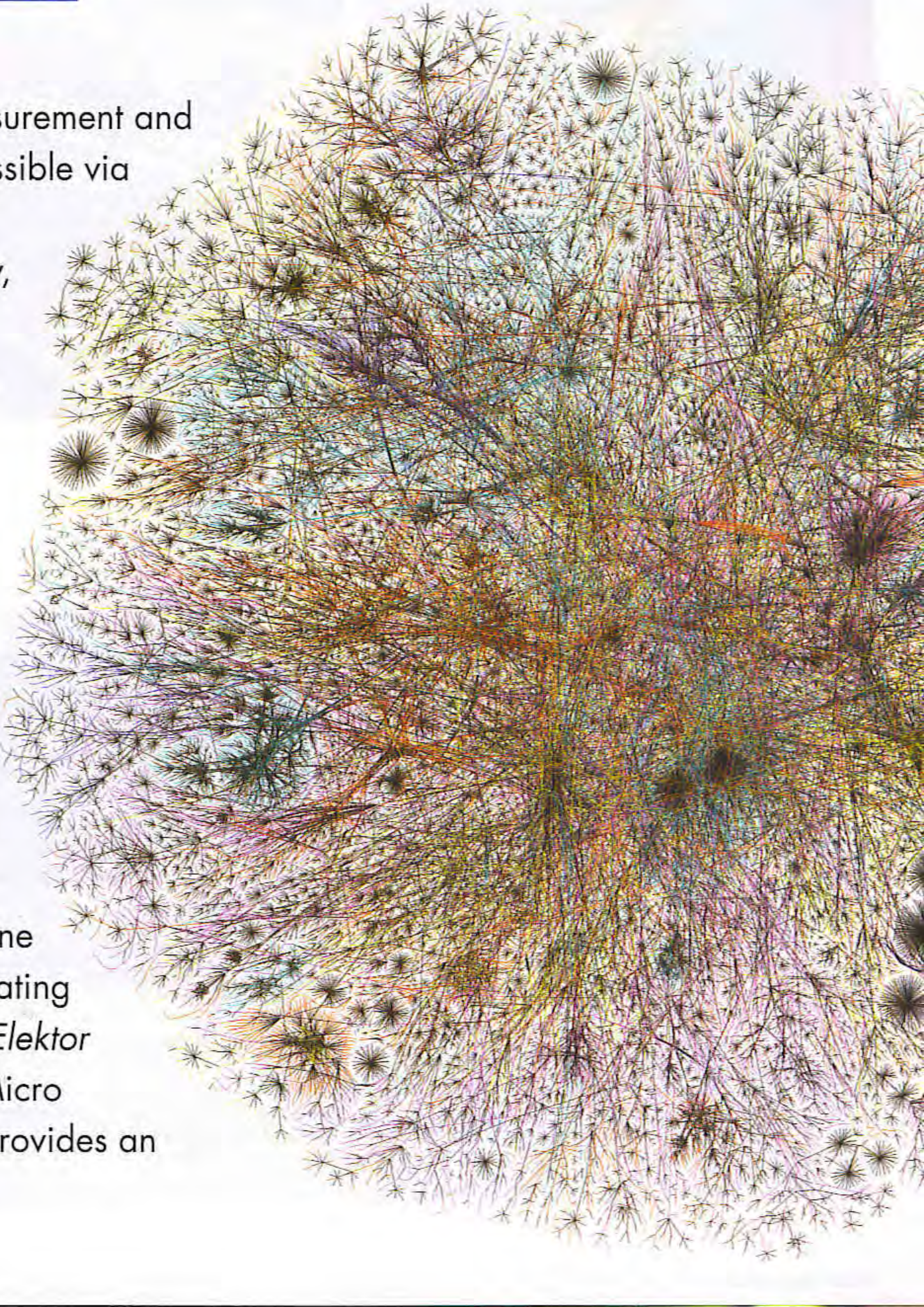
Micro Web

Jürgen Wickenhäuser

Remote measurement and control is possible via the Internet. Unfortunately, webservers usually sit in large, humming grey cabinets. That's not the ideal solution for keeping an eye on your refrigerator, coffee machine or central heating system. The *Elektor Electronics* Micro Webserver provides an alternative.



server control and regulation via the Internet

The *Elektor Electronics* Micro Webserver is a full-fledged node for Internet traffic, despite its quite modest dimensions and complexity. It consists of a microcontroller board with a network interface. Thanks to its compact construction and the versatility of the microcontroller board, the Micro Webserver is an ideal choice for measurement and control applications. Naturally, the fact that it can be read and operated from anywhere in the world via the Internet is a major bonus. Despite these unprecedented features, the necessary hardware is actually minimal. In principle, two ICs are all you need for a complete webserver. To avoid any misunderstanding, this is not some kind of demo or prototype, but a fully functional device suitable for industrial applications, and its potential uses extend far beyond what we can describe here.

