

Webster2 – A low power, personal web server

Craig A. Lindley – 08/02/2008

In an attempt to lower my energy consumption and carbon foot print I decided about a year ago to build myself a low power web server for use in hosting my personal web site. Previously I had used an old desktop PC as a host but I knew it consumed way more energy (100x even without a monitor) than what I had in mind.

My quest began with the design of a web server based upon an 8 bit PIC micro controller running at 20MHz (this design was published in the July 2008 issue of Nuts and Volts magazine). The design consumed little power but was not fast enough to be truly practical. The first generation Webster was like the little engine that could. It tried really hard to push out media rich pages but it took a long, long time to do so.

The second generation of this idea used a faster 16 bit PIC processor with built in ethernet hardware also running the Microchip TCPIP stack. Even with an increased clock rate and word length this design also was quickly dismissed as inadequate.

For generation three, I employed a NetBurner, Inc. (NB) MOD5270 module coupled with custom electronics which included a power supply and an SD card interface. This design had all of the features and performance I needed and was used successfully for my web site for 8 months. Unfortunately this web server was terminally damaged when I tried to change out a part that had failed.

Knowing I had to build yet another web server to replace the broken one and being very happy with the NB hardware, I created the forth generation Webster which I call Webster2. It is called this because it is the second of the designs published. In this iteration I did away completely with custom hardware and based the design on the NB MOD5270LC development kit. In other words, Webster2 is hosted on stock NB hardware which makes building the web server a breeze as you will see.

There are many uses for Webster2 like web servers including:

1. As a personal web server.
2. To provide a web presence for a business for publishing hours, directions, etc.
3. As a toy for children who want to have their own web site.

Webster2's features include:

1. Low power operation. Only 250 mA is required at 3.3 volts or about 0.8 watts.
2. Small size. The packaged web server is about the size of a hardbound book.
3. 100 baseT Ethernet connectivity.
4. Support for static and dynamic IP address assignment
5. Based upon a 32 bit ColdFire processor running at 147 MHz.
6. Support for basic authentication for web site access control.
7. A built in SD memory card interface (64 GByte max) for web page file storage.

Webster2 – A low power, personal web server

Craig A. Lindley – 08/02/2008

8. A built in FTP server allowing for in place update of web site content
9. Easy to build. Webster2 runs on stock, inexpensive NetBurner, Inc. hardware.

Note: at present the Webster2 web server software is configured to serve static HTML pages only and for this reason cannot be used for applications that require write access to the underlying file system. This was a conscience design decision that could easily be changed if your application requires it.

Webster2 and Me

I use Webster2 for my personal web site for numerous reasons. Besides the energy conservation issue, Webster can be put on the Internet without concern for viruses and spy-ware. Webster's software is written entirely in C++ with no underlying operating system with exploitable security flaws. Since, by design, there isn't any write access to the underlying file system, it is unlikely the web site content can be compromised. Finally, since Webster supports basic authentication, access to the web site can be controlled with a user name and password. Of course like any web server, Webster2 could be brought to its knees by a denial of service attack.

Building and maintaining a web site with Webster2 is easy. I compose web pages on my PC using the free *nvu* HTML editor. When finished, the HTML pages can be copied to an SD memory card which is subsequently moved to the server or the pages can be FTPed to a SD memory card already in place on the server. When Webster2's reset button is pressed, the new or modified web site becomes immediately operational.

It is important to note at the most basic level any web server is just a device that understands and implements the HTTP protocol; a web server doesn't influence web site content. A web server's job is to stream requested content to a user's browser whether that be HTML files, image files, PDF files, MP3's, etc. This is an important because it is the content of the served up files that may or may not be compatible with different browsers. If the web site you serve with Webster2 works with Internet Explorer but not with Firefox it is probably not Webster2 that is at fault.

The NetBurner Development Kit (NNDK-MOD5270LC-KIT)

The folks at NetBurner, Inc. have put together a great development kit package for their MOD5270 module . The kit includes everything you need to develop and test code including:

1. A network ready MOD5270 core computer module
2. A MOD-DEV-70 development board which the MOD5270 plugs into containing a switching power supply, an SD memory card socket, a real time clock chip, two RS-232 ports, 8 programmable LEDs, an 8 position dip switch and a reset button.
3. A wall wort transformer for powering the development board
4. A bunch of cables including an RS-232 cable and multiple Ethernet cables

Webster2 – A low power, personal web server

Craig A. Lindley – 08/02/2008

5. An Eclipse based C/C++ development environment for writing, compiling and debugging code running on their platforms. Included are numerous example applications that you can extract code from to get your designs up and going quickly.

