NEC

User’s Manual

# V850 WebServer

## Embedded Web Server Technology
## Demonstration Kit

## Software

Document No. U15888EE1V0UM00
Date published: October 2001

# Table of Contents

# 1. Introduction

## 1.1. Embedded Systems and the Internet

In recent years the Internet has become a more and more important instrument for communication. It is built on standard protocols defined over the last 30 years, so that now any computer can be connected to the World Wide Web.

Currently most computers with Internet access are PCs running large operating systems. Now, however, with the entry of embedded systems into the Internet world, there has become a definite requirement for solutions without the traditional overheads of PC connectivity.

## 1.2. Remote Access Applications

Internet connected embedded systems have become key to remote access applications. Up to now, remote access applications have been based on high-cost, inflexible proprietary solutions to provide

- the communications protocol between a remote location and the embedded system
- the embedded system software
- the graphical user interface at the remote location

If it were possible to use the standard TCP/IP and additional protocols used in the Internet, development costs for remote access solutions would be dramatically reduced:

- the communications protocols are standard: PPP, Ethernet, TCP/IP, UDP, HTTP, FTP
- the graphical user interface is available on every PC without costs: a standard Internet Browser (e.g. Netscape Communicator or Microsoft Internet Explorer).

So it has become necessary to provide a communication software solution for microcontrollers which supports the protocols mentioned above.

## 1.3. Embedded Web Server

To fit these demands , a Web server has to become “Embedded”. An Embedded Web Server works with minimal resource requirements, handles the protocols and provides the low-level functions required to access the hardware.

The product described in this documentation is a hardware and software reference design to demonstrate the NEC V850E family of microcontrollers as a platform for Embedded Web Server technology.

# 2. Product Functions

## 2.1. Destinations of the implementation

The Web Server software makes it easy to demonstrate the V850E in an Internet/Intranet context. With limited knowledge of HTML language, it is also possible to create custom web pages, providing a tailored demonstration platform, specific to the needs of individual companies. An example data logging function and the possibility for sending e-mail complete the function scope.

## 2.2. Software components

The software provided with the Evaluation Kit is based on a real-time operating system and Internet software. For the purposes of instruction within this document, Evaluation software functions are based on the Nucleus Plus RTOS from Accelerated Technologies Inc. (www.acceleratedtechnology.com), which has been proven to run with WebServer.

Details are also provided for running the WebServer Demonstration Kit alone.

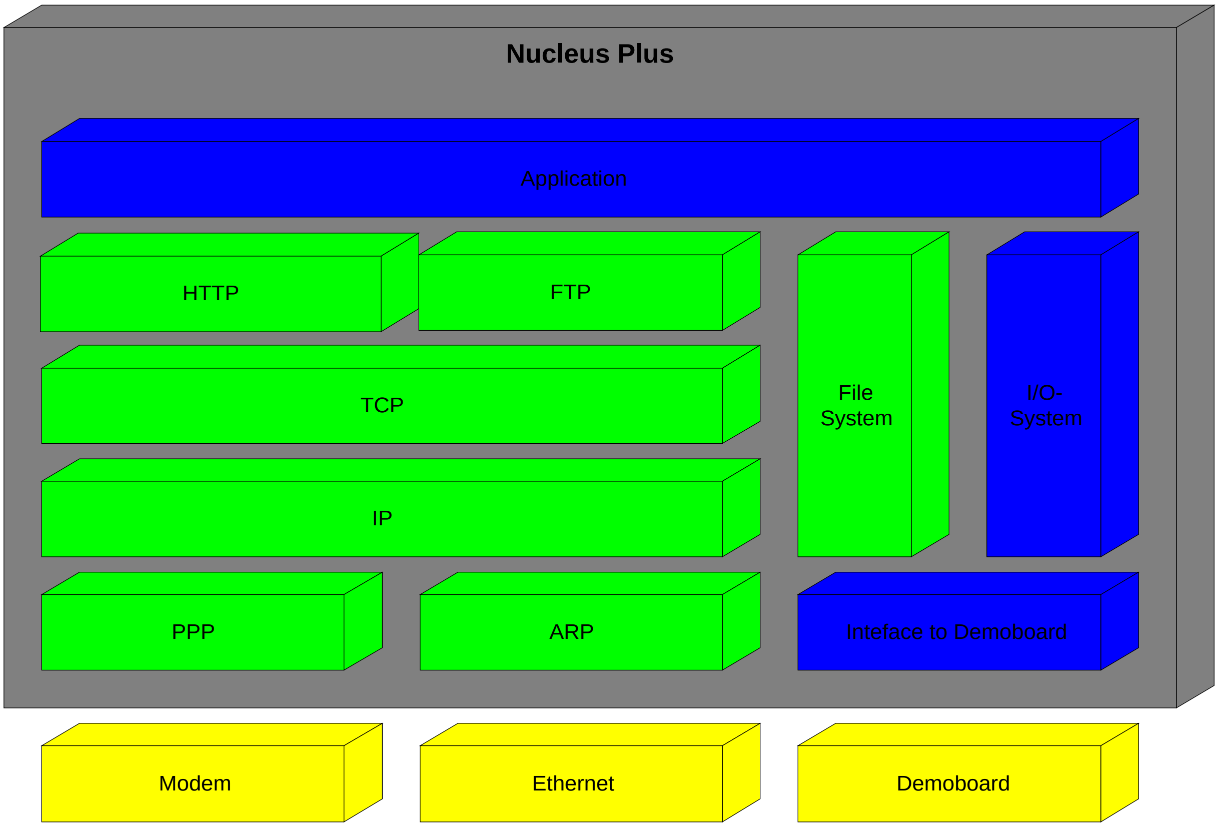
| | |
|---|---|
| Nucleus Plus | Operating System |
| Nucleus NET | TCP/IP Stack |
| Nucleus Webserv | HTTP-Server |
| Nucleus EMAIL | SMTP-Client |
| Nucleus FAL | File Abstraction Layer |
| Nucleus Extended Protocol Package | FTP-Server |

