

ELEKTOR ELECTRONICS

THE ELECTRONICS & COMPUTER MAGAZINE

JUNE 2001
£2.95

www.elektor-electronics.co.uk



**Metronome/
Diapason**



**Modular Dot
Matrix Display**

**32-Channel
Analogue Input
Card for PCs**

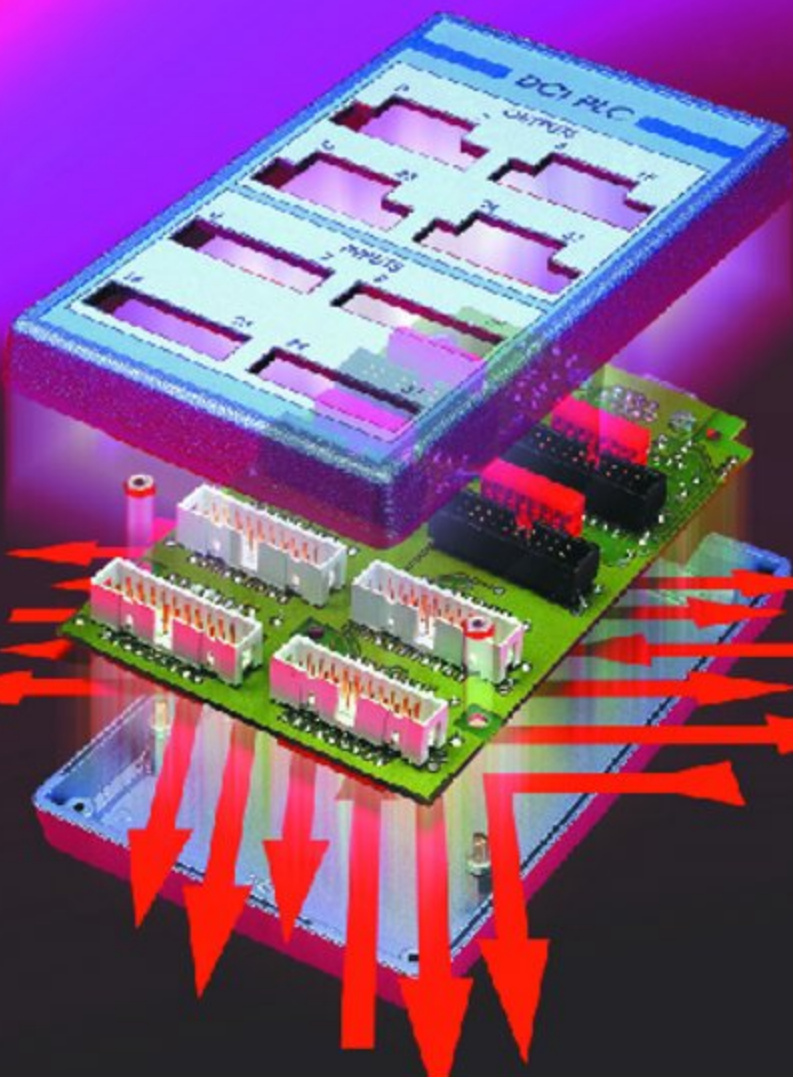
**CAD Software
Programs in
Europe**

In-Car SMPSU

**More MP3
Projects**

DCI PLC

a computer-simulated PLC



DCI PLC

a computer-simulated PLC

By I. de Coninck

PLCs are microcontroller circuit boards that are frequently used in a wide variety of industrial control and automation applications. The PLC described here is implemented in software. A complete PLC is simulated on a standard Windows PC. Using an external connection box, the PC can also sense and control many kinds of real items.



The abbreviation 'PLC' stands for 'Programmable Logic Controller'. These devices are also sometimes referred to as 'freely programmable controllers' in the trade. This term refers to an electronic circuit that can be used to automate machinery and processes. A PLC thus can replace a conventional relay logic circuit.

A PLC always consists of a CPU module and several digital input and output modules. The number of input and output modules depends on the size of the process to be controlled.

One of the major advantages of a PLC is that it makes it very easy to extend or modify the process. Consequently, PLCs have for years been an indispensable element in the automation of industrial systems.

Recently, PLCs have also started to appear in domestic applications. After all, people would also like to have certain things automated in their houses. Automatic garden sprinkler systems, awning con-

trollers, security systems and many other types of systems can easily be controlled using a PLC. Extensions or modifications can be made relatively easily after the system has been installed.

Figure 1 shows an example of a simple motor circuit, in one case built using pushbuttons and relays and in the other case using a PLC. If you want to modify the operation of the circuit, you will have to modify the hardware if you use the relay circuit, but with the PLC all you have to do is to make a few changes to the program.

The objective of the DCI PLC design is to implement a PLC in a relatively simple and inexpensive manner using a minimum of hardware, a serial interface cable and PC software. This saves the trouble and expense of building a real PLC. In addition, programming this controller is very intuitive and user-friendly.

The concept

DCI PLC is actually a program that replaces the CPU module of a PLC, and within which control programs can be developed and modified. With the aid of a serial interface cable and some hardware, the states of 32 inputs can be monitored and 32 outputs can be driven. There are also 32 timers, 32 counters and 32 flags. The program runs under the Windows 95 and 98 operating systems. The program itself will also work under Windows NT, but the hardware cannot be connected because NT does not allow direct hardware addressing.

The minimum system requirement is a 100-MHz Pentium processor. The installed software takes up approximately 3 MB, including the examples.

The hardware of this design consists of nothing more than an assembly that converts PC serial port information into a number of outputs and inputs. The actual input and output interface must be developed by the user. It can be connected to the DCI PLC interface circuit board using flat cables. It is not possible to design a universal interface that satisfies all possible needs, but most *Elektor Electronics* readers should not find it all that difficult to develop

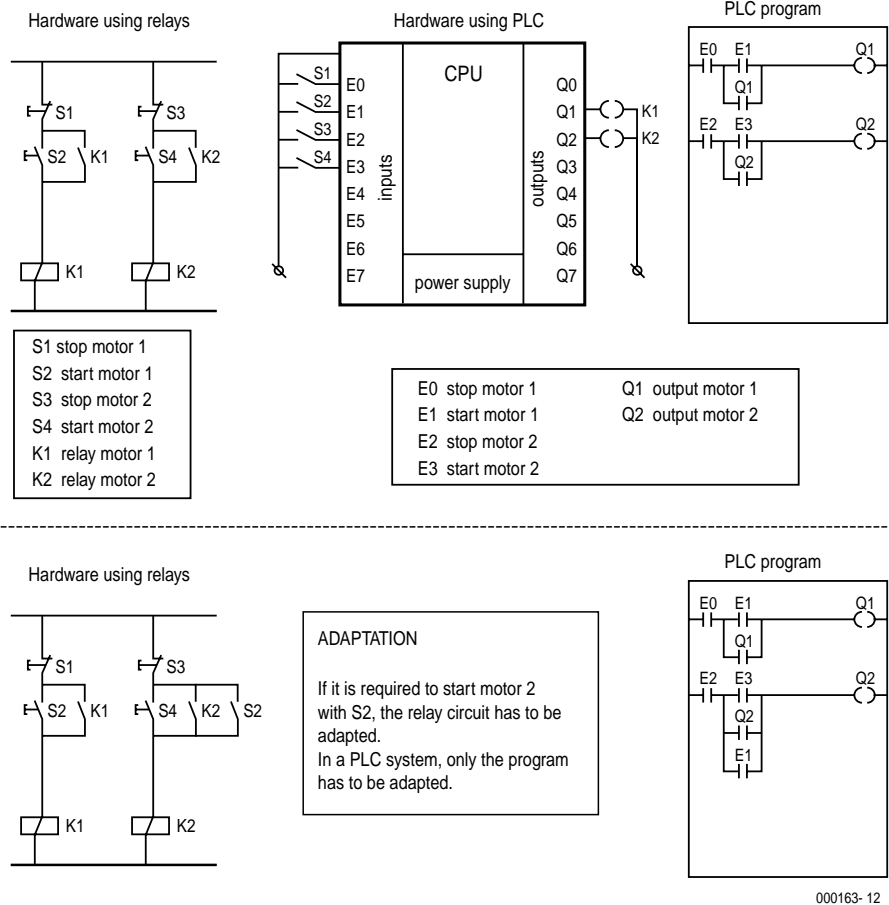


Figure 1. Example of a start/stop circuit for two motors, in one case using relays and in the other case using a PLC.

an interface for a specific application. In addition, at the end of this article we suggest some sample interfaces.