NetBurner Inc's slogan is "Networking in 1 Day!" and it is absolutely true. They provide code for support of most Internet protocols making your application development almost trivial. In addition, they provide excellent customer support and have a large public forum of dedicated developers that can and do help with any problems you might encounter.

DHCP, DNS, FTP, HTTP, NTP, UDP and other Networking Acronyms

There are a number of networking protocols that come into play in Webster2's design. Space does not allow me to discuss these in detail but I wanted to give you an idea how each are used.

DHCP - Webster2 can use DHCP to obtain its networking parameters (IP address, etc) from a DHCP server on the network to which it is attached. If DHCP isn't used, the networking parameters can be set statically via entries in a configuration file.

DNS – Is like a phone book for the Internet for turning host names into IP addresses. DNS entries must be put in place so that Webster2 can be found on the Internet by name.

FTP – Webster2's software includes an FTP server which allows web site content stored on the on-board SD memory card to be updated remotely. Updates can be made using an FTP client program like FileZilla or via the publishing mechanism of your web page design software.

HTTP – The protocol of the world wide web. HTTP is the protocol used between web browsers and web servers for information transfer.

NTP – Webster2's software uses NTP (Network Time Protocol) for keeping its system clock accurate.

UDP - UDP (User Datagram Protocol) can be used by Webster2 to log web server accesses to a remote computer.

NOTE: Because Webster2 uses all of these protocols it may be necessary to configure any firewall software you use accordingly. Most importantly you must allow HTTP requests to pass through the firewall or Webster2 won't be available on the Internet. This feature is called port forwarding in some cases.

Using Webster2 on a Home Network

Before Webster2 can go on-line you must decide if static or dynamic IP addressing is to be used. Using

Webster2 – A low power, personal web server

Craig A. Lindley – 08/02/2008

DHCP is always easiest if your network permits it. For my web site, I connect Webster2 directly to my wireless DSL router and use DHCP so the router assigns an IP address automatically. This works very well. Once your web server has a public IP address as mentioned you must open up the firewall in the router to allow inbound HTTP connections so users outside your network can hit your server. How this is done depends upon your firewall software.

A problem in trying to host a web site on a home DSL connection is that the IP address assigned to your DSL router changes from time to time. This is a problem whether you are using Webster2 as your web server or a professional product like Apache. Regardless of how you access your web site when your router's IP address changes, connectivity to your web site using an old address will be lost.

A solution to this problem that allows one to host a DSL connected web site is a program called No-IP. See Resource sidebar for info. The company, No-IP, offers a free dynamic DNS and web redirection service for just this purpose. No-IP will even provide you the use of a domain name from a list of preexisting domains that you can use for free.

The No-IP client program runs as a service under Windows XP. It continually monitors the public address exposed by your router for your web server and informs No-IP's DNS servers of any changes. So as long as access to your web site is via your chosen domain name, you are insulated from changes to your router's IP address. By running No-IP on any PC in your network, access to your web site will be maintained even if the public IP address of your router/server changes. Pretty slick huh?

Once you register your web server host with No-IP and your server is operational, you should be able to ping your server:

```
ping myhost.mydomain.com
```

Executing this command will verify whether *myhost* in the domain *mydomain.com* is operational or not. You would need to change the ping command to check your server in your domain.

Once you have web content in place and you can ping your server, you should be able to access your web site with a browser (Internet Explorer, Firefox or whatever) by specifying a Universal Resource Locator or URL of the form:

<http://hostname.domainname>

If you don't specify a specific HTML file to access, the file *index.html* will be assumed. Once this works, you are good to go.

Webster2 Hardware

As mentioned, Webster2 is based entirely on the NB MOD5270LC development kit hardware. As such, only two things are required to get Webster2 up and running. First, you need to insert an SD memory

Webster2 – A low power, personal web server

Craig A. Lindley – 08/02/2008

card (64 GBytes max) into the socket on the development board and second you need to download the web server code provided with this article. How downloading is done will be described shortly.