PPP (Point to Point Protocol) and the drivers for the serial interface and the Ethernet-controller are further components available royalty free from ATI.

**The source code provided with the Web Server Evaluation Kit is an example built with a modified ’In memory file system’.**

## 2.3. Structure of the software components

The software components of the WebServer are implemented in the internal flash of the V850E.Only parameter files and the HTML files are stored in the file system. The software structure consists of the components represented in the following diagram. Green signifies components supplied within Nucleus Plus, and blue represents the demonstration software.



The demonstration works with a PPP connection (Modem) and with an Ethernet connection.
It is possible to use both interfaces at the same time.

## 2.4. Summary of demonstration software functions

- Initialization of: I/Os, interrupt table, RAM disk (after reset)
- cyclical read of all analog an digital inputs
- start of data logging on a HTML request
- display of all analog and digital input values on HTML sites
- set of the digital output ports corresponding to inputs on HTML sites
- management of the disk copy function between RAM disk and flash ROM disk

## 2.5. HTTP server

The HTTP server forms the interface between the HTML files stored in the file system and the TCP/IP stack.

Nucleus WebServ receives a request from a web browser. The request is decoded, and information is returned to the client. Returned information is normally static HyperText Mark-up Language (HTML) files. The application program executing with the HTTP server may also dynamically generate information that can be sent to the client.

This program is typically called a server side plug-in, or server script. The client can also accept data from the user through HTML functions (buttons, text boxes, etc.) and pass it to the server plug-in programs. Since HTML is a simple to use text formatting language, the embedded plug-in can accept the data from the user and compose an HTML formatted reply and respond to the user in the familiar web interface.

The server-side-include (SSI) mechanism is a special case plug-in, and an easy way to make web pages with dynamic content. The Nucleus WebServ normally does the job of serving a file by copying that file to the network socket interface connected to the client. Files with a distinguishing extension (`.SSI` by default) are scanned by the server while copying them to the client for special embedded escape strings. When the special string is found, a corresponding (user supplied) server plug-in routine is called, which sends some output into the data stream being sent to the client. The net effect of this is to insert dynamic (generated on the fly) page content by using a static HTML page. This feature is usually used to insert some changing piece of system information (time of day/date, etc.) into an otherwise static web page.

The SSI mechanism is used to include information about the analog and digital inputs. This information is synchronised with the scaling information in the HTML site. In this way it is possible to write the value of the ADC as a real physical value.

With the WebServer demonstration board, a standard HTML administration page is also delivered (index.ssi). The function of this page is to administrate the web server, to set and clear the digital output ports and to start the data logger function.

**Usage of the HTTP server**

An Ethernet or PPP connection must first be set up between the V850E server and your PC.
You can check the connection with the PING command:

```
PING 192.168.200.1     for default PPP connection
PING 192.168.200.2     for default Ethernet connection
```

The name for the PPP connection is “NEC” and the password is “V850”. Please input these strings before starting the PPP connection. The baud rate for PPP is 19200bps.