The hardware

If you look at the schematic diagram for the hardware (**Figure 2**), you will see that it actually consists of little more than a handful of CD4094 and CD4021 shift registers.

The 4094 shift register is an integrated circuit containing eight flip-flops connected in series. With each clock signal, the logic level on the input ('0' or '1') is shifted by one position, so that after eight clock pulses and one strobe pulse we have an 8-bit output signal available. In our circuit, four of these shift registers are connected in series (IC5-IC8). This gives us a total of 32 outputs.

The 4021 shift register has the

advantage (relative to the 4094) that can also be used 'the other way around'. It can thus be used to read in the levels on its eight inputs. The output levels of the eight internal flip-flops appear one after the other on the serial output when clock pulses are applied to the clock input of the IC.

We have chosen to use the PC serial port for the connection to the hardware because it offers several advantages, namely:

- The serial port is relatively robust (short-circuit protection).
- The cable may be connected while the PC is operating.

In the interface hardware, the RS232 signal levels are limited by Zener diodes D1-D3 to +4.7 V and -0.6 V. Resistors R1-R3 provide current limiting.

For more information about using the PC serial interface, see the series 'PC Serial Peripheral Design' (starting with the September 2000 issue of *Elektor Electronics*).

In addition to the shift registers, the circuit

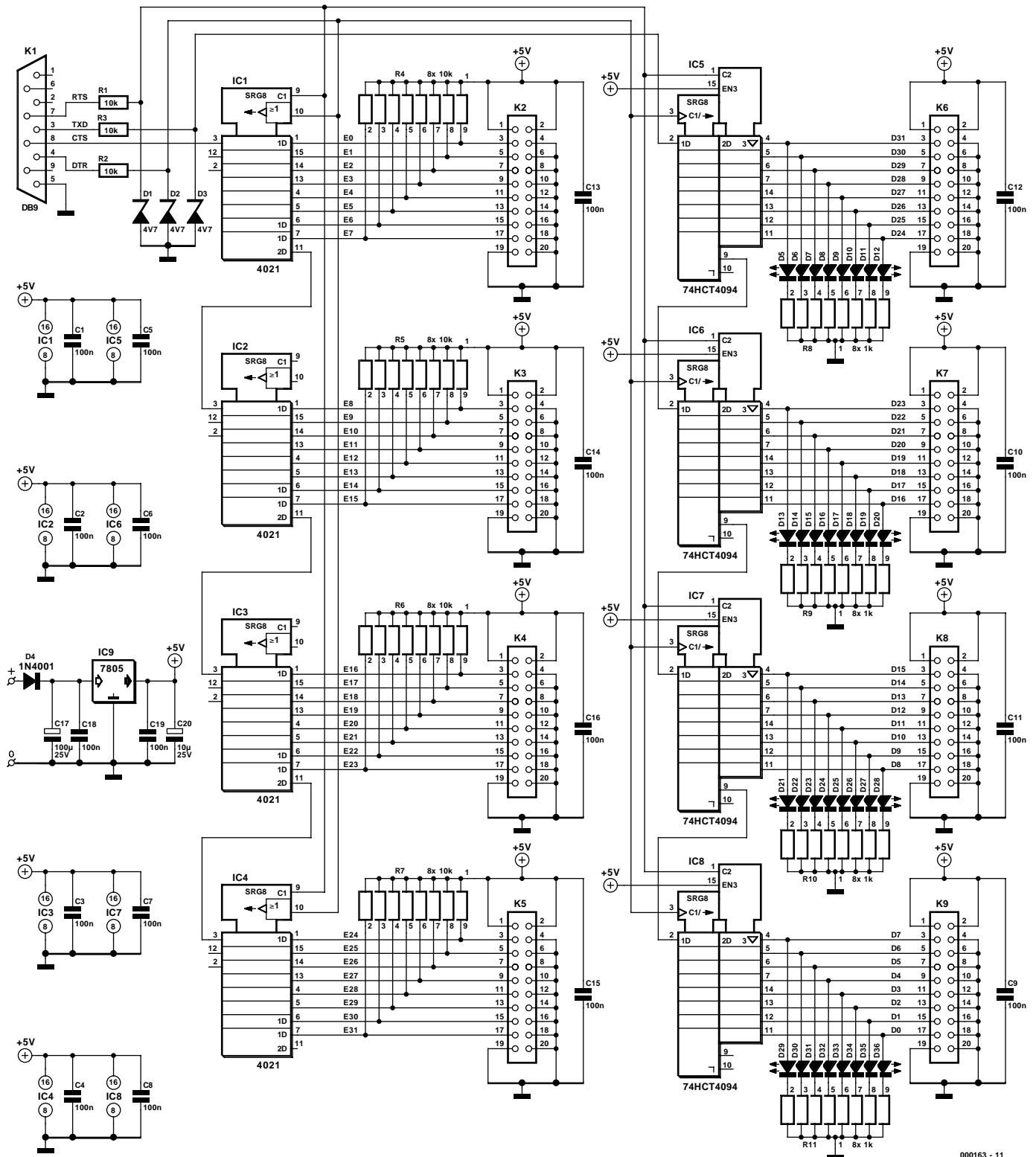
board includes a voltage regulator to provide a 5-V supply voltage. The input voltage may range from 10 V to 20 V. There are also 32 LEDs (D5-D36) with associated series resistors, which make the levels on the 32 outputs

visible. Finally, we can note that all 32 inputs are fitted with pull-up resistors, so that a High level is present on any unused input.

Figure 3 shows the layout of the

printed circuit board that has been designed for this hardware. It is a double-sided board, in order to avoid the use of wire bridges.

Assembling the circuit board is cer-



000163 - 11

Figure 2. The hardware primarily consists of a handful of 4021 and 4094 shift registers.

tainly not difficult. Since the ICs draw only a small amount of current, the 7805 does not need a heat sink. We have chosen to use rectangular LEDs, so that they can be placed

close together in groups of eight located exactly opposite the associated header pins. Standard 2×10 pin headers or box headers are used for connections to

the outside world. A length of flat cable can easily be connected to each header using a press-on (IDC) connector. Each connector has +5 V and ground on the two 'extra' pins, and a ground connection is present opposite each