While packaging of the NB hardware is not required for its operation I decided to package the development hardware to protect it from accidental abuse or damage. At my local electronics shop I found a plastic box made by Serpac (see parts list) that the hardware would easily fit into. See photos one and two to see how the packaging was done.

Since I was going to hide the hardware inside the box I thought it necessary to mount important controls and indicators on the outside of the box. In the end I mounted a power switch, power indicator, reset switch and access indicator on the box. A summary of packaging steps is given below:

1. Lay the NB module into the box with the dip switch facing the front of the box to visualize how things will fit.
2. Remove any of the box's molded standoffs that are in the way using a drill bit larger in diameter than the standoff. The idea is that the NB hardware should be supported in the box by the acrylic feet attached to the bottom of the printed circuit board (PCB).
3. With the PCB held in place, mark the box with the position of two or more of the mounting holes on the NB PCB. Drill these holes for #6 machine screws.
4. Layout the positions of the two switches and the two LEDs on the bottom front portion of the box. Drill holes of the appropriate sizes for mounting. Glue the LEDs into place and mount the switches.
5. File two holes in the top rear portion of the box to allow the power and network cables to protrude.
6. Mount the NB PCB into the box using appropriate screws. The screw heads should be on the outside of the box; nuts on the inside. Tighten the screws but not so tightly as to cause the PCB to bend.
7. Carefully unsolder surface mounted LEDs 8 and 9 from the PCB. Unsolder and remove the reset switch as well. Led8 is the furthest right in the row of 8 LED. Led9 is located to the left of the dip switches. The reset switch is located above the row of 8 LEDs.
8. Cut the connector off of the supplied power cable. Pay attention to the polarity of the cable. There was a white strip on the + side of the cable on the unit I used. If you cannot tell which is the + side of the cable use a multimeter to determine it.
9. You will probably have to extend the power cable by soldering other wires onto it. A six inch extension will probably do it. Again be careful of the polarity.
10. Solder the – end of the power cable to the GND terminal on the PCB. Solder the + end to one side of the power switch.
11. Solder one side of another wire to the other switch connection and the other side to the “7to18V” terminal on the PCB.
12. Determine the + side of the power LED by holding the leads to a 9V battery and noting

Webster2 – A low power, personal web server

Craig A. Lindley – 08/02/2008

which orientation causes the LED to light. Note which lead of the LED is connected to the + side of the battery.

13. Solder a wire from the + side of the power LED to the + connection on the PCB for Led9 which was removed earlier. Solder another wire from the other lead of the LED to the other connection for Led9.
14. If you are feeling lucky, plug in the wall wort and turn the power switch on, if all is well you should see the power LED come on. Turn power back off and unplug the wall wort.
15. Solder two wires from the reset switch to the two pads on the PCB from which the reset switch was removed. Polarity is not important here.
16. Determine the + side of the access LED as you did for the power LED.
17. Solder a wire from the + side of the access LED to the + connection on the PCB for Led8 which was removed earlier. Solder another wire from the other lead of the LED to the other connection for Led8.
18. Connect an ethernet cable to the RJ-45 connector on the MOD5270 module and route it and the power supply cable out of the box.
19. Place the top on the box and screw it to the bottom using the screws provided.

You are now finished with the hardware. Plug Webster2's power cable in and plug the network cable into a network port. Flip the power switch to on and you are ready to go.

Webster2's Software

You will need some of the tools provided by NB on the CD that comes with the development kit to download the web server software to your server. So go ahead and install the development kit on your computer. Also, go out to the DDJ web site and get the zip file containing the Webster2 code. Unzip the Webster2 code into a directory of your choice.

First it is necessary to verify your server is running and available via the network. The easiest way to do this is to bring up the NB IPSetup program. If all is well you should see a MOD5270 device listed in the right hand pane and see the networking parameters your server has adopted in the left. If you don't see your server, make sure the server is plugged in, the power switch is on, the power indicator is lit and the network port you plugged into is active.

