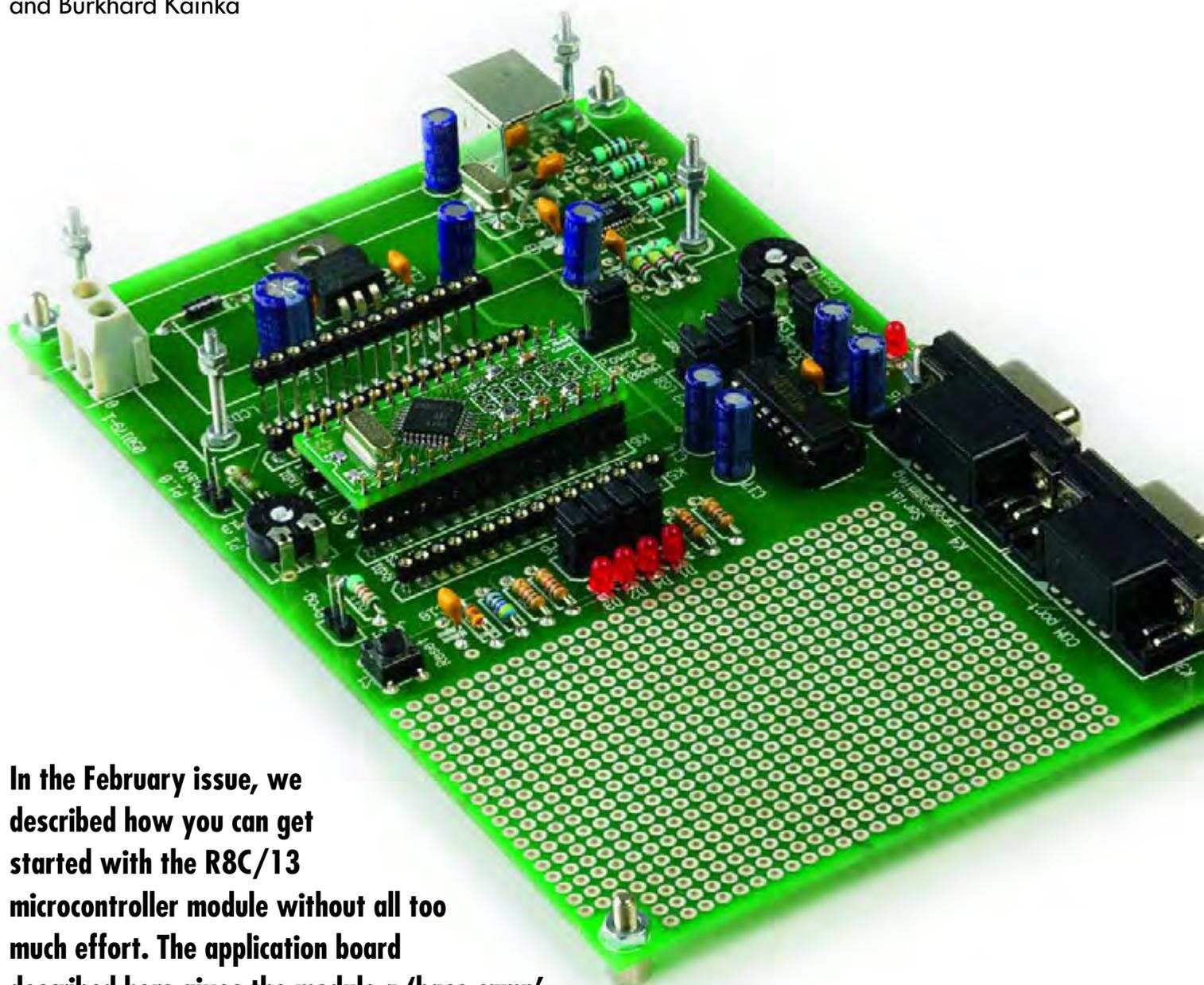


# Application board

## Base camp for high endeavours

Gunther Ewald  
and Burkhard Kainka



**In the February issue, we described how you can get started with the R8C/13 microcontroller module without all too much effort. The application board described here gives the module a 'base camp' with lots of connections to the outside world, including a USB interface and two serial ports. There's also an LCD interface and other useful features for developing your own applications.**

# for R8C/13

The starting point for this design was the question, what do you need to successfully utilise all the major features of the microcontroller with a minimum of effort? The result is a circuit board with the following complement of functions:

- two serial interface ports
- a USB port using an integrated

USB/serial adapter

- an LCD interface
- four LEDs that can be connected to port lines
- a potentiometer connected to one of the many analogue inputs
- a power supply socket and 5-V voltage regulator
- an option for powering the board via

the USB port

- a reset switch
- a jumper for the MODE input

The USB interface on the board deserves special mention. Serial interfaces are becoming increasingly rare on new PCs, but asynchronous serial interfaces are fundamentally necessary for communi-