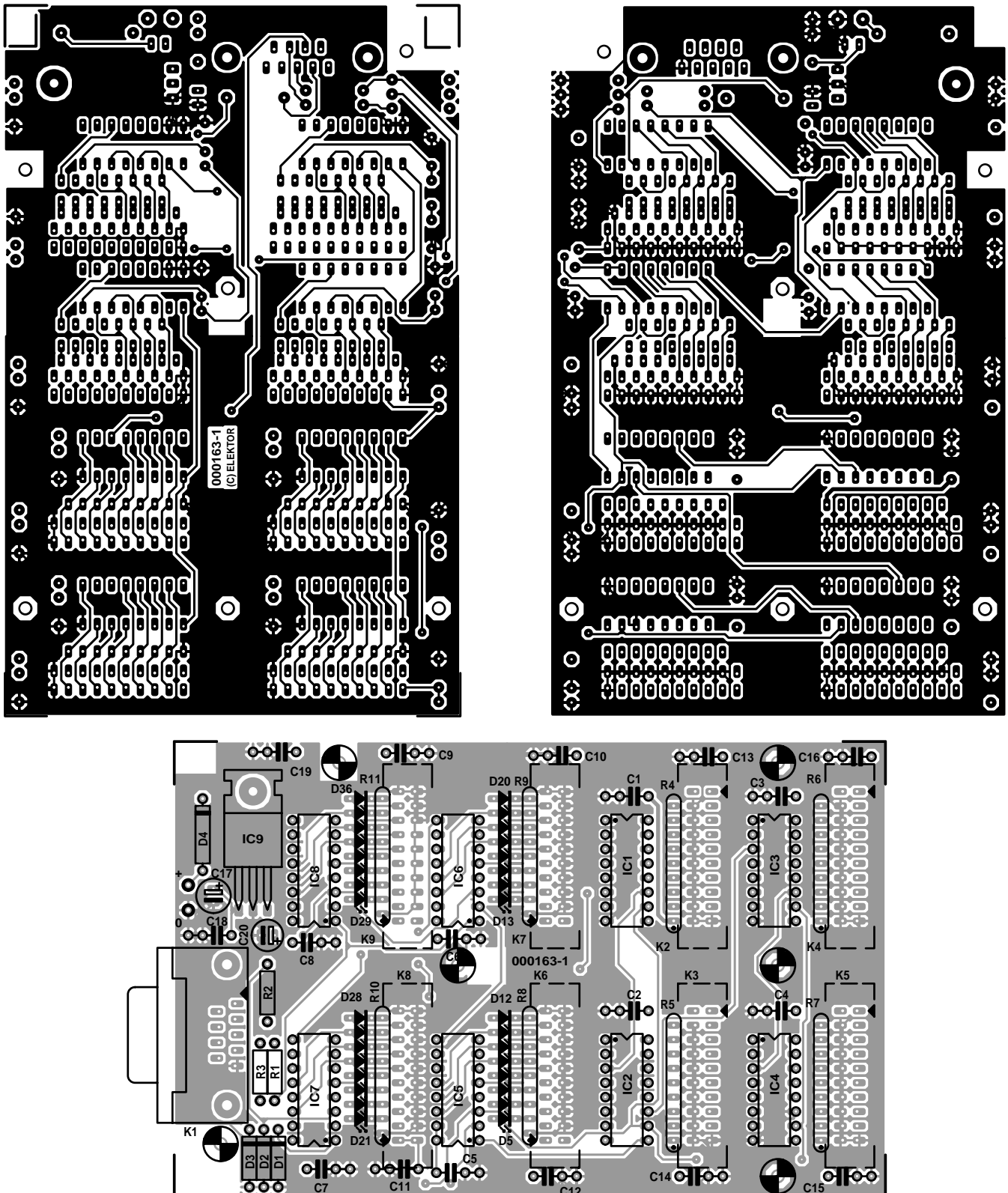


Figure 3. The double-sided printed circuit board for the DCI PLC.

COMPONENTS LIST

Resistors:

R1-R7 = 10kΩ
 R4-R7 = SIL array 8 x 10kΩ
 R8-R11 = SIL array 8 x 1kΩ

Capacitors:

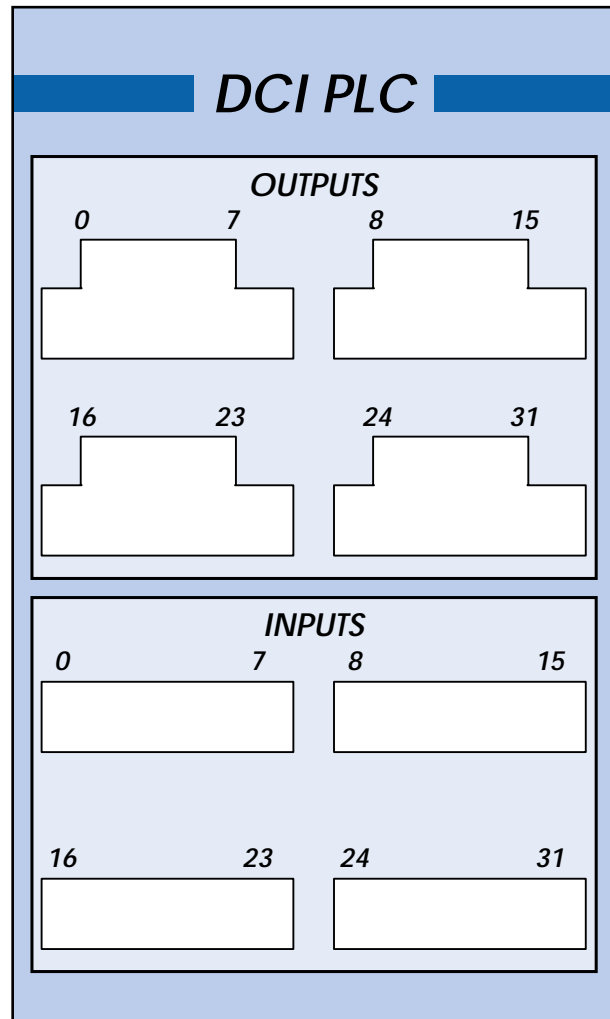
C1-C16, C18, C19 = 100nF
 C17 = 100µF 25V radial
 C20 = 10µF 25V radial

Semiconductors:

D1, D2, D3 = zener diode 4.7 V 400mW
 D4 = 1N4001
 D5-D36 = LED, rectangular, red, high efficiency
 IC1-IC4 = 4021
 IC5-IC8 = 74HCT4094
 IC9 = 7805

Miscellaneous:

K1 = 9-way sub-D socket (female), PCB mount
 K2-K9 = 20-way boxheader
 Case: Teko 160x95x45 mm (Conrad Electronics #526177)
 PCB, order code **000163-1** (see Readers Services page)
 Disk, Windows program, order code **000163-11**



000163 - F

Figure 4. Example of a matching front panel foil for the circuit board shown in Figure 3.

signal connection (see the schematic diagram). When assembling the circuit board, keep in mind that the connectors must be fitted on the solder side. This will make the numbering of the inputs and outputs match the numbering on the front-panel foil (see **Figure 4**).

If you want to finish the assembly a tidy manner, you can mount the circuit board in a small enclosure fitted with a front panel foil, such as the one shown in Figure 4.

Using the software

The DCI PLC program is designed to allow you to easily prepare a PLC program by simply placing blocks.

The length of the ladder logic diagram that is constructed in this manner is limited to 99 lines. Ladder logic diagrams, which are in many ways similar to relay logic diagrams, are frequently used to graphically generate programs. When putting together a ladder

logic diagram, you should always bear in mind that the symbols in the diagram represent instructions rather than physical contacts.

When the instructions in the program are executed, the states of the inputs are monitored. Depending on these states, various outputs are set either High or Low.

The instruction set

The instruction set of the DCI PLC consists of the following elements (see Figure 5):

1. **Test for ON.** This instruction checks whether a voltage is present. It can be assigned to an input, output, flag, timer or counter. The status of this instruction is TRUE whenever a voltage is present (if the instruc-

tion is assigned to an input or output).

