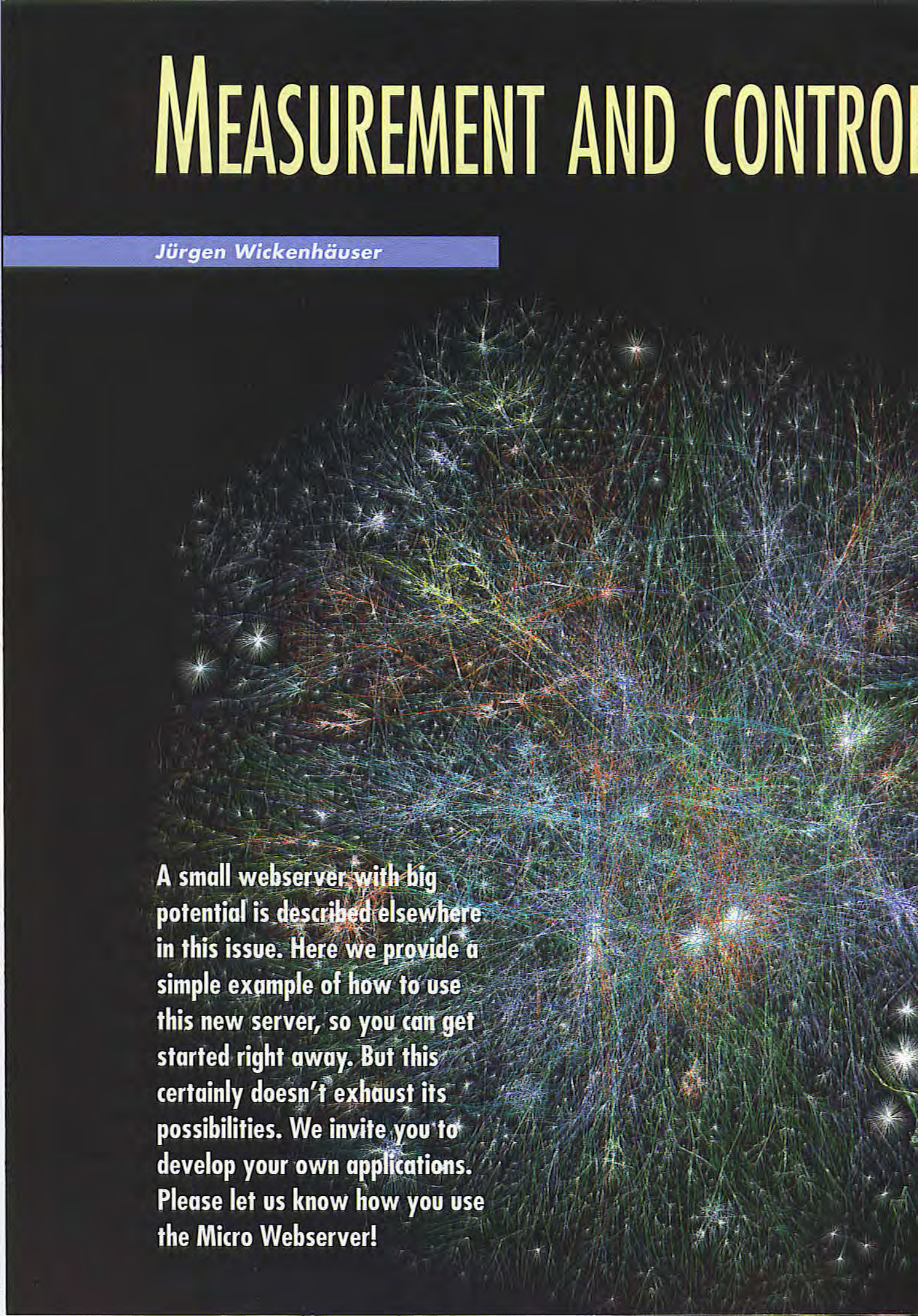
MEASUREMENT AND CONTRO

Jürgen Wickenhäuser



A small webserver with big potential is described elsewhere in this issue. Here we provide a simple example of how to use this new server, so you can get started right away. But this certainly doesn't exhaust its possibilities. We invite you to develop your own applications. Please let us know how you use the Micro Webserver!

VIA THE INTERNET

USING OUR MICRO WEBSERVER

As an example of user-developed applications for the Micro Webserver, here we show how a software clock can be used to incorporate variables from the MSC1210 board in a web page, and how variables on the board can be modified via the Internet. We also show you how to connect an analogue temperature sensor to the board, and we describe the software you need to make the measurements accessible throughout the world via the webserver.