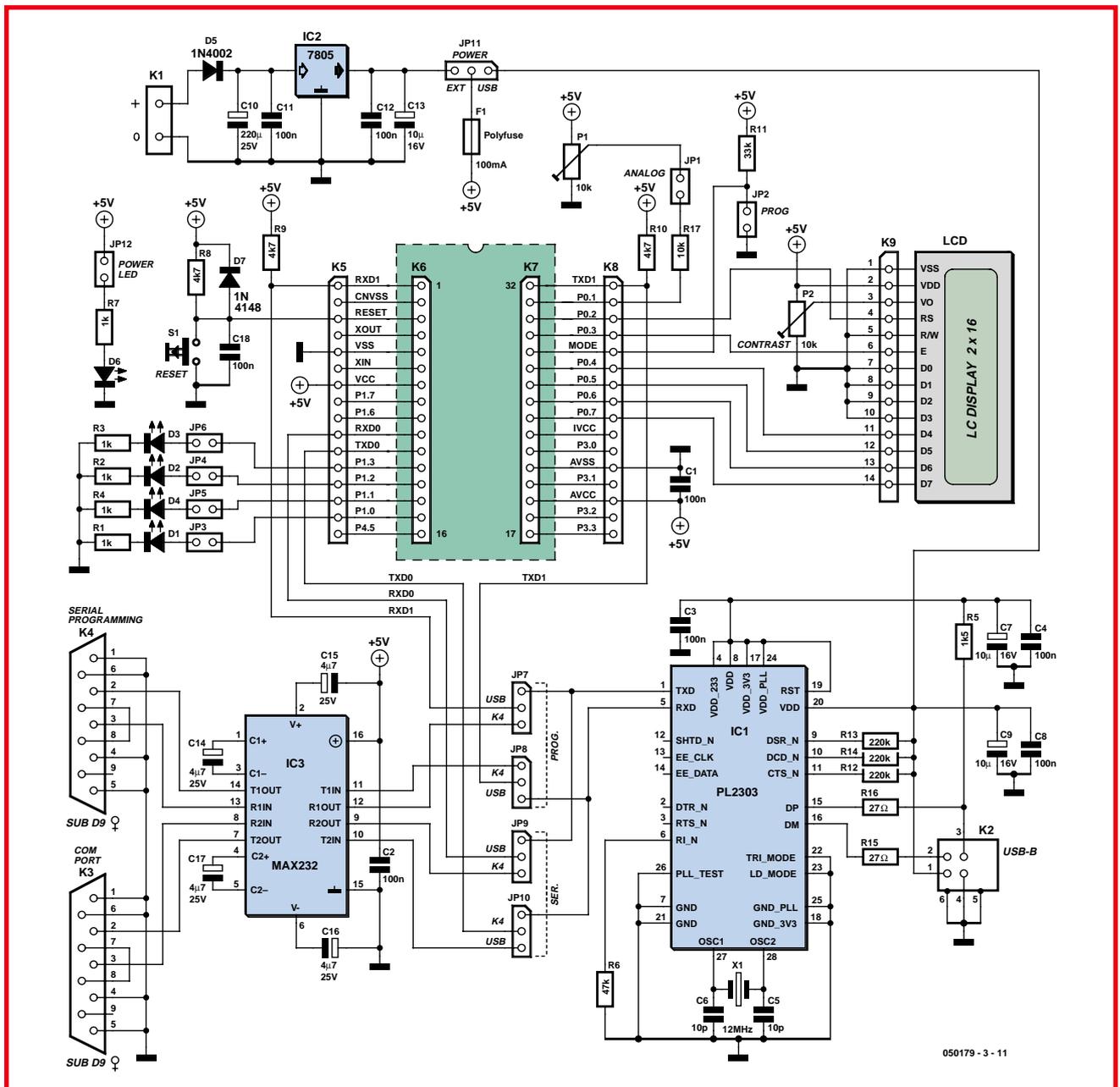
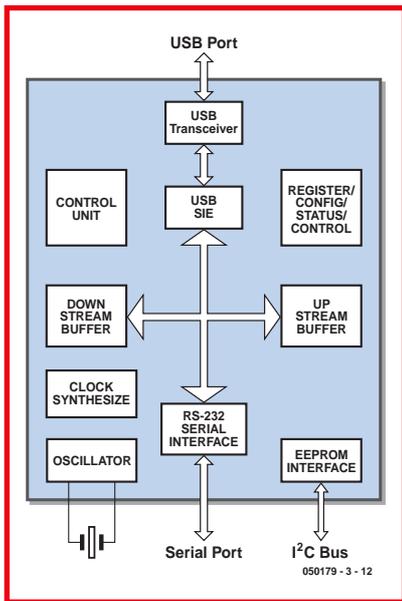


Figure 1. The circuitry on the application board provides two serial ports, a USB interface and an LC display to connect the R8C daughter board to the outside world.