2. **Test for OFF.** This instruction checks whether a voltage is absent. It can be assigned to an input, output, flag, timer or counter. The status of this instruction is TRUE whenever no voltage is present (if the instruction is assigned to an input or output).
3. **Positive edge detection.** The status of this instruction becomes TRUE when a rising edge is detected. This remains valid for the duration of only one program cycle. This instruction can only be assigned to a flag.
4. **Negative edge detection.** The status of this instruction

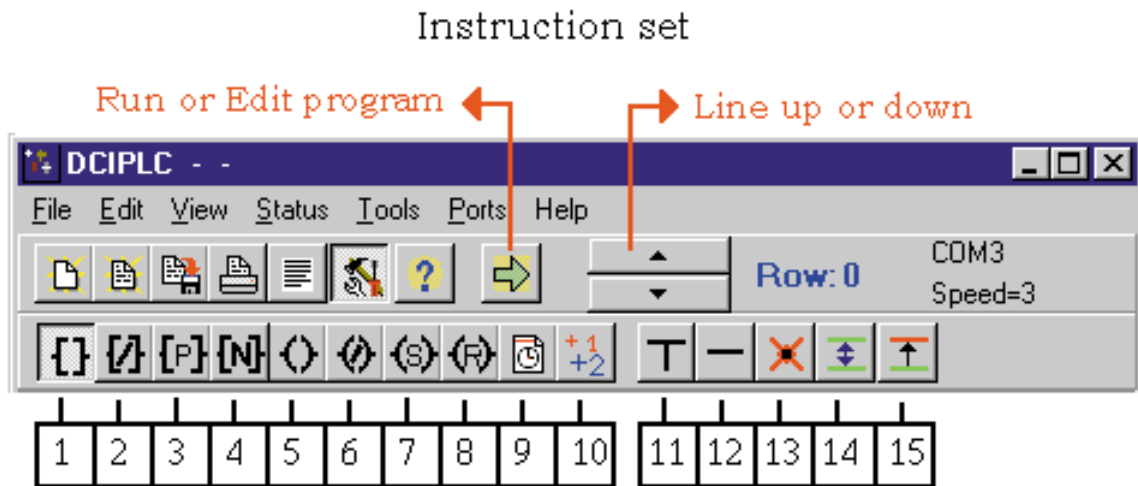


Figure 5. The toolbars of the DCI PLC program. The numbers under the buttons refer to the descriptions of these buttons in the text.

becomes TRUE when a falling edge is detected. This remains valid for the duration of only one program cycle. This instruction can only be assigned to a flag.

5. This symbol stands for an **internal or external output** and can be assigned to a flag or an output. This output will become High when the result of the logical evaluation of the entire row is TRUE.
6. This symbol stands for an **inverted internal or external output** and can be assigned to a flag or an output. This output will become High when the result of the logical evaluation of the entire row is FALSE.
7. **SET internal or external output.** This symbol can be assigned to a flag or an output. This output will become High when the result of the logical evaluation of the entire row is TRUE. Even if the logical evaluation of the entire row later becomes FALSE, the output will remain High. Only **RESET output** can change the output state back to Low.
8. **RESET internal or external output.** This symbol can be assigned to a flag or an output. This output will become Low when the result of the logical evaluation of the entire row is

TRUE. Even if the logical evaluation of the entire row later becomes FALSE, the output will remain Low. Only **SET output** can change the output state back to High.

9. The **timer** is an instruction that remains High or Low for a set interval (it can also be alternately High and Low). This instruction must always be placed in the last column of the row. The timer starts when the logical evaluation of the entire row is TRUE, and it continues to run as long as the logical evaluation remains TRUE. If the logical evaluation of the entire row becomes FALSE, the timer is reset. The variable *t1* represents the interval in seconds during which the timer is Low, while the variable *t2* represents the interval in seconds during which the timer is High. The variables *t1* and *t2* may have values from 1 to 9999. When a timer is assigned, a value must be assigned to *t1* or *t2*, since otherwise an error message will occur. A 'flasher' can be created by assigning values to both *t1* and *t2*. To implement very long intervals, a timer can be combined with a counter.

10. The **counter** instruction generates a High level after a defined number of pulses. Counting

starts when the logical evaluation of the entire row is TRUE, and it continues as long as the logical evaluation remains TRUE. If the logical evaluation of the entire row becomes FALSE, the counter is reset. The counter value can be preset from 1 to 9999. A counter has two inputs: one for counting up and the other for counting down. A flag, input, timer, output or counter can be assigned to each of these inputs. At least one of the inputs must receive an assignment. The state of the counter becomes High only when the count equals the preset value! The counter instruction must always be placed in the final column.

11. Start or end of an **OR instruction.** This instruction can only be placed in columns 1 through 8, and it must always have a start and an end. Only one OR instruction can be used per row, and the maximum span of an OR instruction is 7 rows. In the /examples folder on the diskette, there are two files (or.plc and bador.plc) that show what can be done (and what cannot be done) with an OR function.
12. **Line** instruction. This instruction is the equivalent of a signal line in a schematic diagram. It always shows the level of the previous cell.

The following tools are available for making programs:

- delete an instruction
- insert a row
- delete a row

Note: A 'flag' is a sort of internal output that

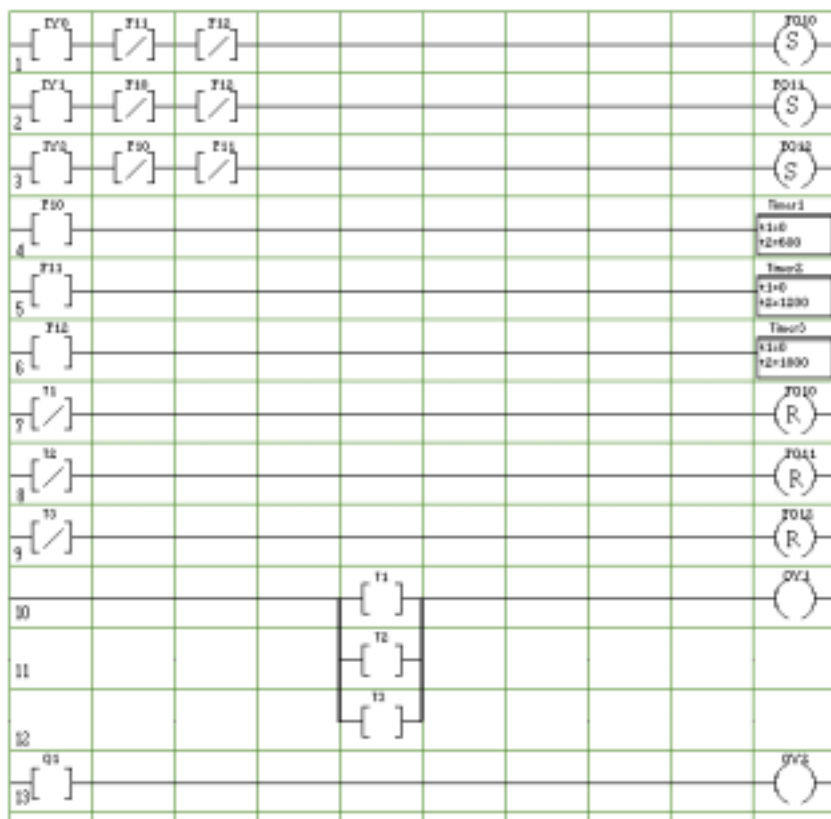


Figure 6. Structure of the sprinkler program.

is used to store an intermediate result. A flag can be 'set' and 'reset' just like an output, so it is effectively an invisible output. The program also runs under Windows ME.

Creating a program

When generating a program, it is a good idea to work from left to right and to only start a new row after the

current row has been completed. The second point is also necessary because the program checks for empty rows.

You should also make a habit of always assigning a particular output only once when generating a PLC program.