Once you can see your server, bring up the NB AutoUpdate tool. The IP address you saw in the IPSetup program should again be displayed. Click the browse button and navigate to the directory you unzipped the Webster2 code into. Find the file, *Webster2_APP.s19*, in the release directory. Click the Update button and the web server code will be transferred to your server.

Before Webster2 can be used it must be configured. Webster's operation is controlled by a configuration file called *config2.dat* that must reside in the root directory of the server's flash drive.

Webster2 – A low power, personal web server

Craig A. Lindley – 08/02/2008

Webster will blink all eight on board LEDs and halt operation if the configuration file is not found or if an error is detected in its format. Below is a typical configuration file.

```
// Webster2 Configuration File - Version: 1.1
// NOTES:
// 1. Case is significant
// 2. No blank lines allowed; Use // for comment lines and formatting
// 3. Be careful with spaces around usernames and passwords
// 4. Don't use tabs
//
Realm Name: Your web site's Name
Hostname: YourHostname
Use DHCP: yes
//
Use Authentication: no
Username: roy
Password: rogers
Pages Requiring Authentication: page1.HTML,page2.HTML,page3.HTML
//
Use FTP Security: no
FTPUsername: rogers
FTPPassword: roy
//
Use NTP: yes
NTP Servers: 207.200.81.113,164.67.62.194
Time Zone: -6
Use DST: yes
//
// Static IP Addresses - Only necessary if not using DHCP
//
Static IP Address: 192.168.0.32
Static Subnet Mask: 255.255.255.0
Static Gateway Address: 192.168.0.1
Static DNS Address: 205.171.3.65
//
// Use UDP Logging
//
Use Logging: yes
Logging Hostname: your logging hostname
```

The configuration file is a human readable text file with each line terminated by carriage return and line feed characters. The code (in *config.cpp*) that parses this file is not very forgiving so it is important to not deviate from the format shown.

Most entries in the configuration file are probably self explanatory. The table below gives the details.

<i>Configuration Tag</i>	<i>Value</i>	<i>Notes</i>
--------------------------	--------------	--------------

Webster2 – A low power, personal web server

Craig A. Lindley – 08/02/2008

//	Comment tag	All text from the tag to the end of the line is ignored.
Realm Name	Name assigned to your web site	The Realm Name will be displayed whenever a page is authenticated.
Hostname	Name assigned to the web server host	This name will be associated with web server host
Use DHCP	yes or no (must be lower case)	If yes, the web server gets its IP address from a DHCP server on the network. If no, all of the static entries in the configuration file are used instead.
Use Authentication	yes or no (must be lower case)	If yes, a user must be authenticated to access some portions of the web site. If no, the web site is wide open. If authentication is being used, users will be presented with a login dialog box where she/he must specify the correct username and password to gain access to the web site.
Username	The authenticated user's name (case is significant)	This entry is only important if authentication is being used.
Password	The authenticated user's password (case is significant)	This entry is only important if authentication is being used.
Pages Requiring Authentication	A comma separated list of HTML pages that require authentication to view.	Whenever users navigate to a page in this list they will be required to authenticate. Once the session is authenticated, users can visit any page of the web site.
Use FTP Security	yes or no (must be lower case)	If FTP security is turned on, a username and password will be required to access the FTP server built into Webster2. This feature helps keep web site content secure.
FTPUsername	The authenticated user's name (case is significant)	This entry is only important if FTP security is being used.

Webster2 – A low power, personal web server

Craig A. Lindley – 08/02/2008

FTPPassword	The authenticated user's password (case is significant)	This entry is only important if FTP security is being used.
Use NTP	yes or no (must be lower case)	If yes, Network Time Protocol (NTP) requests will be made across the Internet to get the current time.
NTP Servers	A comma separated list of server IP addresses that support the NTP protocol	Each of these IP addresses will be tried in turn to get the current time. Webster2 system time will be set to the time value returned.
Time Zone	Value in hours	Offset from UTC time for server's location.
Use DST	yes or no (must be lower case)	Determines whether day light saving time is to be used or not.
Static XXXX:	Static network values	All of the static entries must be filled in correctly if static network addressing is to be used.
Use Logging	yes or no (must be lower case)	If yes, SysLog calls in the web server code will be sent via UDP.
Logging Hostname	Name of machine to receive logging data	If Use Logging is yes and Logging Hostname is specified, UDP packets will be sent to the specific host. If Use Logging is yes and no Logging Hostname is specified, UDP packets with logging information will be broadcasted.