Basic design

The underlying technology is rather complex. Consequently, in this article we must omit a large number of interesting details that are not essential for a 'simple' webserver. However, readers who want to know all the details will find what they're looking for in the accompanying software. The interface is without question unusually user-friendly. For example, the program variables can be used directly in websites. It's hardly possible to make things any easier.

The Micro Webserver is programmed using the C language. But don't let yourself be discouraged if you aren't familiar with C, since this project is certainly suitable for beginners as well.

Connection

Internet and Ethernet are closely related. Ethernet is a standard that defines the connection. The transmission speed is normally 10 or 100 Mbit/s, and it is automatically configured when the connection is established. We use the 10-Mbit/s variant in this project, since it is more than adequate for an embedded webserver.

We assume you already have an Ethernet network. The webserver can thus be connected directly to a hub or switch, so the Internet can be accessed via Ethernet. There are also agreed conventions regarding how Internet communication takes place (via Ethernet, for instance). All of this is specified in the TCP/IP protocol. Here we assume that the network to which the Micro Webserver is connected can also 'speak' this protocol. From a technical perspective, there's no reason why the Micro Webserver cannot also be directly connected to a PC using a crossover cable. However, describing this in more detail is beyond the scope of this article, since in some cases the PC settings must be changed for such a connection.

Hardware

After all these introductory diversions, it's time to get down to brass tacks. The hardware platform is the by now well-proven MSC1210 board (originally described in the 2003 Summer Circuits issue). If you do not already own a copy of this outstanding board, you can obtain one from *Elektor Electronics* together with the extension described here (**Figure 1**).

The extension is thus new. In principle, it's simply a 'custom' network card for the MSC board. This card is built around the CS8900A Ethernet driver IC (refer to the schematic diagram in **Figure 2**). As usual with network cards, there are two LEDs (D1 and D2) to indicate the status of the network connection. D1 flashes for 6 ms each time a data packet is received or transmitted, or if there is a collision between two packets. The second LED indicates whether the CS8900A is receiving proper link pulses. These pulses are used in Ethernet networks to synchronise transmitters and receivers, and D2 will be on if this synchronisation is successful.

The network IC also has a complete 10Base-T transceiver. 10Base-T is the standard for 10-Mbit/s Ethernet over twisted-pair cable. The circuit requires only a few external components. The transformer just ahead of the RJ54 connector provides electrical isolation from the rest of the world.

The printed circuit board (**Figure 3**) has a 'prototyping' area to provide extra space for user applications, in addition to the space on the MSC1210 board. Several spare signal lines are available in the leftmost row of the prototyping area (see Figure 2). Two extra LEDs and a pushbutton switch are also placed on the LAN board. The