Web pages

After configuring the microcontroller board and network card for proper operation according to the procedure described in the companion Micro Webserver article, you're no doubt keen to conjure up some data on the Internet. With this Micro Webserver, you can do this by using web pages. Websites are generated using the HTML language. HTML code can be generated and edited using any desired text editor. For example, you can use the Notepad, which is a text editor that is supplied with Windows. There are also plenty of alternatives available on the Internet. Special code editors are also available. Such editors can automatically display the HTML code in a different colour, among other things. This is a handy feature, since 'regular' text is clearly distinguished from code, which makes finding errors go a lot faster.

The following is a simple example of a bit of HTML code:

```
<html>
 <head>
 <title>A simple
page</title>
 </head>
<body>
Hello <a
 href=http://www.elek-
tor.org>Elektor</a>!
</body>
</html>
```

The result produced by the code for this page is shown in **Figure 1**.

We have no intention of describing all the details here, since innumerable manuals and tutorials can be found on the Internet, and they go into much more detail than we possibly could here. Just try using Google (www.google.com) to search for "HTML manual" or "HTML tutorial".

HTML and the microcontroller

To make data from the microcontroller board visible via web pages, variables must be incorporated into the HTML code. The server then fills these variables with actual data when the code for the web page is executed. With the FlexGate TCP/IP stack, this is very easy. The '@' symbol is used to mark a variable. If you want to use the actual '@' symbol in an HTML page, write '@@' in the code.

Wherever there is a variable in the code, the stack automatically enters the corresponding C variable. The C variable must always have the type string, which means it must be an array of type char. The sample file SET.HTML (**Figure 2**), which is included with the uC/51 compiler, shows how this appears in HTML. This file is required for configuring the server, and it can be downloaded free of charge from the Elektor Electronics website.

In lines 19-21 of this file, you will see the variables hr, min and sec. The current time is set here. Lines 28 and 29 contain the code for check boxes that depend on the values of ls3 and ls4. These two variables contain either 'd' or an empty string. This yields either 'checked' or 'checke', respectively. A feature of most browsers is that they ignore anything they don't recognise. Although this technique is not especially elegant, it means that 'checke' will not be interpreted. As a result, the checkbox will not be ticked if the variable does not contain a 'd'.

Controlling the microcontroller

Of course, users must also be able to modify data in the microcontroller via the Web. In HTML, this is done using structures called 'forms'. Here we use the GET method. This causes the contents of a form to be added to the called URL when it is sent. For instance, if you want to set the clock to 16:29:35 and you click on 'Set Clock', the REPLY.HTML page is called with the following parameters:

```
http://.../reply.html?A1=16&A2=29&A3=35?A
9=Set+Clock
```

In this line, '?' marks the start of a parameter string and '&' separates the individual parameters. The '+' symbol indicates a space. Special characters are converted to their equivalent hexadecimal values, which are preceded by '%'.

Incidentally, an HTML page can contain several forms, whose parameters are then combined in the URL.

The FlexGate TCP/IP stack assumes that all variables start with 'a' or 'A' (not case-sensitive), followed by a number in the range 1–255.

Associated C code

Now you know exactly how to incorporate variables in the HTML code, but what about the code for the MSC1210?

As described in the companion Micro Webserver article elsewhere in this issues, the ELM_FLEX.C file is where users configure the server. In the program loop in which the actual server runs, whenever a page is requested a check is made to see whether specific parameters must be passed with the URL. For example, if an HTML document such as REPLY.HTML is

Suggested applications

The Micro Webserver makes an excellent platform for all sorts of applications where it's handy to be able to observe or control something via the Internet. The following is a list of components and Elektor Electronics circuits that could be used to provide interfaces to the outside world.

A complete weather station can be built using:

- a temperature sensor (using a Pt100 sensor, or digitally with an LM76 or the like)
- a lightning detector (June 2003)
- an anemometer (May 2004)
- a hygrometer (such as the HS1100 used in the January 2004 'Climate Logger' project)
- a rain-barrel gauge ('Rainwater Storage Gauge', December 2000, or 'Precision Level Gauge', December 2001)
- a light intensity sensor (LDR instead of Pt100)

Remote control or monitoring of household appliances and fixtures:

- temperature monitoring (Pt100, LM75A or the like)
- on/off control for a coffee machine, central heating or lighting (with a relay)
- sun awning and roller blind control (with a relay)
- outside lighting (with a relay, possibly with an LDR)
- intruder detection (IR detector from a DIY home improvements shop, or the vibration detector from the December 2002 issue)