You have two ways to make configuration changes. Either you can remove the SD memory card from the server and connect it to your computer via an SD card reader and make changes to the *config2.dat* file directly or you can edit a local copy of *config2.dat* and use FTP to update the server's configuration. Either way, once the configuration changes have been made and the reset button hit, Webster2 should come back up with the new configuration operational.

Once you have determined your configuration requirements and have them coded in the *config2.dat* file you can start working on your web site's content. Remember you should have a file in the root directory named *index.html* or *index.htm* as this will be the entry point for your web site. Other than the index file how you arrange your HTML files and other web site content is up to you. Remember you

Webster2 – A low power, personal web server

Craig A. Lindley – 08/02/2008

have a full FAT file system on your web server so filename case is important and long filenames are permitted. Also you can organize your files into subdirectories if that fits your style.

In this iteration of the web server software I wanted to have some form of logging so I could see when and by whom my web site was accessed. My first inclination was to use a log file in the file system but I rejected this approach as flash memory has a large but finite number of write cycles. Writing to a log file for every web site access could damage the flash over time. Instead I decided to log across the network to another machine. I used NB's SysLog function in the web server code and I use NB's UDP Terminal Tool on the remote machine to see all accesses to my web site. It is pretty cool. System logging uses UDP port 514.

Within the GET processing code in the web server, I turn Webster2's access LED off and on. This causes this LED to flash whenever my web site is accessed.

Finally, if you want to modify and/or extend the Webster2's web server software I provide, you need to create a new project within the Eclipse development environment and import all of my code. With this as the starting point you can make all of the changes you desire.

Conclusions

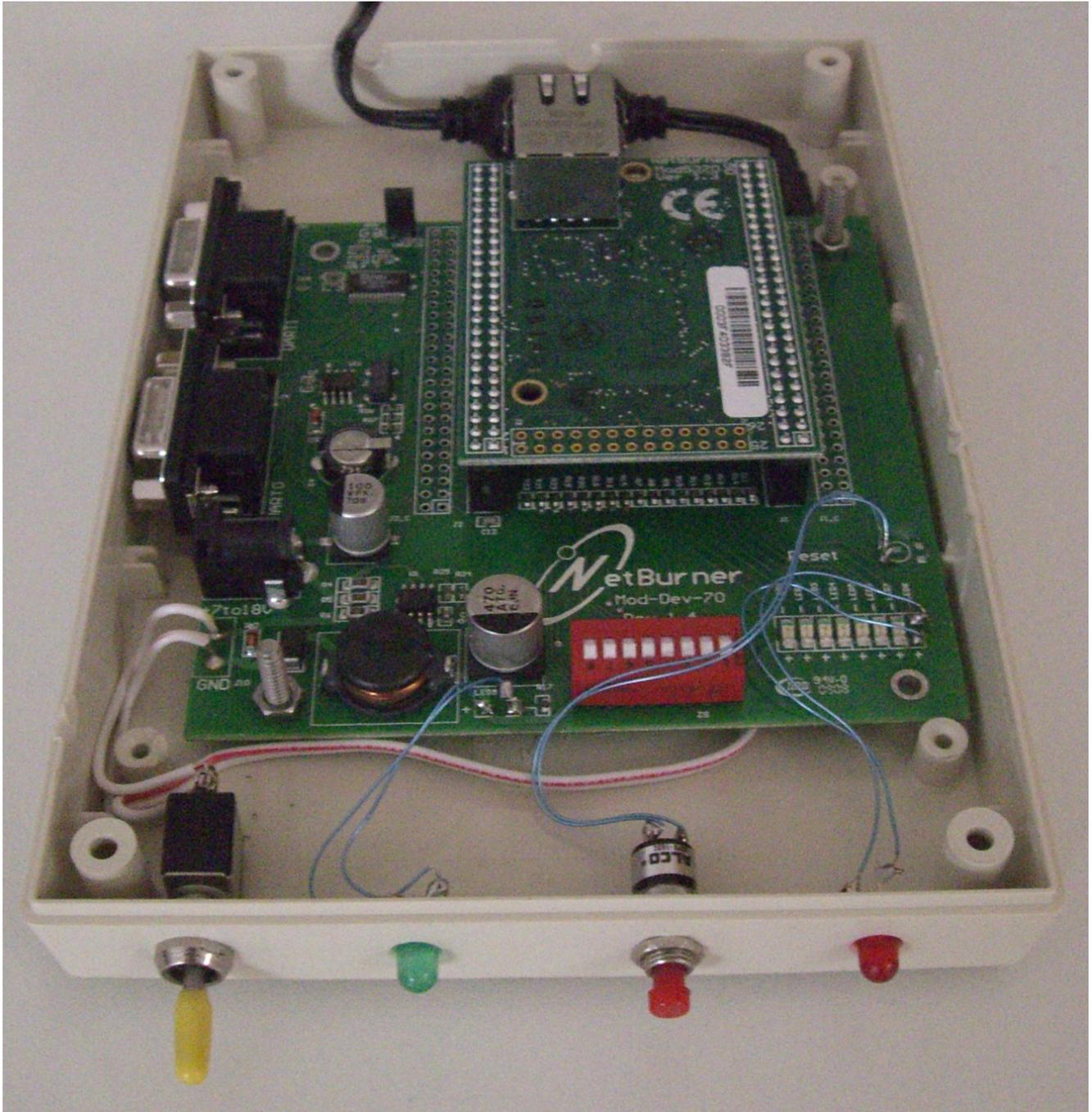
Webster2 was fun and easy to build and will help me conserve energy. Webster2 is the 4th generation web server I have designed and built and is fully capable of supporting all my current web server needs. It has the stability and performance necessary for serving pictures, large documents, music and video files for my family and friends.