Every PLC works in the following manner:

1. When the PLC starts up (changes to Run mode), all outputs are set to inactive (no voltage).
2. The states of all inputs are read in.
3. The user program processes the instructions sequentially using the signal states acquired in Step 2. If the state of an input signal changes while the instructions are being processed, this does not have any effect during the cycle currently being executed. However, the states of timers, counters and flags are taken into account during the processing of the program.
4. After the user program has been processed, the results are sent to the outputs.
5. Steps 2, 3, and 4 are continuously repeated in a cyclic manner.

If the same output is used more than once in a program, this output will be assigned the state that it receives in the last row in which it is located! The DCI PLC program processes a complete user program 20 to 100 times per second. The actual rate depends on the speed of the PC and the length of the program.

An example

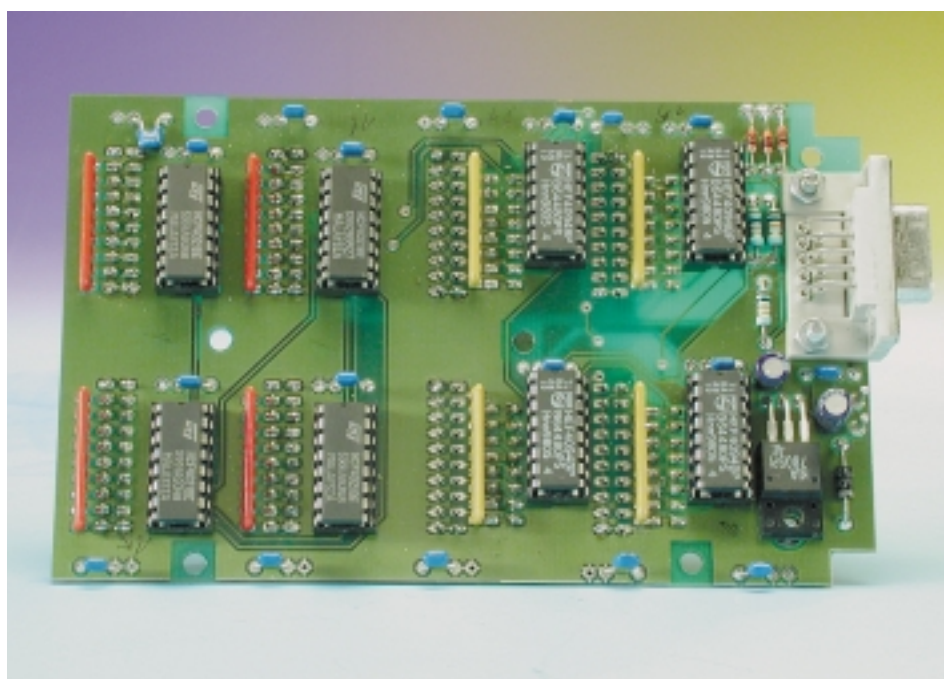
The following example is a simple sprinkler controller for a garden. We assume that two locations in the garden must be watered. We want to be able to choose from three options, depending on how dry the ground is: a short sprinkling time, a long sprinkling time and a very long sprinkling time. This is what we need:

Two sprinkler valves:

- QY1: Output Yes no. 1
- QY2: Output Yes no. 2

Three selection pushbuttons:

- IY0: Input Yes no. 0, for 10 minutes' sprinkling



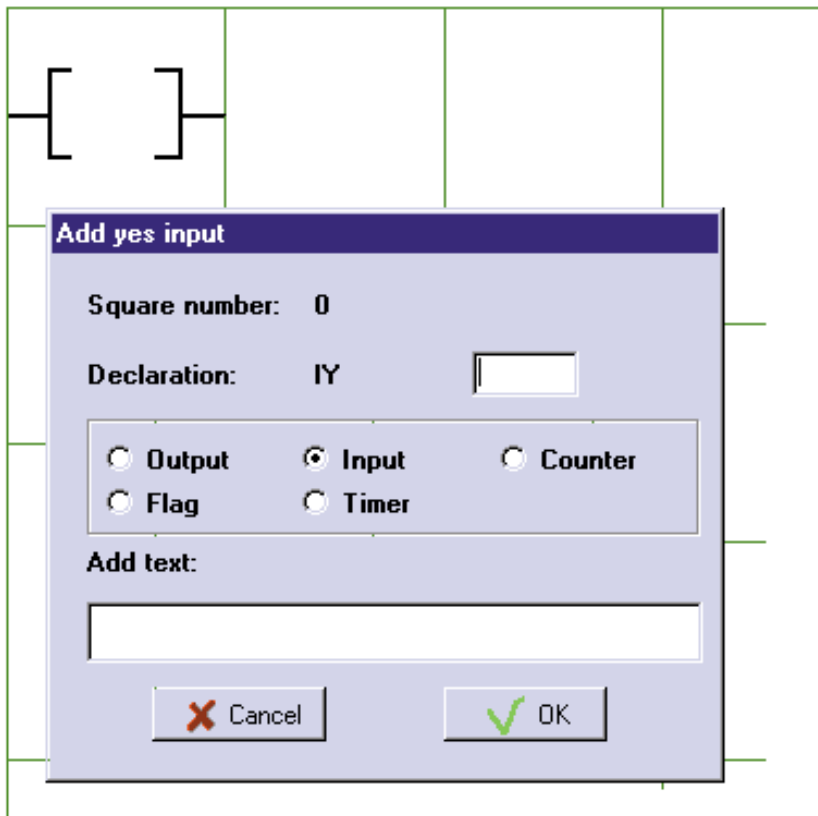


Figure 7. When an instruction is placed, a dialogue window appears for the assignment, optional comment and so on.

- IY1: Input Yes no. 1,
for 20 minutes' sprinkling
IY2: Input Yes no. 2,
for 30 minutes' sprinkling

tion (2) in each of the following two cells, and assign these instructions to F11 and F12.

Now let's see how to enter the program.

First, in the DCI PLC program, select the button **Test for ON** in the toolbar (number 1 in Figure 5) and click on cell 0 in row 1.

The chosen instruction will now be located in this cell, and a dialogue window will appear to allow you to assign the input and add a comment. In this dialogue, select 'Input' and enter '0' as the assignment. This will cause input 0 of the PLC to be assigned to this instruction. If you wish, you can enter a comment, which can later be viewed by clicking on the cell with the right-hand mouse button.

If you now click on 'OK', input 0 will be assigned. This is shown by the fact that 'IY0' appears at the top of the cell.

Next, place a **Test for OUT** instruc-

F11 and F12 are flags that will be set and reset later on in the program. They are used to prevent another time from being chosen after one of the pushbuttons has been pressed.

Now select the **Line** instruction (12) and click once in cells 3 through 8. The lines have now been placed.

Select the **Set output** instruction (7) and click in cell 9. The output instruction will be placed and a dialogue for assigning an output will appear. Select 'Flag' and enter '10'. You can also add a comment if you wish.

Now you can see that the first row is already finished. Generate the remaining rows in a similar manner, as shown in **Figure 6**.

