First we tested how many watts the Serial Device Server consumed while it was idle. Second we tested how many watts the Serial Device Server consumed while `rtnt` was running. Third we tested how many watts the Serial Device Server and a MicroZ consumed together while running `rtnt`. The following graphs display our results:



The blue line represents the wattage consumption of the Serial Server while idle. The red line represents the wattage consumption of the Serial server while we run `rtnt`. Below is a summary of the data sets collected for these two graphs:

Statistic	Blue Graph	Red Graph
Minimum	3	3
Median	3.6	3.7
Mean	3.605	3.698
Max	3.7	3.8

The graph below describes the wattage consumption of the Serial Device along with a MicroZ while rtnt is running.



Below is a summary of the data set we used for this graph.

Statistic	Red Graph
Minimum	0
Median	11.8
Mean	11.4
Max	13.5

7.0 Future Testing of the DSE-100D

The following are different scenarios we would like to consider in the future use of Serial Device Servers for real time collection of GPS data.

7.1 Freewave Radios and Serial Device Servers

Several GGN sites employ freewave radios that allow for a wireless communication between the GPS site and our main data collection center. A current site where a pair of freewave radios is being used is at our Marshall Field testing site in south Boulder, Colorado. Our next task will be to attach a Serial Server to a freewave radio and study the efficiency of the real time data collection.

7.2 Dealing with Power Outages

Many of our GPS sites have periodic power outages. We would like to develop a way to either always maintain power to the Serial Device Server or have a way in which communication can resume automatically after a power failure.

7.3 Dealing with broken Ethernet Connections

We would like to figure out how we automatically resume an Ethernet connection upon losing it.

7.4 Setting up Serial to Ethernet Tunnel using two Serial Device Servers

The following diagram should be rather simple to setup and test. This will be one of the next Serial Device Server setups we will examine.