If the PING works properly you can start your browser and type

```
http://192.168.200.1 or
http://192.168.200.2
```

The server sends the index.htm to the client.
The default contents of the index.htm is:

```
<HTML>
<HEAD>
<META HTTP-EQUIV=\"Refresh\" CONTENT=\"0; URL=index.ssi\">
</HEAD>
</HTML>
```

This is a meta tag requests the server to send the file index.ssi.

**HTML file “index.ssi”**

In this section we describe only the lines of the HTML file index.ssi that are not part of the default HTML.

```
<FORM ACTION="/post_demo" METHOD="post">
```

This tag defines the interface function for all post operations. The name of this function is post_demo. The registration of this function as a server plug-in is part of web server software.

**Post function**

The post_demo function scans all names of HTML user action tags.
The following names are supported:

| | |
|---|---|
| SETIO | Set of the state of the 4 digital outputs |
| STARTDL | Start of the data logger |
| SAVE | Copies the “In memory file system” into the flash area |

The type of all these input tags has to be “submit”.

```
<INPUT TYPE="submit" NAME="SETIO" VALUE="Set Ports">
<INPUT TYPE="submit" NAME="STARTDL" VALUE="Start Data Logger">
<INPUT TYPE="submit" NAME="SAVE" VALUE="Save Filesystem">
```

At the end of the post function the server sends the following meta tag to the browser.

```
<HTML>
<HEAD>
<META HTTP-EQUIV=\"refresh\"CONTENT=\"0;URL=index.ssi\">
</HEAD>
</HTML>
```

That means that the browser requests the index.ssi again.

**SSI function**

Server Side Includes are actually plug-ins that are executed on the target prior to the completion of a GET operation. The Web Browser sends a request to the Web Server in the form of a URL. The Web Server detects the “.SSI” extension on the request. It then searches the requested file for the unique “<-#” comment string. When found, the function specified by the string following the special comment symbol is generally passed some parameters (not necessary) and executed. The result is generally HTML output. This capability permits the inclusion of dynamic data, controlled by the target, in an HTML file.

The following SSI plug-ins are used in the demonstration application:

```
<!-#ad_value chn=0;0;1023;0;2.5;2>
<!-#digin_value chin=2;low;high>
```

```
<INPUT TYPE="radio" NAME="RADIOA0" VALUE="HIGH"
<!-#digin_value chout=0; ;CHECKED>>
<INPUT TYPE="radio" NAME="RADIOA0" VALUE="LOW"
<!-#digin_value chout=0;CHECKED; >>

<!-#save_fs_state >
```

The ad_value function supplies the analog value of one AD converter channel. The first parameter is the number of the channel (0..7). The next 4 parameters are for the linear interpolation and the last parameter determines the number of decimals.

```
<!-#ad_value chn=0;0;1023;0;2.5;2>
         | | |  | |  |
         | | |  | |  2 decimals
         | | |  | 2.5 value that corresponds with 1023
         | | |  0 value that corresponds with 0
         | | 1023 second value of the ADC for interpolation
         | 0 first value of the ADC for interpolation
         0 number of the ADC channel
```

In this example the HTML site shows values from “0.00” to “2.50”.
The resolution of the ADC is 10 bit.
For the interpolation only positive values are allowed.

The digin_value function supplies one string in the case of a low level, and another string in the case of a high level of a digital port. The input ports are queried with chin and the output port with chout.

The first parameter is the number of the channel (0..7).

```
<!-#digin_value chin=2;low;high>
```

In this example the HTML site shows “low” in case of low level and “high” in case of a high level on channel 2.

The following examples are special applications of the digin_value function:

```
<INPUT TYPE="radio" NAME="RADIOA0" VALUE="HIGH"
<!-#digin_value chout=0; ;CHECKED>>
<INPUT TYPE="radio" NAME="RADIOA0" VALUE="LOW"
<!-#digin_value chout=0;CHECKED; >>
```

The SSI function completes the tags for radio buttons corresponding to the actual value of a digital output port. The number of the port can be 0..3. In one case the function supplies a blank and in the other case “CHECKED” for a checked radio button.