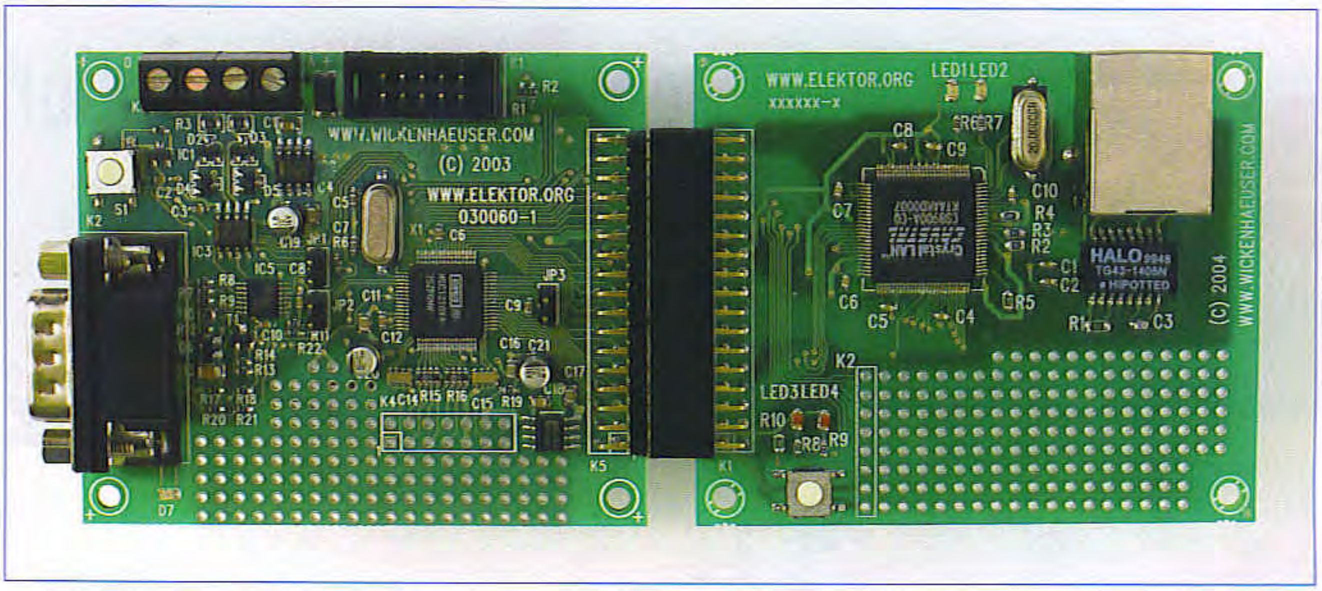


Figure 1. The MSC1210 board with the network extension: a powerful pair!