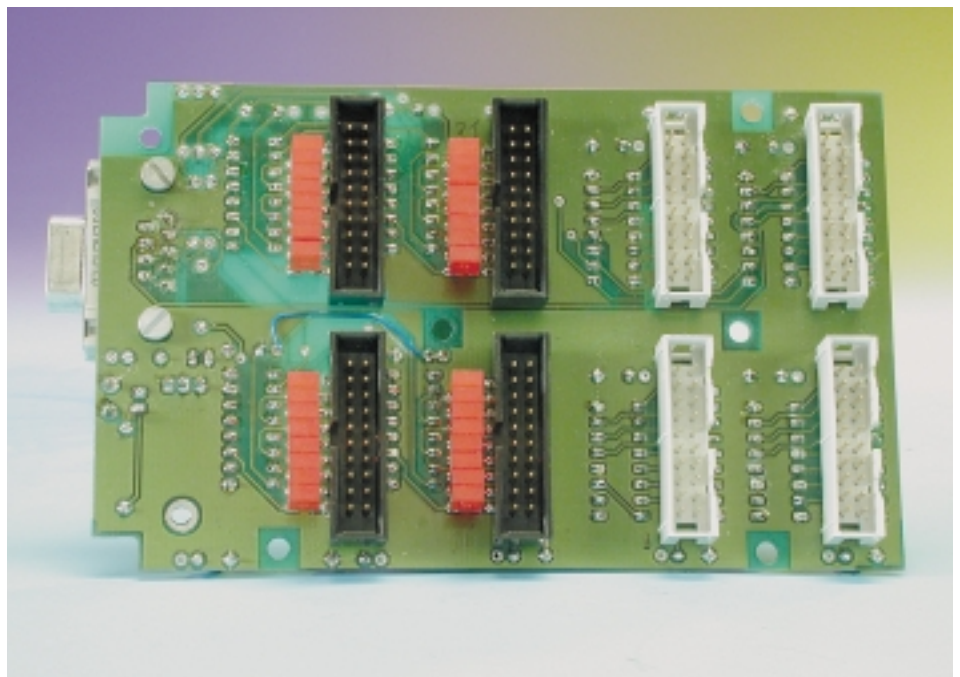
Once you have finished, you can save the program, and it is then ready to be tested.

To test the program, or to run it without the hardware, you can use the simulation mode. To do so, click on 'Simulation mode' in the 'Status' menu. A window will appear, in which you can enter the states of the 32 inputs and check the states of the outputs. Set input 1 High and then Low. This starts timer 2, and outputs 1 and 2 will then remain High for 20 minutes.

The program gives a visual representation of the states of the inputs and outputs and also displays the current values of the timers.

Once sprinkling has started, setting one of the inputs High no longer has any effect. This is because rows 1, 2 and 3 have been blocked.

After the hardware has been connected, you



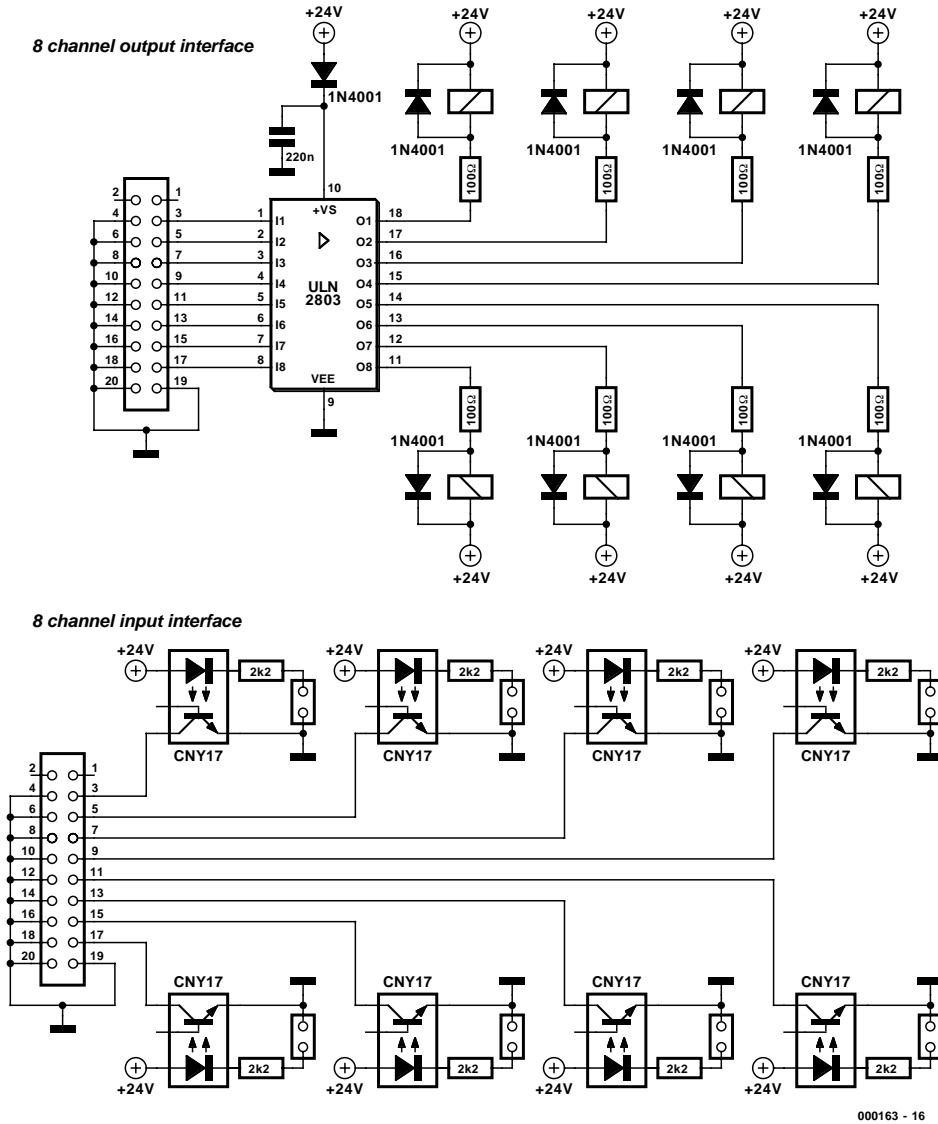


Figure 8. Two examples of possible interfaces: (a) eight relay-driven outputs; (b) eight optocoupler inputs.

interval, the timers are no longer updated and inputs are not checked. If you observe the above recommendations, the program will check the inputs at least ten times per second and the timer error will not exceed two seconds in 24 hours.

On a 400-MHz Pentium machine, the sprinkler program achieves a speed of 90 cycles per second.

The diskette containing the program (which is also available from the *Elektor Electronics* website) also includes some other simple examples that you can use for practice.

Interfacing

For coupling the controller to various types of devices, relays and sensors, readers of *Elektor Electronics* can draw on a vast store of circuits that have been published in earlier issues. Still, we would like to provide a pair of simple examples.

Figure 8a shows an 8-channel output interface consisting of a buffer IC and eight relays. This can be used to supply power to eight devices or circuits (with the proper choice of relay, it is even possible to switch the mains voltage, but you must make sure to use a safe construction).

Figure 8b shows an 8-channel input interface. This consists of eight optocouplers with associated current-limiting resistors. If the two terminals are connected together, a High level appears on the corresponding PLC input.

(000163-1)

can make the proper port settings using the 'Ports' menu.

You can choose from COM1-COM4, and you can also set the speed. It is a good idea to initially set a low speed and then check in RUN mode whether the inputs generate the proper responses and the outputs are correctly driven. Click on 'Run mode' in the 'Status' menu, or click on the toolbar button with the green arrow.

You can return to the edit mode at any time by clicking on 'Edit mode' in the 'Status' menu or by clicking on the button with the red circle. This causes all hardware outputs to be reset to zero.

The following tips will help you obtain the best results from the program in the Run mode:

- Make sure that as few programs as possible are active.
- Disable screen savers.
- Do not start any other programs.
- Do not install a diskette or CD-ROM in the PC while the program is running.

These measures allow Windows to reserve the maximum possible amount of time for the DCI PLC program, which has a beneficial effect on its performance. If a CD-ROM is inserted in the tray of the CD player, for example, the DCI PLC program will stop responding for a few seconds, since Windows gives priority to the CD-ROM drive. During this

The author has his own website for the DCI PLC:

<http://home.planetinternet.be/~dc11cd/>

PCB Drilling Machine (4)

Part 4: mechanical construction

Design by T. Müller (Radix GmbH)

www.radixgmbh.de

Now we come to the mechanical assembly of the PCB drilling machine. The components come ready-made: all that is needed is a little care in fitting them together.