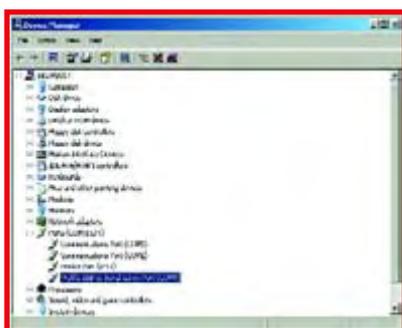


**Figure 2.** Block diagram of the Prolific interface converter.

caution with the microcontroller. For that reason, we decided to simply include a USB/serial adapter on the board. In contrast to the equally viable approach of using an external USB adapter, this has the advantage of allowing the board to be powered via the USB port. A total of four jumpers determine which serial interfaces of the microcontroller are routed via the RS232 ports or the USB port. For instance, you can load and debug programs via the USB interface, or you can use the USB interface for serial communication with the microcontroller and your programs.

## Microcontroller ports

The schematic diagram (Figure 1) essentially shows the principal fea-



**Figure 3.** The virtual COM interface in the Windows XP Device Manager window.

tures of an R8C system as already mentioned in the February issue. The MODE jumper and Reset switch are important features. Interface drivers are also necessary for the RXD1 and TXD1 leads of the debug interface, which are routed here via JP7 and JP8 to allow either RS232 or USB to be used. Depending on the jumper settings, you can use either the MAX232 and connector K4 or the PL2303 USB controller and a USB link.

In addition to the serial debug interface (UART1), the R8C/13 has a second, fully independent serial interface (UART0) that communicates via the TXD0 and RXD0 leads. As is well known, the MAX232 can handle more than just two lines – in fact, it can handle four. That's why jumpers JP9 and JP10 and a second serial interface connector (K3) are on the board. You can also select either RS232 or USB for the UART0 interface.

Now for a few words about the microcontroller ports. The microcontroller leads are routed to single-row socket headers in order to leave all options open. However, specific functions have also been assigned to most of the port leads. Here 8-bit port P0 is intended to be used for connecting the LCD module, but it can of course be used for some other purposes if you don't need an LCD. A standard alphanumeric LCD can be driven in 4-bit mode using only six port pins. Leads P0\_2 to P0\_7 are thus used by the LCD.

P0\_0 is reserved for the debug interface (TXD1) and is only available for user applications if the debug function is not used. That still leaves P0\_1. As this lead can also be used as an analogue input, a potentiometer providing a variable input voltage of 0–5 V can be connected to it via JP13. A supplementary protective resistor (R17) prevents overloading if P0\_1 is accidentally configured as an output while connected to the potentiometer. The two jumper pins can also be used as a measurement input. If you want to connect a sensor that outputs both positive and negative voltages, you can use the potentiometer to adjust the zero point to where you need it. The four lower bits of Port 1 can be connected to the four LEDs via jumpers and series resistors. That's ideal for your initial programming exercises, such as those described in the February issue. If you haven't already tried the 'blinker' example program, now's your chance to get up to speed.

If you remove jumpers JP3–JP6, these leads can also be used as four additional analogue inputs. Just add a few lines of code, and you have a four-channel digital multimeter with LCD display!

A total of seven individual port leads have been left open, but you're bound to think of a use for them. Applications such as an I\_C or SPI interface and the like can always use an extra lead or two.

## It takes a bit of juice

There are two options for powering the board, which are selected using JP11. Either you can use the power supply connector (K1) and voltage regulator IC21, or you can use USB with its built-in 5 V supply voltage.

If you're using USB anyhow, it makes sense to dispense with an external power supply. However, a bit of caution is advisable here. According to the USB specification, every USB down port is supposed to be protected against overloads by a PolySwitch protective device. However, painful experience proves that it just ain't so. A simple short on the USB supply line results in a small puff of smoke from the PC motherboard and causes the corresponding down port to be permanently disconnected from the supply voltage. For cost reasons, manufacturers obviously do nothing more than placing small series resistors in the supply tracks, which then simply melt through.

For this reason, a PolySwitch protective device (F1) rated at 100 mA is included on the application board for the specific purpose of protecting the PC. That creates peace of mind and allows you to experiment in a relaxed mood.

## USB interface

Next we come to the USB interface. The PL2303X, which sports the designation 'USB to Serial RS232 Bridge Controller', comes from the Taiwanese manufacturer Prolific (sales rep: Glyn). Attentive readers may recall having already seen this Prolific IC in a built-in, encapsulated form in an interface converter cable described in the September 2005 issue of *Elektor Electronics*.

This IC is a full-speed USB device for USB 1.1 and thus compatible with the more recent USB 2.0 interface. That means the data transmission rate on the bus is 12 Mbit/s. R5 connects the

D+ lead of the USB connector to 3.3 V to signal a full-speed device to the PC. That causes the PC to load a driver for a virtual serial interface. Such a driver allows serial interfaces to be operated at transmission rates up to 1,228,800 baud (that's 1.2 Mbaud!). In line with the capabilities of the R8C/13, various standard baud rates are used here. The virtual serial interface behaves exactly the same as a real RS232 interface for downloading and debugging programs.