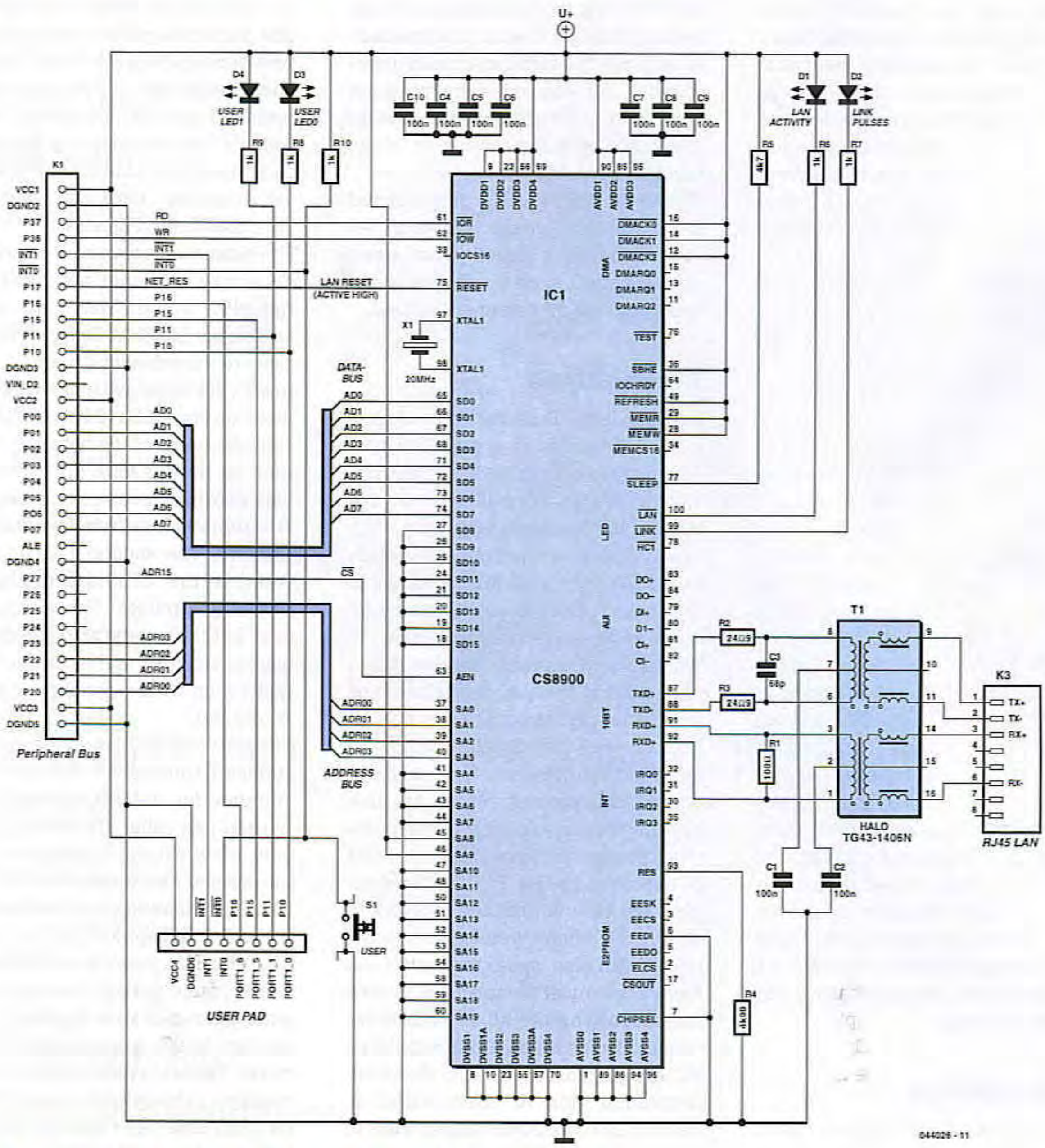


Figure 2. The network card is built around the CS8900 network IC.

Applications

The Micro Webserver is ideal for the following applications:

Automatic online weather station:
- temperature
- precipitation
- lightning detection
- wind strength and direction
- relative humidity
- rain barrel level
- light intensity

Web interface for home appliances and fixtures:
- refrigerator or freezer temperature monitoring
- remote control for coffee machine, central heating or lighting
- controlling sun awnings or roller shutters
- outside lighting
- intruder detection
- greenhouse climate control

Access control and registration in combination with:
- badge readers
- light barriers
- door openers
- RFID tags

Monitoring and controlling machinery:
- rpm
- voltage and current
- temperature
- liquid level
- flow rate / discharge rate
- pressure
- valve control
- relay control or PWM (servo) control

Terminal for a central database (in combination with an LC display and barcode reader)

placement of the connector for the link to the 'motherboard' allows the extension card to be located next to the motherboard or underneath it. In the latter case, the two boards can sandwiched together using standoff bushes.

Although the design of this project is especially simple, there is one thing that must be mentioned. The current consumption of the LAN IC is 100–120 mA, which is relatively high compared with the current drawn by the microcontroller. The 5-V supply voltage is taken from the MSC1210 board. To prevent the voltage regulator on that board from becoming overheated, we strongly recommend that the entire circuit be powered from a voltage of 7.5 to 9 V, but definitely no higher than this.

Online

There's actually not much more to say about the hardware. Configuring the board is fully described in the text box. Once you've gotten the server 'up', you can start testing.

This is where things start to get interesting. To start off, simply connect the board to the network. LED D2 will be continuously on if an Ethernet signal is detected. This is a promising start, but the real test comes next. It consists of trying to 'ping' the server using the Windows Command Prompt window (DOS command window). On a PC connected to the network, type the following command in the command line:

```
ping 192.168.1.156
```

(of course, the IP address here must be the address previously assigned to the Webserver). LED D1 should start blinking as an indication that data is being transferred via the Ethernet, and a reply from the server should appear in the command window.

Ping is a simple protocol that allows a few bytes to be transmitted and waits for an 'echo'. It's a really handy way to quickly check a network connection.

If the ping test is OK, you can then access the webserver using a web browser. In the browser window, enter the following address:

```
http://192.168.1.156
```

(use the address that has previously been assigned to the webserver). And that's it: what you see next comes from that little board (see **Figure 5**).

In the terminal download window, you can also see which page was requested.

Internet references

[1] www.wickenhaeuser.com
µC/51 compiler with source code

[2] www.mikrocontroller.info/kabelsalat/
Wiring diagram for a null-modem cable

[3] www.ti.com/msc MSC121x home page

[4] groups.yahoo.com/group/TI-MSC
MSC121x users group. Definitely worth the effort. Free, but registration is required.

[5] groups.yahoo.com/group/TI-MSC/files
You can find tools for the MSC121x here, such as the original TI downloader.