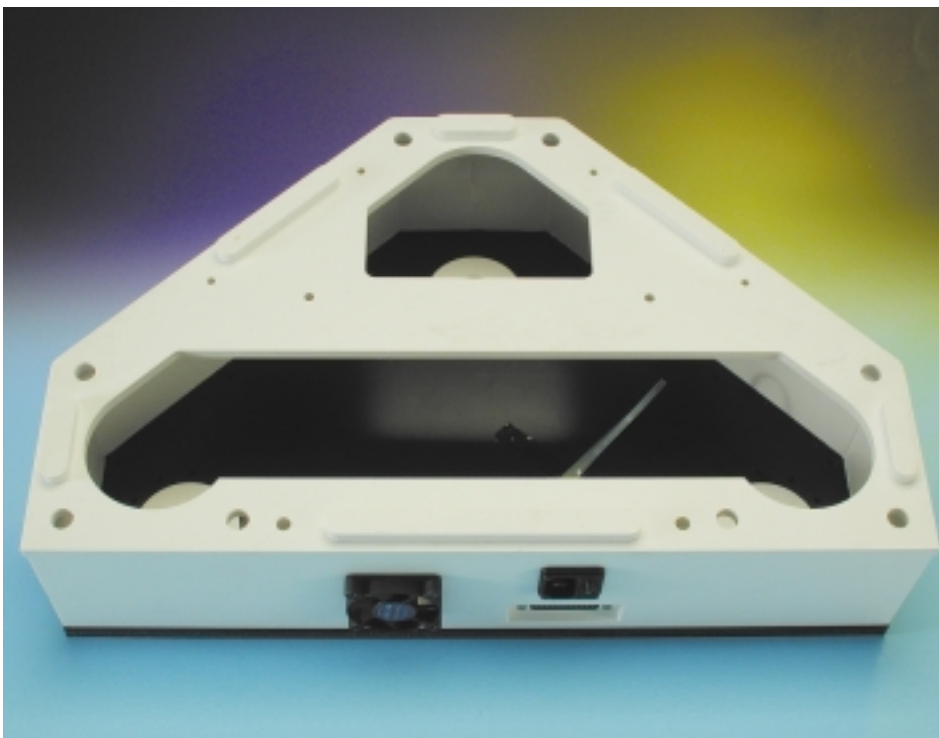


Figure 1. The machine enclosure, with some parts already fitted.

The kit of mechanical parts for the PCB drilling machine consists of ready-drilled and machined parts that simply have to be fitted together. All the parts required to build a machine with one tool arm are included in the

kit, even down to the necessary screws and glue. Not included, however, are any drills or milling tools that you may want to fit to the machine.

A toolkit suitable for constructing the machine can be ordered from Radix. It contains all the required tools in the correct sizes: slotted and cross-head screwdrivers, box spanners, open-ended spanners, Allen keys, brushes for removing swarf, cleaning cloths, side cutters, blades etc. All the tools are of the highest quality. By purchasing the complete set you can be sure that you will always have the correct tool to hand during construction.

All the components - except those that are glued together, of course - can be subsequently disassembled, and so the exact order of assembly is unimportant. However, the sequence suggested below has been found to be the simplest.

PREPARATION

All the components which are to be glued together are either slotted or have milled tongues that fit into the slots. All the parts are supplied perfectly free of dirt and grease, and therefore need not be cleaned. Unnecessary handling of the areas to be joined attracts dirt and grease which make the glue much less effective.

After milling a certain amount of swarf and other material may remain on some edges and corners, which must be removed. Vigorous brushing will quickly remove the unwanted material. Alternatively, you can use compressed air at at least 8 bar, but only if you are absolutely certain that the air is perfectly free of oil. If you do get oil on the parts, you will find it impossible to clean them adequately, and there is no point in even trying to glue them together: you will have to order a new kit of components. The best way to remove unwanted material from the parts is with an ordinary stiff toothbrush (although not one with a flexible head!). Thoroughly brush the corners and joints until all the swarf is removed. Mould flash may remain around some parts: this can be cut away using a craft knife.

You will need an absolutely flat, perfectly clean working surface at least 50 cm square. A useful tip is to use an (undamaged) offcut of plain-coloured plastic-coated kitchen worktop, which can be bought at the DIY store for a pound or so, or perhaps even for free. Offcuts of fibre-board, for example, are generally not suitable, because they are generally warped.

STEP 1: GLUING THE MACHINE ENCLOSURE

We start by gluing the white machine base plate to the three side cheeks. These parts are large and thick, so we can use them as a gentle introduction to the technique of gluing a concealed joint. All the other parts are glued in exactly the same way. After this gentle warm-up, we will be able to move on to the rather fiddlier job of gluing the drilling arms.

Clear away all the drinks and greasy foods from the working area. Put away everything that is not needed for the gluing operation and anything that might distract you while working. Your hands should be clean and as free of grease as possible. Best (although somewhat uncomfortable) is to wear light working gloves made for exactly this purpose, which keep dirt and grease from the objects being glued.

As a trial, bring together the three

All about glue

The degree of rigidity and stability that can be obtained with a glued construction is significantly higher than can be obtained with screws. The assembly is as strong as if it were made from a single piece of material. All joints must be left to dry for at least 24 hours to ensure a rigid construction. Assembly proceeds in two stages: gluing and then screwing, and should therefore take at least two days.

'Gluing' is not really the correct term here: it would be nearer the mark to say that the plastic components are 'welded' together. The process can only work if the glue and the plastic are perfectly compatible with one another.

Some chemicals in glues tend to dissolve certain plastics. Often this effect is not desirable: if you have ever accidentally tried to clean a plastic surface with an inappropriate solvent, you will know what we mean. Acetone, although excellent for removing grease, dissolves many plastics including Plexiglas (a.k.a. perspex).

Acetone alone is not viscous enough to be a satisfactory glue, and it evaporates too quickly. Neither, unfortunately, does it dissolve PVC (polyvinyl chloride).

The plastics we wish to bond are polystyrene (PS) and various polyvinyl chlorides. Gluing PS to PS or PVC to PVC presents no difficulties, as specialised glues are available for each. However it is difficult to bond one to the other: PVC was developed to be resistant to practically all chemicals, including those used for gluing. After hundreds of experiments with fourteen different types of glue we finally came to the conclusion that there was no single suitable glue for our application.

However, we did find a glue composed of seven chemicals that can bond PS to PVC. If the truth be told, it is a rather unpleasant mixture and we were rather surprised that it wasn't an explosive! Delivering such a mixture creates some problems since there is practically no type of plastic suitable for making a container for it.

Fortunately our problems were solved when a company brought out a new glue which meets our requirements in terms of viscosity, spreading characteristics and tendency not to drip or form filaments, and which can bond PS to PS as well as PS to PVC.

More than enough of this glue is included in the kit to build a drilling machine with one tool arm. Be generous with the glue, since using too much will not do any harm. To reiterate: use only the supplied glue, even if you see some cheap-looking alternative 'suitable for most plastics' in the model shop or DIY store. Other glues are simply nowhere near as strong.



Figure 2. In a single motion, draw out a line of glue along the full length of the joint.

