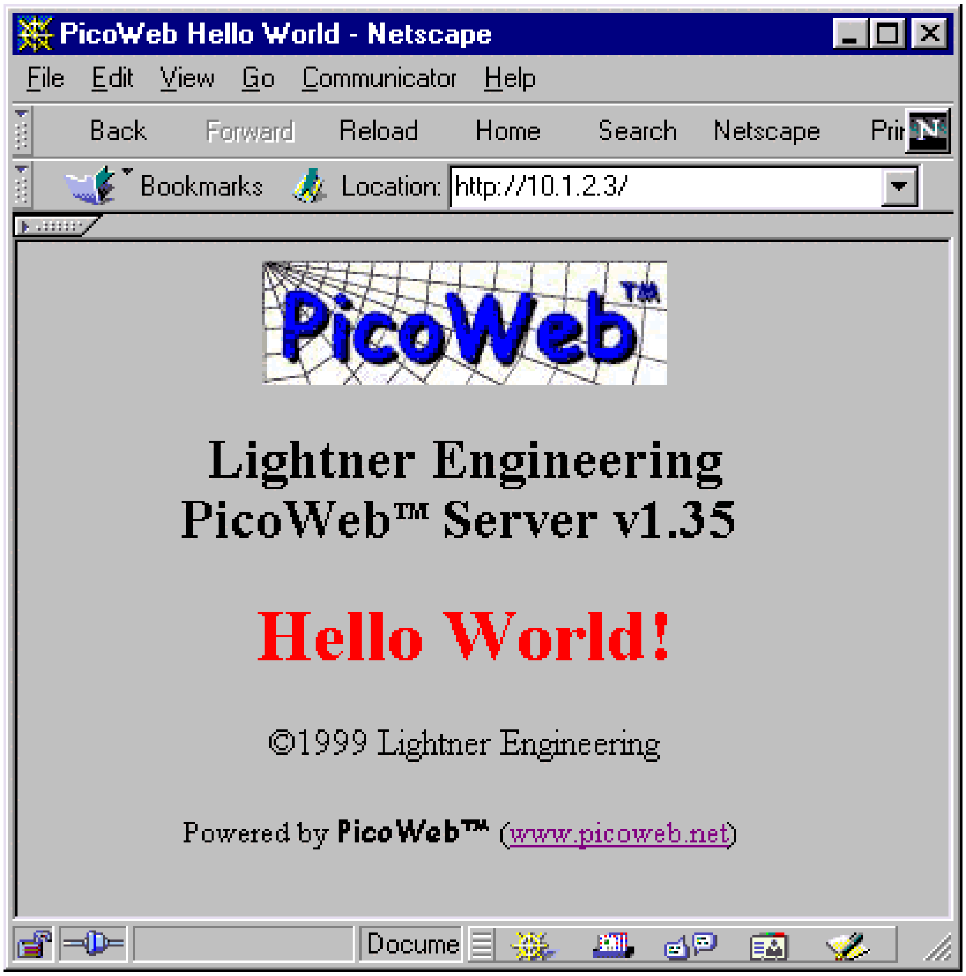
PicoWeb “Hello World” Project

This PicoWeb project can be found in the Picoweb development system “samples” directory in a file named `samples\hello\hello.pwp`. This project consists of a single HTML document or Web page (`hello.htm`) which references a single JPEG image (`picoweb.jpg`).This project sets up a PicoWeb server to display the “home page” shown in Figure 1. No user-supplied application-specific CGI code is required to support this very simple PicoWeb project.


PicoWeb Hello World - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Prir
Bookmarks Location: http://10.1.2.3/
PicoWeb™
Docume

Lightner Engineering
PicoWeb™ Server v1.35

Hello World!

©1999 Lightner Engineering

Powered by PicoWeb™ (www.picoweb.net)

Figure 1- PicoWeb Hello World Project Web Page

Project File - The Hello World project file `hello.pwp` is shown in Listing 1. The project file calls out two “Web server files”:

- `hello.htm` - the HTML code which produces Figure 1
- `picoweb.jpg` – the JPEG image with the PicoWeb logo shown in Figure 1

These are the files which will be stored in the PicoWeb serial EEPROM memory as part of the Web server’s “file system”

```
//---------------------------------------------------------------------
//
// PicoWeb Project File for Simple Hello World! Web Page
//
//---------------------------------------------------------------------
//
// application-specific preprocessor definitions
//
#define BANNER "\r\nPicoWeb Hello World!\r\n"
#define EEPROM_IP         /* use file "ip" for default IP address */
#define NET_CONFIG_IP     /* allow IP address reconfiguration via net */
#define ENABLE_WATCHDOG   /* use Atmel watchdog timer hardware */
#define DEBUGGER          /* include debugger firmware */

#define CLOCK 7372000     /* processor clock rate */
#define BAUD_RATE 19200   /* serial port baud rate */

//
// application-specific HTML and image file names
//
hello.htm       // ii00  (default Web page or ”home page”)
picoweb.jpg     // ii01  (PicoWeb logo image)

//
// included application-specific pcode and/or AVR assembly language follows
//
#avr_reset
;---------------------------------------------------------------------------
; this code is executed each time microcontroller is reset
;---------------------------------------------------------------------------
;

#avr_slow
;---------------------------------------------------------------------------
; this code is executed each trip through "slow idle" loop (~1 sec period)
;---------------------------------------------------------------------------
;

#avr_fast
;---------------------------------------------------------------------------
; this code is executed each trip through "fast idle" loop
;---------------------------------------------------------------------------
;

#avr_asm
;---------------------------------------------------------------------------
; application-specific CGI pcode and AVR assembly routines go here
;---------------------------------------------------------------------------
;
```