As a plus, the Webster2 code is fully hackable as you have access to the development environment, the web server source code and all of the NetBurner example programs. No telling what bells and whistles could be added.

Webster2 will cost you about \$130 to build but you will have the satisfaction of knowing you did it yourself. That can earn you some serious bragging rights with your high tech buddies.

Webster2 – A low power, personal web server
Craig A. Lindley – 08/02/2008

Photo 1 – Inside the Webster2 chassis



Webster2 – A low power, personal web server

Craig A. Lindley – 08/02/2008

Photo 2 – The finished Webster2 Web Server

From the left are the power switch, the green LED power on indicator, the pushbutton reset switch and the red access LED. Hole are filed in the rear of the chassis to allow the power supply and network cables to pass through.



Webster2 – A low power, personal web server

Craig A. Lindley – 08/02/2008

Webster2 Parts List

Designation	Value	Notes
MOD5270LC	NB development kit hardware. Part number: NNDK-MOD5270LC-KIT	Available directly from NetBurner, Inc.
Box	Serpac Model #271 or #271-I Dimensions: 7" x 4 3/4" x 1 3/4"	See www.serpac.com
SPST power switch	Any brand off/on switch	I used a miniature toggle switch
SPST momentary contact pushbutton switch	Any brand pushbutton switch	This is the server's reset switch
LED power on indicator	Any brand of LED	I used a green LED for the power on indicator.
LED access indicator	Any brand of LED	I used a red LED as the access indicator
Memory Card	SD memory card	64 GB max - 1 GB usually adequate
Misc. hardware	Screws, nuts and spacers, etc.	For mounting the NB module

Resources

1. Information about the NetBurner NNDK-MOD5270LC-KIT development kit can be found at: www.netburner.com.
2. Information about the No-IP program can be found at: www.no-ip.com.
3. Code for the web server presented in this article can be found at: www.ddj.com.
4. Info on the NVU web authoring system can be found at: www.nvu.com
5. The article on the first generation of Webster web server was published in the July 2008 issue of Nuts and Volts Magazine. See www.nutsvolts.com.
- 6.

Author's BIO

Craig A. Lindley is a degreed hardware engineer who, until recently, had been writing large scale Java applications. Craig has authored over 35 technical articles on various programming topics and has published five books dealing mostly with multimedia. He can be contacted via email at: calhjh@gmail.com.

Craig A. Lindley has no affiliation with NetBurner, Inc. other than being a very satisfied customer.