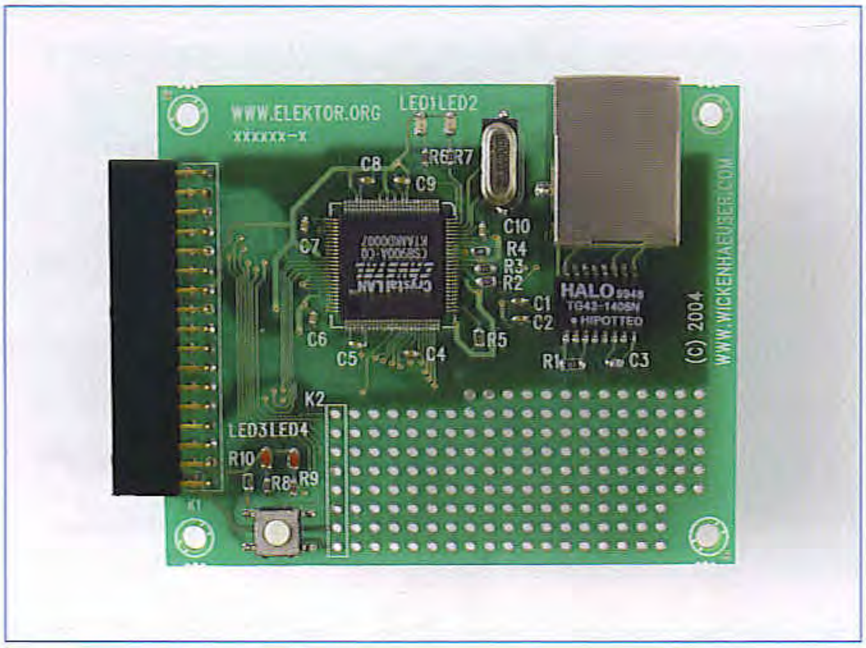
[6] www.cirrus.com/en/pubs/proDatasheet/cs8900a-4.pdf
Data sheet for the CS8900A network driver

How it works

What actually happened when you requested the web page? First, you made a connection to an IP address. Actually, it's a bit more complicated than that: you made a connection to a 'socket' at a particular address. A socket is a sort of 'connector', in this case one that only fits web links. Each socket is also assigned a specific port number. Port 80 is frequently used for webservers. You can see this in the program line `SOCKET_SETUP(i, SOCKET_TCP,80,FLAG_PASSIVE_OPEN)`. The final parameter here indicates that the socket is passive, which means it waits for requests from clients. The sockets are created in a FOR loop. The number of sockets created determines how many clients can be connected to the server at the same time. As each socket costs memory, the total number is limited. The CS8900A IC used here also has a buffer (approximately 4 kB) for incoming Ethernet packets. That's not especially large if several users want to connect to the server at the same time, or if large items such as images are requested. Actually, this doesn't matter all that much, since TCP allows the occasional packet to remain unanswered. If necessary, the client resends unanswered packets on its own initiative.

After the sockets have been created, ELM_FLEX.C initiates the A/D converter of the microcontroller a few lines later in the code. For more information about the A/D converter, see the companion Micro Webserver article 'Measurement and Control via the Internet' in this issue.

After this, the program enters a endless FOR loop. In this loop, poll_web-



COMPONENTS LIST

Resistors (SMD):
R1 = 100Ω, shape 0603
R2,R3 = 24Ω9, shape 0603
R4 = 4kΩ99, 1 %, shape 0603
R5,R10 = 4kΩ7, shape 0603
R6-R9 = 1kΩ, shape 0603

Capacitors (SMD):
C1,C2,C4-C10 = 100nF, shape 0603, ceramic
C3 = 68pF, shape 0603, ceramic, NPO

Semiconductors (SMD):
IC1 = CS8900A-CQ (5 V), shape TQFP100
D1-D4 = chip-LED, shape 0805
Recommended colours: D1 green; D2 yellow; D3,D4 red

Miscellaneous:
T1 = Ethernet transformer type TG43 (Halo) or ST7010T (Valor), see also ref. [6]
X1 = 20MHz quartz crystal, HC49_SMD case
K1 = 34-way DIL pinheader
K2 = 8-way pinheader
K3 = RJ45 connector (screened)
S1 = mini pushbutton

For software, bare PCBs and fully assembled boards, see the 'What you need' box.