The save_fs_state function supplies the string “saved” or “not saved” corresponding to the file system. Press the “Save file system” button to see the difference.

```
<!-#save_fs_state >
```

## 2.6. FTP server

The FTP server is used for receiving files from one client and storing files in the file system as well as sending files on request.
With the “In memory file system” the following FTP functions are possible to use:

```
Dir
Put file name
Get file name
Del file name
Bye
```

Use of the FTP function running in a DOS window is recommended.

Example for starting the FTP client:

```
C:\work_dir\>FTP 192.168.200.1 <CR>
FTP>
```

It is not necessary to input a username or password.
You can now work in terminal mode.

## 2.7. The mode of operation of the file system

The Nucleus Plus „In Memory File System“ consists of 2 layers.
The lower layer manages the memory IC as a file system.
The upper layer, the file abstraction layer, sets up the connection between the file system and all other software components.
The Nucleus Plus file system uses RAM as storage.
The number of files within in the file system will be fixed at the time of compilation.
(8 files with 16 Kbyte filesize)
Only one directory is supported.

When initialised, the application copies the file system from the Flash ROM into the RAM. If there is no valid file system in the flash ROM or the “INIT button” is pressed the application initializes the file system and writes the files that are necessary for proper operation of the WebServer.
If the file system in the RAM was changed by the user, the file system in the flash ROM can be copied by pressing a button on an HTML-page.

## 2.8. Operation of the data logger

The data logger has the task to read in the values of the analog input ports in a fixed time period and to store these data in the file “DATA.LOG”. The time period is committed to 10 seconds.
The data logger function can be started via an HTML page. The function stops if the file is filled up.
The format of the data is ASCII. So a read in into Excel is easy possible.
The file transfer will be made via file transfer protocol (FTP).

## 2.9. Interfaces to the hardware

The hardware interfaces are used in the following manner:

8 digital inputs ports
4 digital outputs ports
8 analog inputs ports

A serial interface is reserved for communication with the modem. A second serial interface is used to debug the system. For the function of the demo application it does not have any importance.

## 2.10. Construction of the file SERVER.INI

The file SERVER.INI includes all information on the configuration of the server and the application.
The format of configuration file is ASCII, allowing editing within any text editor.

Determine the IP addresses of the web server:

```
PPP-IP:192.168.200.1
Ethernet-IP:192.168.200.2
```

Determine the subnet mask:

```
Subnet:255.255.255.0
```

To reset the “SERVER.INI” into the reset state, hold down the init button while turning the power on.
The init button is the only button present on the credit-card sized server board.

## 2.11. Demonstration program V850E_web.c

The V850E_web.c file contains all functions of the demonstration software.
For memory access of the eeprom and the external flash memory two additional modules are available.

| | |
|---|---|
| Flash.c | Access to the external 1 Mbytes flash memory |
| Eeprom.c | Access to the Ethernet controller e2prom (93C46) |

Summary of the functions in the V850E_web.c module:

| Function name | Short description |
|---|---|
| INB | Reads 1 byte from Ethernet controller |
| OUTB | Writes 1 byte into Ethernet controller |
| INW | Reads 2 byte from Ethernet controller |
| OUTW | Writes 2 byte into Ethernet controller |
| Toggle_watchdog | Toggles the watchdog (only necessary if watchdog is enabled) |
| Copy_str | String copy with termination character and maximal length |
| Application_initialize | Initialization of all tasks and memory pools |
| Init_ad | Initialization of the AD-converter |
| AD_convertion | Converts of all 8 channels of the AD-converter |
| SetPortBD | Sets one of the 4 output ports |
| GetPortBD | Reads the state of one port of the 4 output ports |
| GetPort0 | Reads the state of one port of the 8 input ports |
| Check_key | Reads the state of the init key |
| inpol | Interpolation for the AD-converter values |
| Ad_value | SSI function for the representation of AD-values on HTML sites |
| Digin_value | SSI function for the representation of digital ports on HTML sites |
| Save_fs_state | SSI function for the representation of the state of file system |
| Copy_INF_to_flash | Copies the IMF from RAM into flash |
| Post_demo | POST function for the index.ssi HTML site |
| Load_IMF_from_flash | Copies the file system from flash into RAM |
| Create_server_ini | Creates the file SERVER.INI if it not exists |
| Create_index_html | Creates the file index.htm if it not exists |
| Read_server_ini | Reads and interprets the contents of the file SERVER.INI |
| Tcp_server_task | Initialization of all network and file components |
| Ad_to_ascii | String converter of AD-converter values for the data logger |
| Main_task | Task for the data logger and life LED |