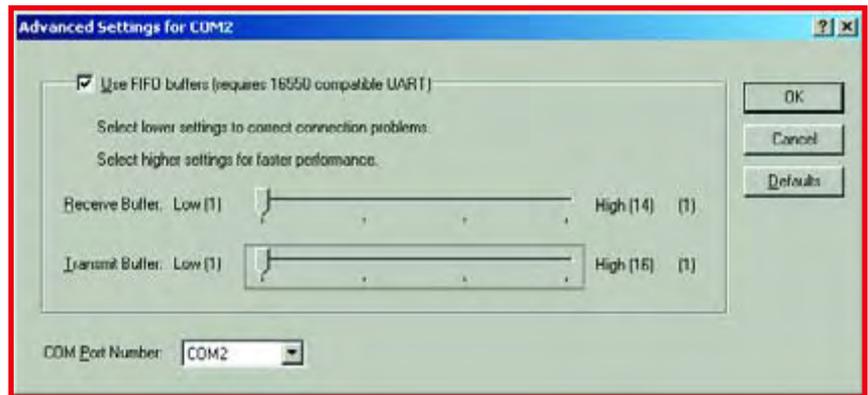
**Figure 2** shows the internal structure of the IC, which consists of a USB transceiver, a serial interface, and two data buffers. The data buffers support fast, uninterrupted data transfers, with data blocks being packed into individual USB frames. It is also possible to connect an I<sup>2</sup>C EEPROM to store user configuration settings, but that option isn't used here.

The non-inverted RXD and TXD signals are available at the serial side of the converter and can be connected directly to the R8C. The inputs of the PL2302X are compatible with 3.3-V and 5-V logic levels. The TXD, DTR and RTS outputs provide either 3.3-V or 5-V logic levels, depending on the voltage connected to pin 4. In this case, this voltage is 3.3 V from the internal voltage regulator of the PL2303. That means the level applied to the serial input of the R8C/13 is 3.3 V, although the level actually expected there is 5 V. That's permissible, though, because the R8C/13 can also be operated at 3.3 V, and besides that a 3.3-V signal at the RXD input of the microcontroller is adequate even when it's operated at 5 V. All inputs of the R8C regard any voltage above 0.2 V<sub>CC</sub> as a logic 'high' level.

## Driver installation

You may have already used a Prolific USB adapter with your system. In that case, the interface will be recognised as soon as the board is connected to the PC. If it isn't, or if you experience problems using the USB port, you should install the latest version of the driver. You can obtain the driver by downloading the file **050179-3-11.zip** from the Elektor Electronics website ([www.elektor-electronics.co.uk](http://www.elektor-electronics.co.uk)). It contains the file 'PL-2303 Driver Installer.exe'.

Run the driver installer file before connecting the board. Any older version of the driver present in the PC will be deleted automatically when the new driver is installed. The next time the board is connected (or another inter-



**Figure 4.** Configuring the buffer sizes and COM port number.

face converter using the Prolific IC is connected), Windows will automatically locate the appropriate driver and load it. After that, your PC will have an additional serial interface that can be used just like a normal RS232 interface. If you have already installed several such interface converters, Windows will assign a high COM number to the new virtual interface. In that case, it's recommended to use Advanced Settings to rename the interface, for example to 'COM2'. You can usually ignore any message saying that this interface is already in use, because it refers to another virtual interface that is not used at the same time. All you need to know is which serial interfaces are actually present in your PC in hard-

ware form. For example, the PC used for our tests had only COM1 present on the motherboard, but it also had an interface card with COM3 and COM4. In that case it's a good idea to name the virtual interface 'COM2' (see **Figure 3**), especially since many older programs only support COM1 and COM2. Problems with high COM numbers have also been observed with the FDT download program, so you should at least keep the number in the single-digit range.

In some cases it may be necessary to configure the FIFO buffer of the virtual interface to the smallest possible size (**Figure 4**) in order to obtain reliable communication. That's because with a large buffer size, it's theoretically pos-

## COMPONENTS LIST

### Resistors:

R1-R4,R7 = 1k $\Omega$   
 R5 = 1k $\Omega$ S  
 R6 = 47k $\Omega$   
 R8,R9,R10 = 4k $\Omega$ 7  
 R11 = 33k $\Omega$   
 R12,R13,R14 = 220k $\Omega$   
 R15,R16 = 27 $\Omega$   
 R17 = 10k $\Omega$   
 P1,P2 = 10k $\Omega$

### Capacitors:

C1-C4,C8,C11,C12,C18 = 100nF  
 C5,C6 = 10pF  
 C7,C9,C13 = 10 $\mu$ F 16V radial  
 C10 = 220 $\mu$ F 25V  
 C14-C17 = 4 $\mu$ F7 25V radial