Access control with central registration and monitoring, in combination with:

- smart card readers (available from Conrad and other sources)
- light barriers (such as 'Simple Infrared Light Barrier', July/August 2002)
- door openers (electromechanical, from DIY home improvements shops)

Monitoring and controlling machinery

- rpm (see 'Rev Counter for R/C Models' in the November 2003 issue for an idea)
- voltage and current (using a voltage divider or sense resistor and optocoupler via the A/C converter input)
- temperature (Pt100 or LM76 sensor)
- liquid level ('Rainwater Storage Gauge', December 2000, or 'Precision Level Gauge', December 2001)
- flow or discharge (flow sensor, available from Conrad and other sources)
- pressure (pressure sensor, available from Conrad and other sources)
- valve controller (with a relay)
- relay or PWM controller (PWM signal via a solid-state relay)

Centralised data access and data processing, in combination with an LC display

('LC Display with I²C Bus', September 2003) and bar-code reader (from Conrad Electronics or another source)

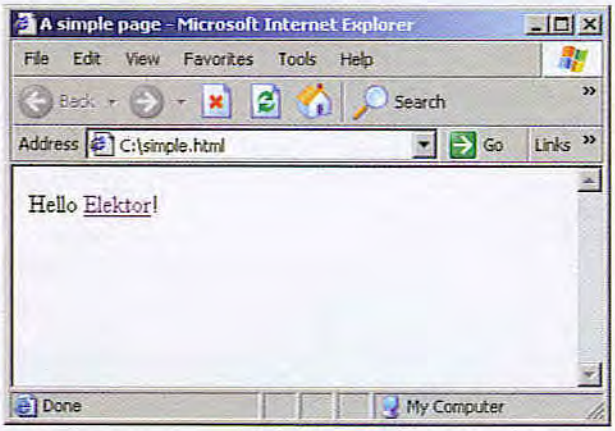


Figure 1. As shown here, a real web page can be created using a few simple lines of HTML code.

requested, url_getarg_no() is used to determine how many parameters are to be passed with the URL. The content is determined using url_getarg_pc(). As already mentioned, the only permissible values for the arguments are 'A1'–'A255' and 'a1'–'a255'.

When copying the arguments to local variables, remember that you cannot copy more data than the amount declared for the variable. One of the primary examples is strings in C: here the final '0' byte also counts!

Pt100 temperature measurement

One of the demo pages for the webserver (T_DISP.HTML) makes temperature measurements. Pt100 sensors are espe-

FlexGate TCP/IP stack v2.0

The FlexGate TCP/IP stack used here has been specially developed for 8051-family processors. In contrast to the more elaborate stacks for PCs, its hardware requirements are quite modest. A complete webserver can be set up using less than 1 kB of RAM and approximately 12 kB of code. The stack is open-source software, which means the source code is freely available. In its basic configuration, this stack can handle the most important Internet protocols, which are ICMP, ARP, PING, TCP and UDP. All that has to be added for a webserver is ARP and TCP. With the FlexGate TCP/IP stack, in principle any desired number of concurrent connections is possible.

The stack is integrated into the uC/51 compiler. This is a complete development environment for ANSI C (see reference [1]). The demo version is normally restricted to 8 kB of code, but for the Micro Webserver the limit is automatically increased to 16 KB.

cially suitable for this purpose. As the MSC1210 provides a precise reference voltage and its A/D converter can be calibrated using the same voltage, any inaccuracies that may be present do not affect the measurement results.
If you use a precision resistor and Pt100 sensor (DIN class 1/3 is the most suitable), you don't even have to calibrate the circuit. If the components are not so precise, you can achieve a perfectly adequate calibration using an ice-water bath (0° C) and a warm-water bath at approximately 40° C with a fever thermometer.

Unfortunately, Pt100 sensors are not fully linear, but with the indicated component values the error is within 0.5° C over the range of –10° C to +50° C (see **Figure 3**).
As a constant current of approximately 2 mA flows through the Pt100 sensor, a certain amount of self-heating can occur with types having a small package. This is often stated in the data sheet where relevant. The voltage divider incorporating the Pt100 sensor is connected to the U_{ref}, AIN7 and AGND/AINCOM terminals of the MSC1210 board (see **Figure 4**).