Listing 1 – PicoWeb “Hello World!” Project File (`hello.pwp`)

The project file `hello.pwp` also contains several preprocessor #define statements, as follows:

- `#define BANNER` – specifies a text string that is printed each time the PicoWeb server is reset
- `#define EEPROM_IP` – specifies that the PicoWeb IP address is stored in on-chip EEPROM (the address will come from the file `ip`)
- `#define NET_CONFIG_IP` – enables changes to PicoWeb IP address using the program `setip`
- `#define ENABLE_WATCHDOG` – enables the watchdog timer hardware on reset
- #define DEBUGGER – includes the serial port debugger firmware
- `#define CLOCK` – specifies the clock rate of the Atmel microcontroller (in Hz)
- `#define BAUD_RATE` – sets the baud rate for the serial port (used by the debugger)

```
// HTML file for PicoWeb "hello" project
<html>
<head>
<title>PicoWeb Hello World</title>
</head>

<body text=#000000 bgcolor=#c0c0c0>

<center>

<p><img src="picoweb.jpg" alt="PicoWeb Logo" width=175 height=54></p>

<h2>Lightner Engineering<br>
    PicoWeb<sup><font size="-2">TM</font></sup> Server v$$VERSION$$
</h2>

<h1><font color="red">Hello World!</font></h1>

<p>&#169;1999 Lightner Engineering</p>

<p><small>Powered by <b><font face="Comic Sans MS">PicoWeb&#153;</font></b>
&nbsp;(<a href="http://www.picoweb.net/">www.picoweb.net</a>)

</center>
</body>
</html>
```

Listing 2 – PicoWeb "Home Page" (`hello.htm`)

Since this PicoWeb project does not require any user-supplied CGI code, the #avr_reset, #avr_slow, #avr_fast, and #avr_asm code sections are empty. In fact, those directives could be removed from the project file.

PicoWeb HTML Coding - Listing 2 shows the contents of the project's "home page" (`hello.htm`). To those familiar with HTML, this file looks like an ordinary HTML document in every way except for the first line. Any line which begins with "//" in the *first column* is treated as a comment and that line is replaced by a *blank line* when it is prepared for down loading into the PicoWeb server's "file system".

Also note that the text string "`$$VERSION$$`" will be replaced with the current PicoWeb firmware *version string* when HTML files are processed for downloading into the PicoWeb server's "file system".

Building the Project - The Hello World project can be built using the PicoWeb development system as in the following examples. The examples that follow assume that commands are being entered into an MS-DOS Prompt Window under Windows 95/98. Note that the environment variable `PWDEV` needs to be set to the full path name of the PicoWeb development system "base directory", and the "bin" subdirectory needs to be in the `PATH`. Assuming that the PicoWeb development system "base directory" is `C:\PWDEV`, then the following commands will set up the environment for PicoWeb development:

```
C:>set PWDEV=C:\PWDEV
C:>set PATH=%PATH%;%PWDEV%\bin
```

Before beginning building the Hello World project two text files may need to be edited:

`ip` – must contain the IP address assigned to the PicoWeb server (e.g., `10.1.2.3`)
`ether` – must contain the Ethernet address assigned to the PicoWeb server (e.g., `0.1.2.3.4.5`)

After assigning IP and Ethernet address, the Hello World project can be compiled and readied for download into the PicoWeb server. This can be accomplished with the following command:

```
C:>pwbuild hello
```

This will cause the generation of a number of files, including an assembly language listing file, `hello.lst`, as well as three data files needed to download the firmware and Web pages into the PicoWeb server hardware:

`hello.rom` –	Atmel micocontroller program memory binary data (a.k.a., ROM file)
`hello.ep` –	Atmel microcontroller on-chip EEPROM program binary data
`hello.el` –	PicoWeb “Web file system” data (i.e., HTML code and images) and CGI pcode binary data to be loaded into the external serial EEPROM

Downloading the Project - The first two files can be loaded into the PicoWeb server hardware using a PC parallel port by attaching the PicoWeb programming cable to the connector on the PicoWeb server and to the PC’s parallel port, then entering the following command:

```
C:>pwavrld hello
```

This will erase the program and EEPOM memory in the PicoWeb server’s Atmel microcontroller, then download new firmware and data into the chip.

At this point, if the PicoWeb server is plugged into the network, it should now be active on the Ethernet. In fact, one should be able to “ping” the PicoWeb server from the development system PC using the PicoWeb server’s assigned IP address.

At this point, the Web pages and CGI pcode routines stored in the file `hello.el` needs to be loaded into the PicoWeb server. This can be done with the following command:

```
C:>pwnetld hello
```

which will download the data in these files over the network using the PC’s network card. This step also verifies the data just downloaded into the PicoWeb’s serial EEPROM chip by reading back the newly downloaded data.

TIP

Note that the above two download steps can be combined into one step by using the following command:

```
C:>pwload hello
```

This automatically executes `pwavrld`, followed immediately by `pwnetld`.

Accessing the Web Pages - Assuming that the above steps show no problems, the PicoWeb server can now be accessed over the network as a Web server. One should be able to retrieve the Web page shown in Figure 1 by using either of the following URLs:

```
http://xx.yy.zz.ww/
```

or

```
http://xx.yy.zz.ww/hello.htm
```

where `xx.yy.zz.ww` is the IP address previously assigned to the PicoWeb server.

Note: You may need to reconfigure your Web browser in order to successfully access the PicoWeb server if your browser currently uses a “proxy server”.