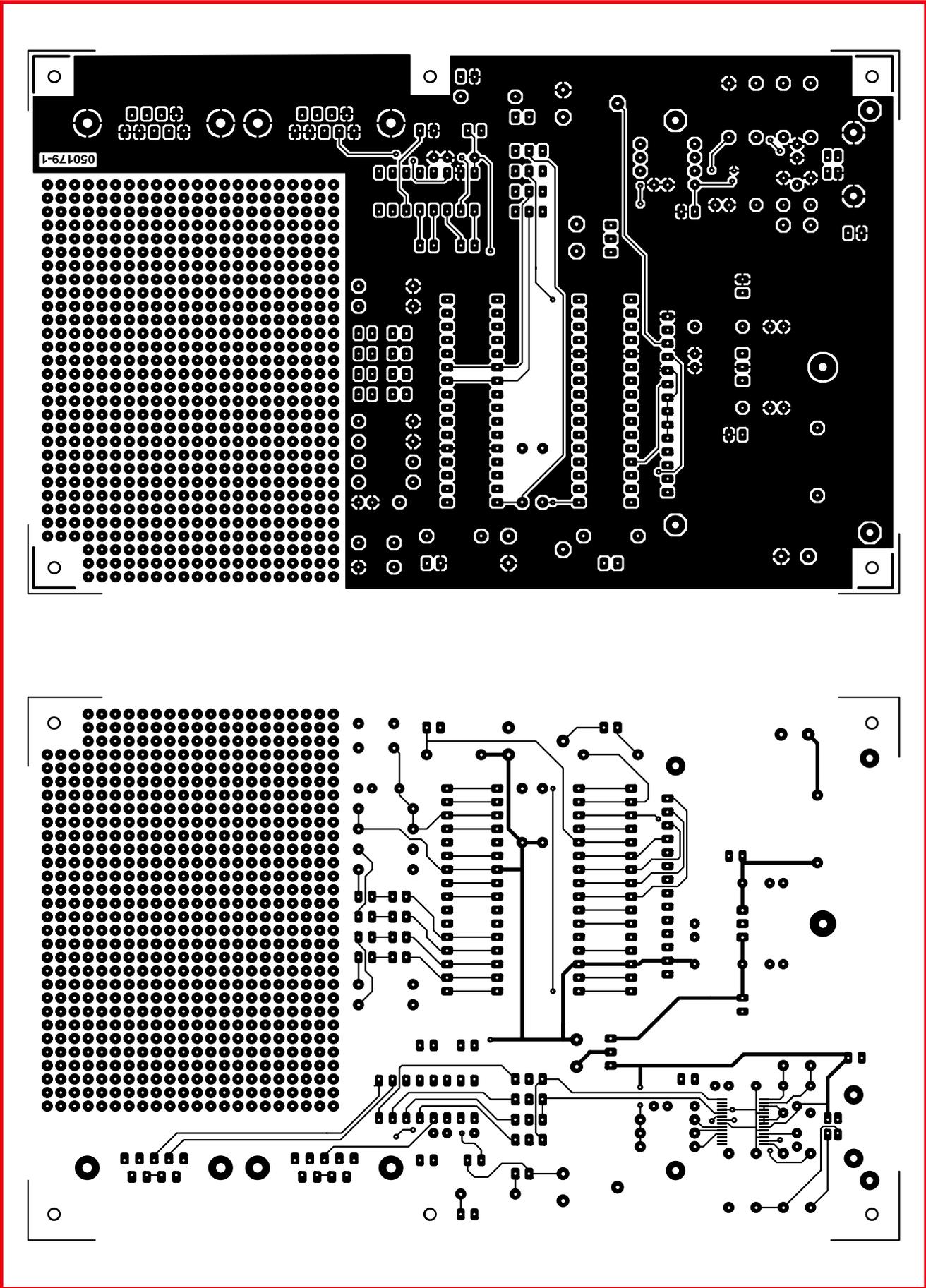
### Semiconductors:

D1-D4,D6 = LED  
 D5 = 1N4002

D7 = 1N4148  
 IC1 = PL2303X (Prolific)  
 IC2 = 7805  
 IC3 = MAX232

### Miscellaneous:

JP3-JP6,JP12 = jumper  
 JP7-JP11 = 3-way SIL pinheader  
 K1 = 2-way PCB terminal block, lead pitch 5mm  
 K2 = USB socket, type B for PCB mounting  
 K3,K4 = 9-way sub-D socket, angled pins, PCB mount  
 K5-K8 = 16-way SIL pinheader  
 K9 = LCD module, 2x16 characters  
 S1 = pushbutton with 1 make contact (e.g. TS695)  
 F1 = 100 mA Polyfuse  
 X2 = 12MHz quartz crystal  
 PCB, bare, order code 050179-1  
 PCB, ready-assembled and tested, order code **050179-92**  
 LCD with backlight, order code 030451-72  
 Poly-LED display, order code 030451-73