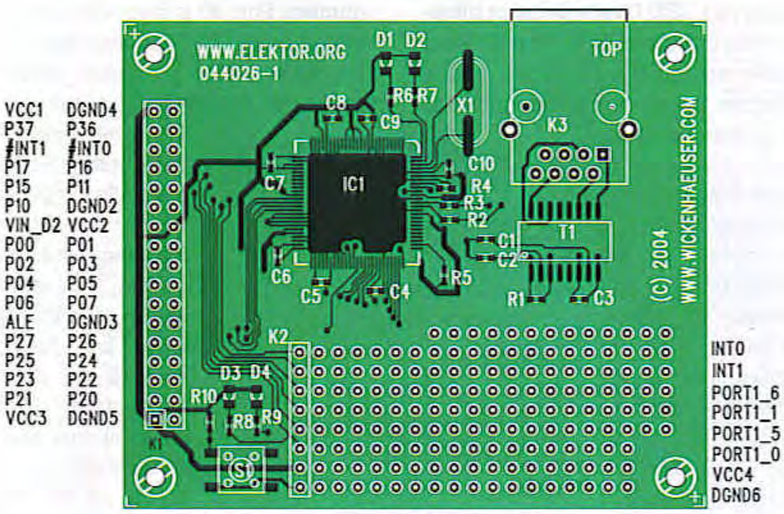


Figure 3. The network card for the MSC1210 board.

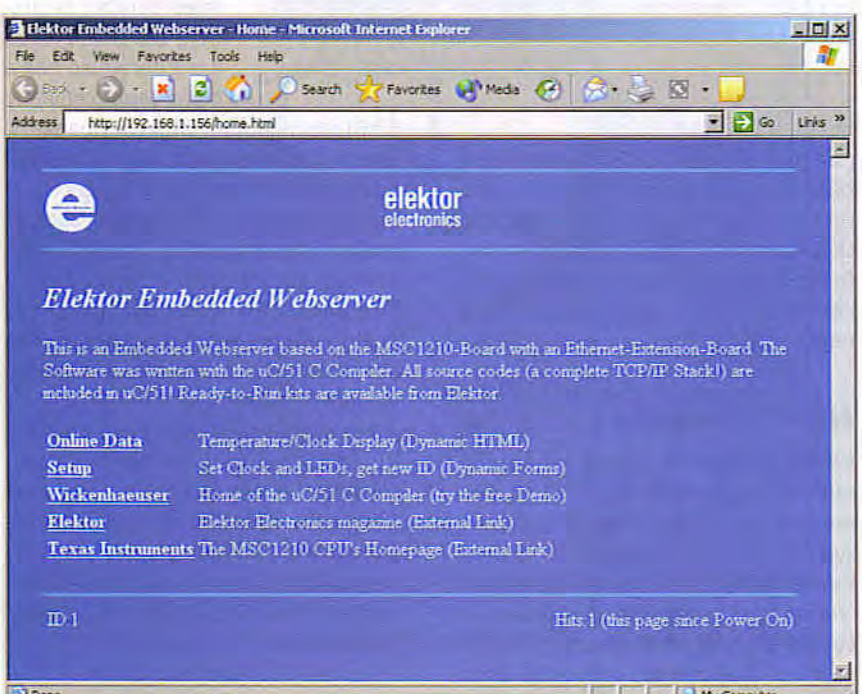


Figure 4. The web page sent by the micro webserver.

Configuring the board

The Micro Webserver only works in a TCP/IP network. Just like all other computers in a TCP/IP network, the microcontroller is assigned a unique address, which is its IP address. Before you start programming the microcontroller, you must manually specify this address, since the Micro Webserver does not work with automatic address assignment. The default IP address is set to 192.168.1.156. It belongs to a range of addresses that are specifically reserved for networks that are not directly connected to the Internet. Subscribers to ADSL or cable Internet use addresses in this range for their local networks. Addresses having the form 10.0.0.x also belong to this category. It may also be possible to request a 'real' Internet address for your Micro Webserver, but that depends on your provider. In any case, you must personally check which address range is used in your network and which addresses are available to be assigned to the server.

After choosing an address, you can turn your attention to the necessary programming software and C files. Part of the required source code (the part that implements the actual webserver) is included with the uC/51-compiler (from version 1.20 onwards). A fully functional demo version of this compiler can be downloaded free of charge from the author's home page (see reference [1]). The only difference between the demo version and the registered version is that code size for the Micro Webserver is limited to 16 kB, but that's more than enough for this application. Sample source code for initialising the webserver and implementing web pages (including several sample pages) is included in the package.

After installing the uC compiler, you must first use MakeWiz to create a workspace. In MakeWiz, open the file ...\SRC\MSC1210\ELM_FLEX\ELM_FLEX.MAK. Then change something in the text (for example, add your own version number) so that the Save button will be enabled. Tick the 'Write JFE-Workspace File' check box and save the file **(Figure 5)**.