Processing the measurement data

The ELMET.C file contains not only initialisation subroutines, but also interrupt subroutines. The software clock and the A/D converter both work with interrupts. This is used to continuously maintain the value measured by the A/D converter in a temporary variable of type long. To save computation time, this value is only converted into a temperature after it reaches ELM_FLEX.C. The macro AD_FILT8 can be used to configure the A/D converter to always return the average value of several measurements.

In this simple demo program, the A/D converter is calibrated once only using set adval bip(). However, for reliable measurements it's a good idea to regularly recalibrate the A/D converter, such as every few minutes. One way to do this is to call the initialisation routine in the main loop if a flag is set by the interrupt routine for the software clock. This method is better than calling the function directly in the interrupt routine. This is because as a matter of principle the interrupt routine should be exited as quickly as possible, in order to avoid delays in responding to any other interrupts.

Conclusion

The software clock and making temperature measurements using a Pt100 sensor with the A/D converter are simply two sample applications. Naturally, the server can be used to implement just about any task you can imagine. Some of the possible applications we can think of are listed in the 'Suggested applications' box. This list is far from complete, and it is actually intended to stimulate you to use the MSC1210 board and the new network interface to develop your own applications. We're very much interested in seeing your ideas. We'll certainly report all the attractive, interesting and clever applications in *Elektor Electronics*. And as always, we always reward applications suitable for publication with a suitable payment. Let us hear from you at editor@elektor-electronics.co.uk, subject: Micro Webserver.

(040147-1)

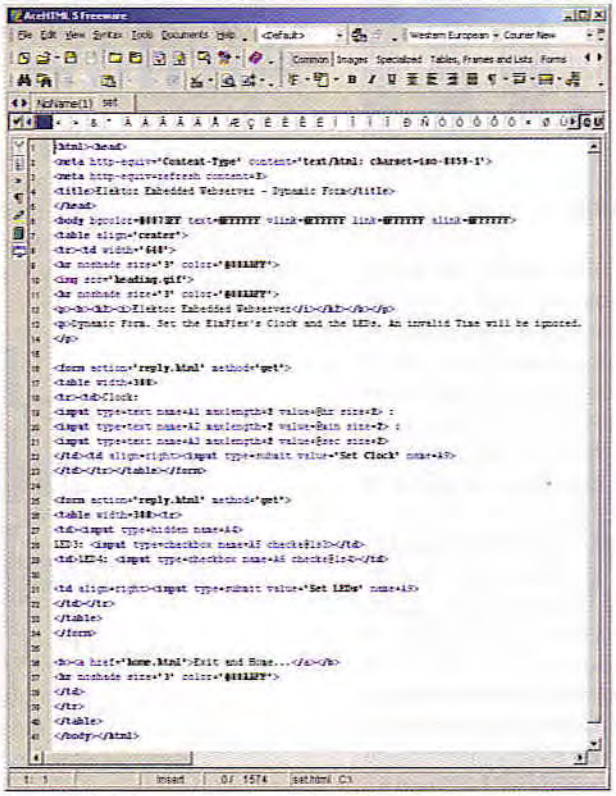


Figure 2. SET.HTML displayed in the AceHTML editor window.

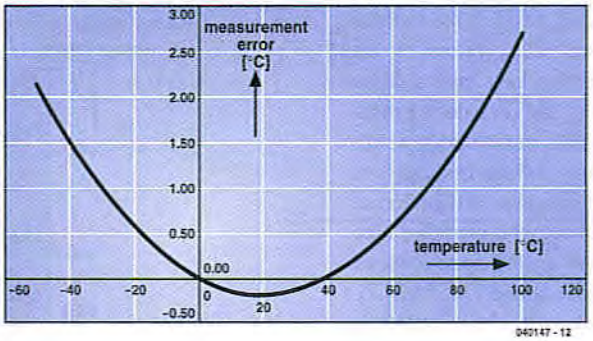


Figure 3. Non-linearity of the Pt100 temperature sensor.

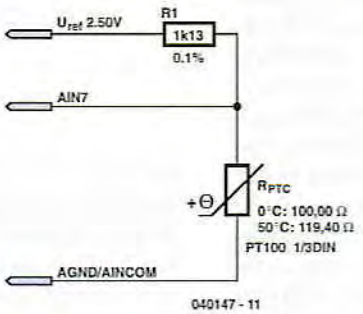


Figure 4. Connecting the temperature sensor to the reference voltage and A/D converter.

Internet references

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

[2] www.ti.com/msc The MSC121x home page

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

[4] freeware.acehtml.com
Free HTML editor. Registration not necessary.