**Figure 5a.** The track and ...

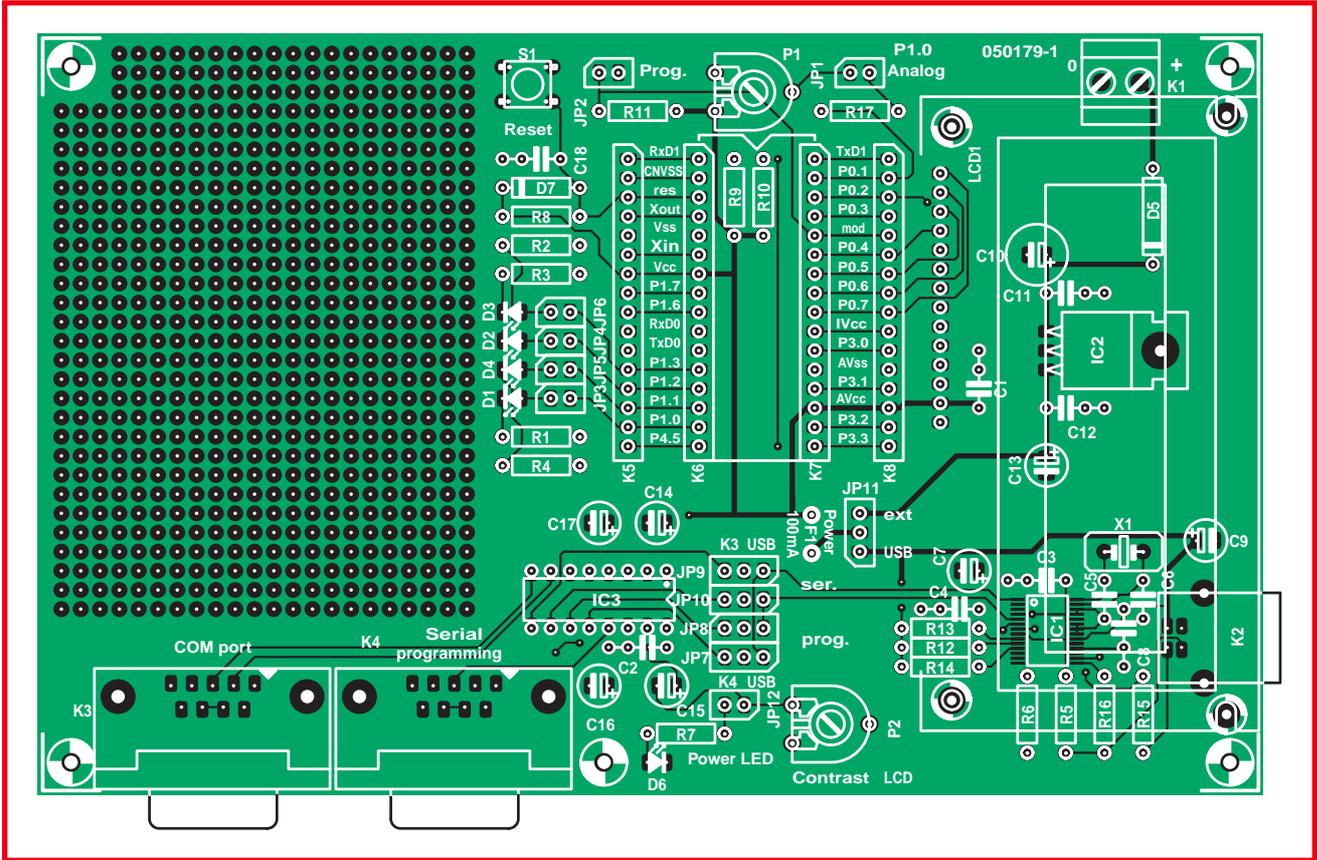


Figure 5b. ... component layouts of the double-sided circuit board.