Now you can start the JFE editor (with thanks to Jens Altmann). In JFE, use 'Open Workspace' to open the file ...\SRC\MSC1210\ELM_FLEX\ELM_FLEX.WSP. All of the files belonging to the project will appear in the editor window. Now you have to specify the previously determined IP address in the ELM_FLEX.C file. You can do so in the line `COMPOSE_IP(my_ip, 192.168.1.156).`

A workspace that has been created using MakeWiz causes three special buttons to appear in JFE: 'MAKE', 'RE-MAKE' and 'DL.BAT'. The MAKE button causes the project to be compiled, but it limits processing to the files that have actually been modified. The is the usual (and fastest) way to generate the hex file you need for programming the microcontroller. RE-MAKE must be used if something not present in the workspace has been modified, such as a header file (.h). This command causes everything to be recompiled. Finally, DL.BAT causes the result to be sent to the MSC board. This actually amounts to simply executing a batch file, to which JFE passes a parameter. This parameter is always the name of the target file, which in this case is ELM_FLEX (with no extension).

The specific command line that initiates downloading to the MSC board is stated in the batch file (which is also located in the project folder). In this case, the command line is `download /F%1.hex /X11 /P1 /T /B9600.`

Parameter P1 indicates that COM1 of the PC must be used for programming. This can be changed if necessary.

So far, so good: you've modified the IP address in ELM_FLEX.C, you've compiled the project, and your finger is just itching to press DL.BAT — but hang on a moment! Before you can download anything to the board, you have to acquire a copy of the original Texas Instruments downloader (Downloader.exe). You can obtain this from the MSC group site at Yahoo (reference [4]), among other places, and it can be placed in the project folder. If you wish, you can also place it in a more general location, but in that case you naturally have to specify its new location in DL.BAT.

Be sure to fit jumpers J1 and J2 on the MSC1210 board (J3 must remain open). If J1 and J2 are not fitted, the board is protected against resetting and modifying the firmware via the PC. Finally, you need a null modem cable to connect the board to the PC, but that should be obvious. After you've found a place for the downloader, modified DL.BAT if necessary (to specify a different COM port or change the path to the downloader), connected the board to the proper PC port, and powered up the board, you're finally ready to click on DL.BAT in JFE.

If everything goes as it should, the MSC1210 board will return a short greeting message, and if '<NET FAILURE>' is not included in this message, the Ethernet board has been successfully recognised. In addition, one of the red LEDs on the MSC board should blink slowly.

After downloading the code, don't forget to remove jumpers J1 and J2.

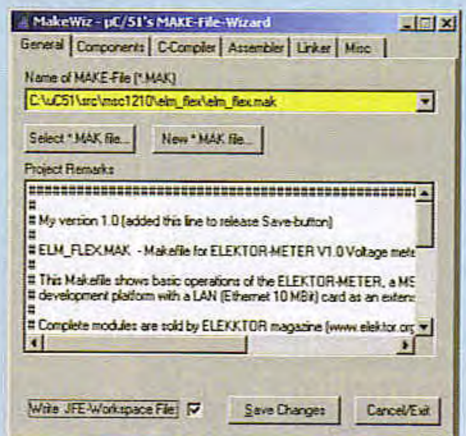


Figure 5. Use MakeWiz to store the project.