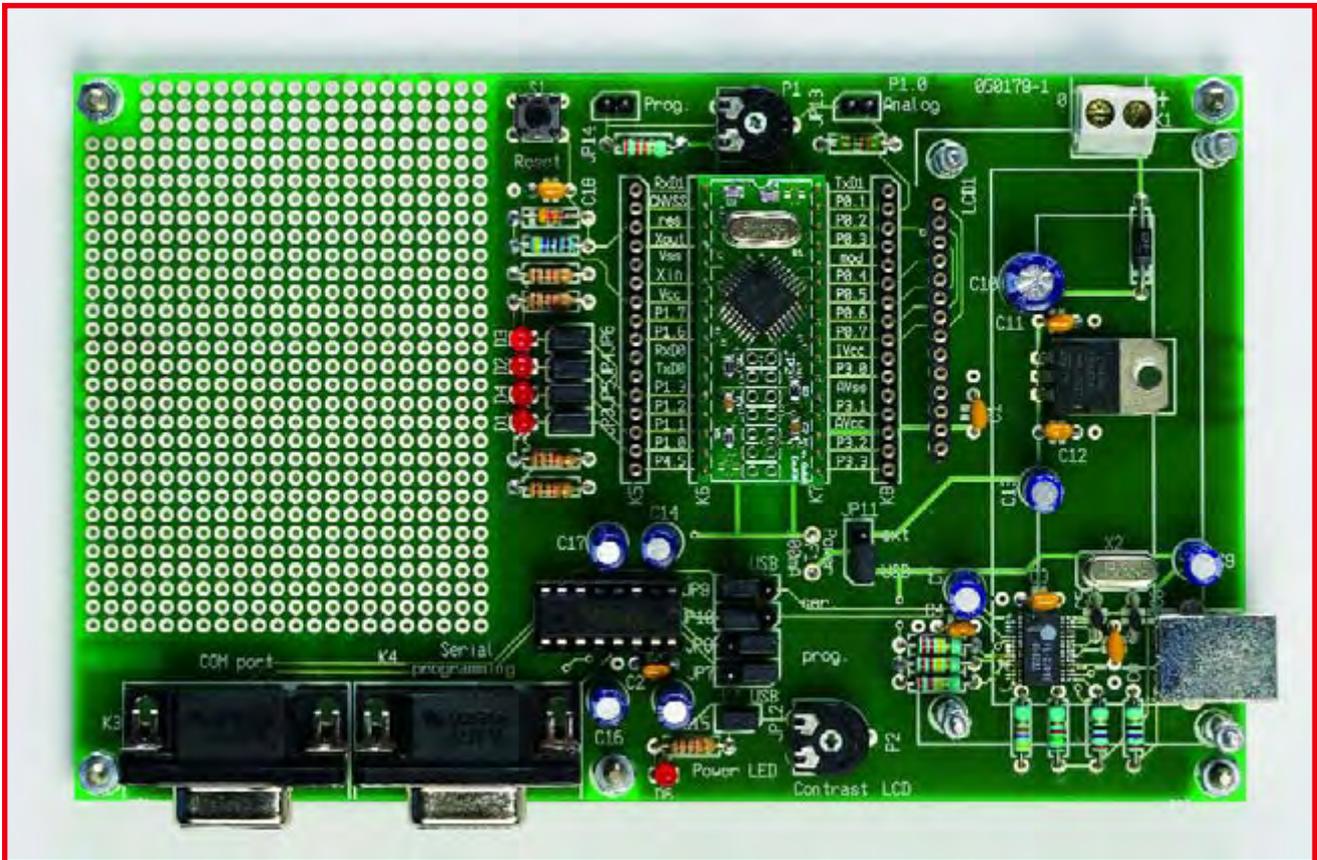


Figure 6. The fully assembled prototype board. The component overlay is slightly different from the production version shown in Figure 5.

sible for the PC software to be waiting for reply while the query is still stuck in the buffer. If errors occur when downloading programs into the flash memory of the microcontroller, reducing the size of the buffer may improve the situation.

## Board assembly

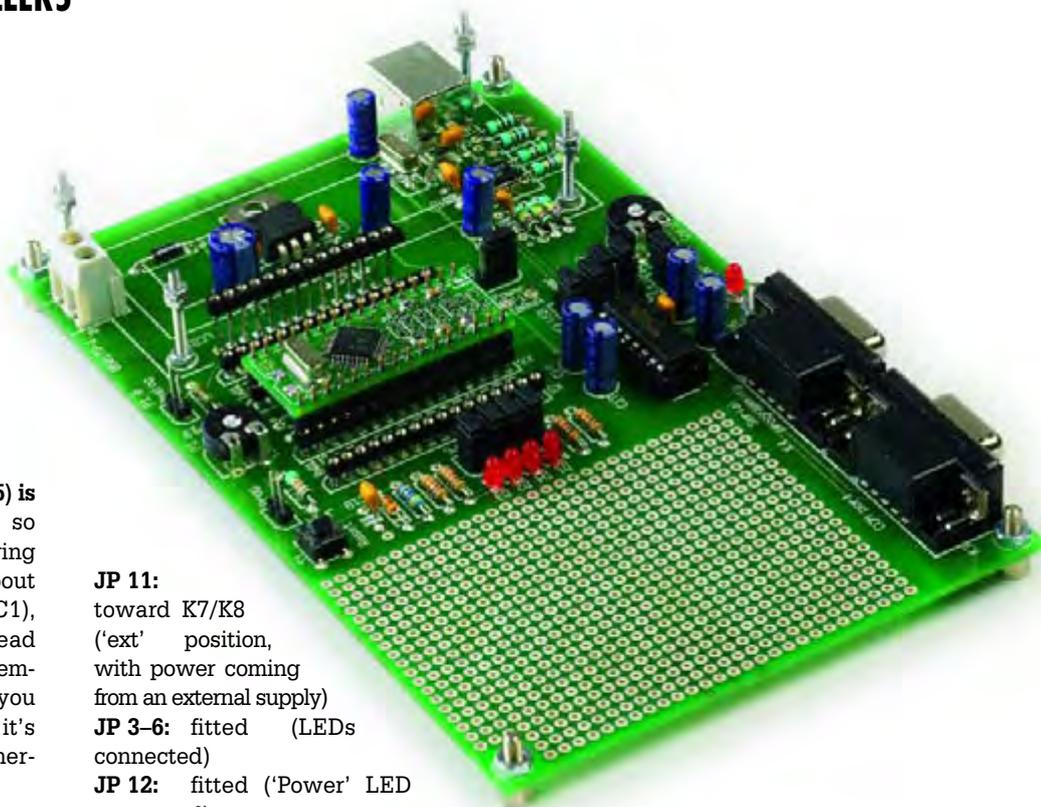
The printed circuit board (Figure 5) is also available fully assembled so readers with relatively little soldering experience don't have to worry about problems with the Prolific IC (IC1), which is only available in a 28-lead SMD package (SSOP-28). If you assemble the circuit board yourself, you should start with this IC because it's easier to solder if the board is otherwise empty.