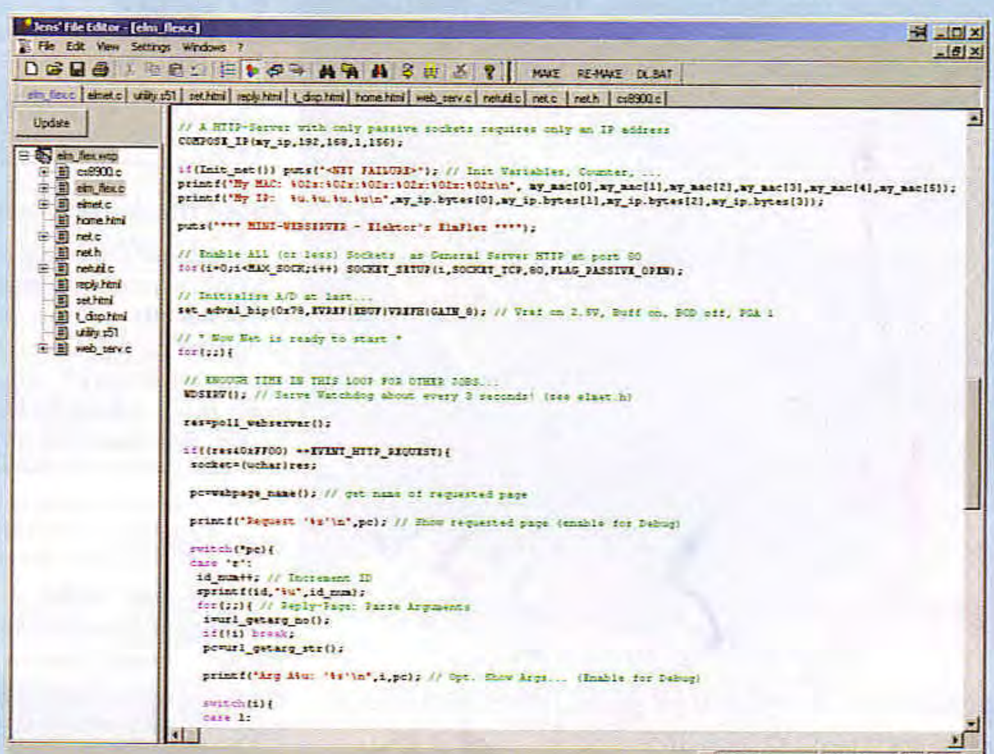


Figure 6. With JFE, all files are easily accessible.

What you need

The Micro Webserver consists of:

- the MSC1210 'Precision Measurement Central' board (see the July/August 2003 issue of Elektor Electronics)
- the 10-Mbit Ethernet network card (RJ45, twisted pair)
- the µC compiler with the necessary software (Readers Services order code 044026-11)
- the TI downloader program (Downloader.exe)

The MSC1210 microcontroller card and the associated network extension are available from Elektor Electronics. The µC compiler, including all the necessary source code, can be downloaded at no charge from www.wickenhaeuser.com or from the Elektor Electronics website.

The programming software for the MSC board (Downloader.exe) is available from reference [4]. Updates will be available from the author's website.

Prices:

- ready-made MSC1210 board: £69.00 (US$112.50) (assembled and tested; Readers Services order code 030060-91)
- ready-made network extension for the MSC1210 board: £41.95 (US$73.95) (assembled and tested; Readers Services order code 044026-91)
- combined package: assembled MSC1210 board, network extension and all related Elektor Electronics articles on diskette: only £103.50 (US$184.95) (Readers Services order code 044026-92)

For die-hard DIYers, bare PCBs are also available for the MSC1210 board (Readers Services order code. 030060-11) and the network extension (Readers Services order code 044026-11). Note that most of the components are SMD types, and some of them are very difficult to obtain as one-offs as well as solder by hand.

server() is called periodically. As long as the result returned by this call is '0', other (user-written) routines can also be executed in this loop. However, it's important to ensure that user-written extensions do not take up too much processor time, since the webserver will otherwise become inaccessible.

The FlexGate TCP/IP stack works with events. The Micro Webserver only responds to EVENT_HTTP_REQUEST (page request) and EVENT_SOCKET_IDLETIMER (which has a period of approximately 0.5 s). If a client wants to access a page, the name is first requested using webpage_name(). Following this, webpage_bind() is used to prepare the corresponding page for the reply. Pages that are to be externally available must be declared in advance as array extern code uchar (see ELM_FLEX.C).

This completes the process if the requested page does not contain any dynamic data. However, dynamic data is exactly where the power of this handy little device lies. An example of dynamic data is measurement data coming from the microcontroller board. Such data can easily be incorporated into a web page. And in the other direction, you can remotely control the microcontroller outputs via a web page. To find out more about this, see the companion article 'Measurement and Control via the Internet' in this issue.

Naturally, there's a lot more we could say about the Internet portion of the software (the TCI/IP stack), but that goes beyond what we had in mind for this article. If you want explore this question in more detail, have a look at the manual for the stack. You'll find it in the folder ...SRC\FLEXGATE\ that comes with the microcontroller compiler. In addition, Texas Instruments is presently preparing an application note for this project. The details will appear in due time on the TI website.

(044026-1)

About the introductory illustration:

This jumble of lines may appear chaotic, but it actually represents a reasonably well-organised entity: the Internet. This 'map' was automatically generated by a program that literally combs the Internet. In its travels, the program also came close to the server where the Elektor Electronics website is hosted. See www.opte.org.

Indicators:

Cyan:	*Asia Pacific*
Pink:	*Europe, Middle East, Central Asia, Afrika*
Yellow:	*North America*
Blue:	*Latin America and Caribbean*
Red:	*RFC1918 IP Addresses*
Black:	*Unknown*