Numerous *Elektor Electronics* SMD projects have shown that they can be soldered without using special SMD tools. Start by soldering two diagonally opposite leads in place, after first carefully positioning the IC. A check with a loupe will show whether you have a steady hand. If necessary, you can still adjust the position relatively easily at this stage. Continue by soldering an entire row of leads with a sufficient amount of solder. Any excess solder can be easily removed afterwards with a length of solder wick. If your final inspection with a loupe does not reveal any solder bridges, everything is OK. The author has also found that strong reading glasses (3 dioptr) are an excellent alternative to a loupe. They're almost as good as stereo microscope, at only a fraction of the price.

The remaining components shouldn't present any special difficulties. You can omit the pin headers (K5 and K8) if desired. If you wish, you can later connect flat cable here or wire only the connections you actually need to the prototyping area. A socket header has proven to work well as an LCD connector. The matching LCD module can then be fitted with long pin headers and simply plugged into the board.

## Initial test

Before using the board for the first time, you should re-inspect the component assembly and solder joints and fit the necessary jumpers. The following jumper settings are important:



- JP 11:** toward K7/K8 ('ext' position, with power coming from an external supply)
- JP 3-6:** fitted (LEDs connected)
- JP 12:** fitted ('Power' LED connected)
- JP 2:** fitted (debug mode)
- JP 7-10:** toward IC3 (both interfaces via RS232)

You should check the supply voltage on the board before plugging in the R8C/13 daughterboard, in order to avoid any risk to the microcontroller. LED D6 should light up after you connect an AC adapter with 9-V DC output. The voltage measured on K6 between pin 5  $V_{SS}$  (Pin 5) and pin 7 ( $V_{CC}$ ) must have a value of +5 V. After verifying this, disconnect the AC adapter and plug in the daughter board while no power is applied to the board. Now comes the most exciting moment! Reconnect the 9-V AC adapter. The 'Power' LED will light up. Connect K4 to the COM1 serial port of the PC. You can immediately download a program now, because JP2 is configured for debug mode. Use the FDT program as described in the February 2006 issue of *Elektor Electronics*. A good choice for your first download is the 'port\_toggle' program. The programming tool will indicate whether the download process completed successfully.

Next, remove jumper JP2 and briefly press the Reset button. The four LEDs connected to Port P1 should start blinking.

If you want to use the USB interface, fit JP7 and JP8 in the proper positions (toward the USB connector). That will connect RXD1 and TXD1 to the USB converter IC. After that, you must configure FDT to use the virtual COM interface. Everything else works exactly the same as with a real RS232 interface.

## KD30 debugger

Up to now, we've only described how to download finished programs into the microcontroller and run them. When you're developing a program, especially one that's relatively complex, it's helpful to use a debugger. That allows you to interrupt the program at any desired location, view memory contents, execute single steps, and much more.

The microcontroller must always be in debug mode when you use the debugger, which means that jumper JP2 must remain fitted. The Reset button isn't necessary in this case, because you can only use software commands from the debugger to start and stop your program.

We've prepared an extensive description of how to get started with the debugger and placed it on the Elektor Electronics website at [www.elektor-electronics.co.uk](http://www.elektor-electronics.co.uk). Depending on the amount of feedback we get, a special R8C activity page may also be created, so check out the website.

(050179-3)

## References:

[www.glyn.com](http://www.glyn.com)  
[www.glyn.de](http://www.glyn.de)  
 E-mail: [elektor-r8c@glyn.de](mailto:elektor-r8c@glyn.de)

## R8C forum:

For hints and tips about using the R8C/13 board, check out the 'R8C 16-bit Micro Starter Kit' topic in our Forum menu at [www.elektor-electronics.co.uk](http://www.elektor-electronics.co.uk). Naturally, we would be pleased if you post your own tips and tricks on this forum for the benefit of other readers.