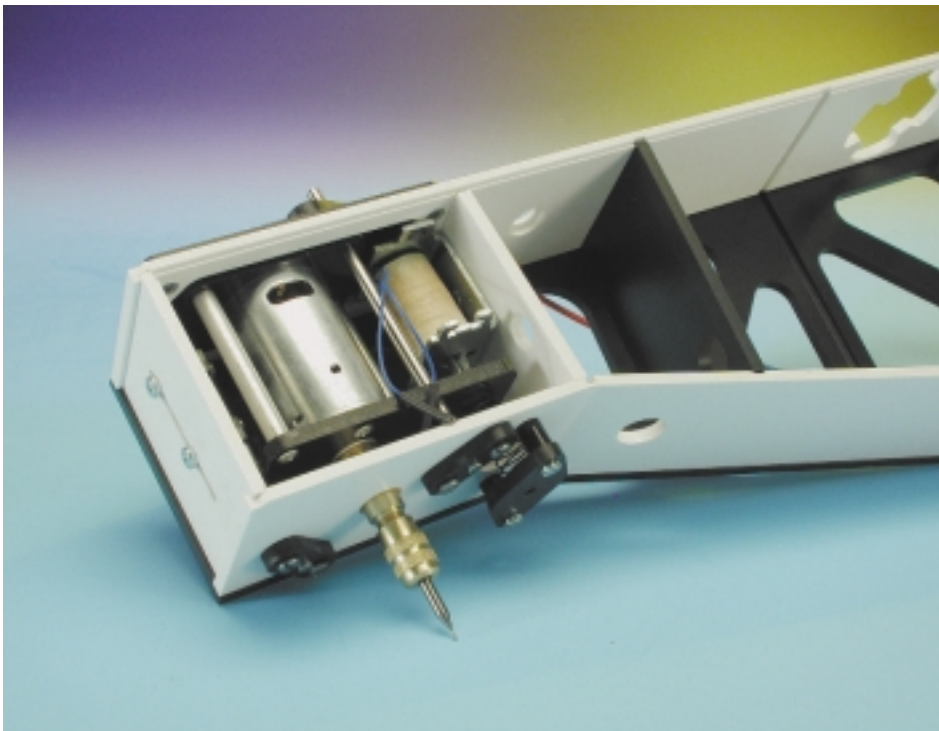


Figure 3. Partially-glued tool arm with drill motor and solenoid already fitted.

side panels, the white base plate and the black top plate (**Figure 1**). First put the base plate down on the working surface with the joints facing upwards and then guide the rear panel into its slot. Hold the first side panel at a slight angle and fit it first to the slot in the rear panel and then into the slot in the base plate. The side panel is now held at two

points. Repeat the procedure with the second side panel.

Now fit the black top plate to the three side panels. With a little toing and froing the top plate will seat on the side panels. Check with a ruler that the enclosure is the same height everywhere and hence that the top



Figure 4. Tool arm, immediately before gluing of second side panel.

plate is indeed properly seated.

Now pick up the whole enclosure, holding together the base plate and top plate and inspect it all around and from inside. It should feel solid and stable.

Before reaching the point of no return, go through the following checklist:

- Have you removed all unwanted material and are the joints clean?
- Do the parts fit together smoothly without jamming?
- Do the side panels fit cleanly without fouling the rear panel?
- Imagine that there is already glue in the joints and look at where your hands are. How far are your fingers from where the glue will be?

When you are convinced that the parts fit as well as possible, you can begin gluing. Otherwise, take the assembly apart and try again, until everything can be fitted together smoothly.

So much for practice; now for the real thing. Lay the five parts on the working surface. In a single motion, apply a line of glue along the full length of the joint in the rear panel, as shown in **Figure 2**. Fit the rear panel LOOSELY and without force onto the joint in the base plate. It is better if the rear panel does not stand perfectly upright but rather leans slightly backwards. Apply glue also to the groove in the first side panel and to the groove which will fit with the rear panel. Fit the side panel along these joints, again starting at the rear panel. Proceed in the same way with the other side panel.

Offer up the top panel, **without** glue, until it seats properly. This is just to hold the assembly square: it will be glued later. Before finally gluing the lid to the white parts of the machine enclosure, we will fit some of the parts inside. In principle this could be done after the enclosure is finished, but the job is easier without the lid fitted.

Leave the assembly for at least an hour without moving it, so that the glue can set.

STEP 2: GLUING THE TOOL ARM

Take the nine parts for the tool arm

out of their packaging: two black side pieces, a white base plate and a white top plate, and three white and two black reinforcing side pieces. As before, clean the parts with a brush. Before putting the parts together, ten M3 nuts should be fitted into the holes provided. It is easiest to put the nuts on the table and push the plastic sheet down on them. The photograph (**Figure 3**) shows how these are used later to attach the spindle/magnet unit; in the procedure we are following here, we fit only the nuts, leaving the drilling head to be fitted afterwards.

Place one of the black side panels in front of you, preferably with the straight edge at the back. Fit the white top plate. All the joints here use the same tongue-and-groove arrangement with the other panels as the rear panel in the machine enclosure. Fit the two small external side pieces, again beginning with the joint with the top plate, proceeding to the side plate and finally with the three interior panels. Fitting the base plate, which is fixed at two points, begins at one side: then follow around the side panel. Ignoring the drilling head for the moment, the arm should now appear as shown in **Figure 4**.

Now — surprise, surprise — fit the other side panel and seat it firmly by pushing it gently to and fro. Even without gluing, the arm is surprisingly stiff. If you are happy that everything is fitting smoothly, proceed to gluing; otherwise, take it apart and try again.

Try not to get glue on your fingers, or else you will leave fingerprints all over the arm enclosure. If this does happen, the best thing to do is let the glue dry and then scrape it away with the craft knife.

When all nine parts of the arm are glued together, it must immediately be checked for squareness. Lay it on each side in turn on the flat working surface. If it is not true, push down hard on the assembly to force the parts together.

As we shall see later, the construction of the arm includes a means of ensuring that the tip of the tool (and that, after all, is the only point that matters) can move in a straight vertical line. Faults in the gluing which make the arm out of

true can be compensated for, but of course this is best not relied upon.

Construction of the arm will take a good hour or so; it is a good idea to take a break at this point. Next we shall turn our attention back to the black top plate of the machine enclosure, which is sitting, unglued, on top of the side pieces. We lift the top plate carefully off and put the glued parts to one side.

STEP 3: FITTING THE MECHANICAL PARTS TO

THE ENCLOSURE LID

Take the pack containing the four cone bearings and the two packs labelled 'Mechanik 1' and 'Mechanik 2' out of the packaging. These comprise a complete set of mechanical parts for one axle (**Figure 5**). The cone bearing consists of two separate parts, an outer race and an interior part with the rollers. The bearings are rather oily: this oil should be cleaned away except on the race and on the inner part, so that only the moving parts are lubricated. Push two of the races into the holes in the enclosure top plate, one for the turntable and one for an arm. Push the bearing some distance through the hole so that about half a millimetre protrudes on the other side.

Fit the long M4 countersunk screws in the four 4 mm holes in each race, each with a precision washer. Now turn the top plate over. As long as the screws do not fall out, you should now see the ends of eight screws, onto which you should fit the 10 mm thick cable supports. This is where the connecting cables for the tool arm will be run.

Press both remaining races into the two remaining black bearing flanges and as before push them through so that about half a millimetre protrudes. The bearing flanges are fitted over the top of the cable supports. Now fit a precision washer and an M4 nut on the end of each screw. As the nuts are tightened each race will be lifted slightly and brought into exact alignment, flush and square in its hole. Now slacken the nuts by a quarter to half a turn. The nuts should not rattle, but the components should now be able to move slightly from side to side. The



Figure 5. The cone bearing consists of two components: an outer race and an interior part with rollers.

whole assembly is shown in **Figure 6**.

Now take the pack with the short aluminium shaft out of the packaging. Here you will also find a neoprene ring, three curved black securing pieces, a white plastic securing washer, and a large plastic sheet, with three M6 screws, to which the turntable will later be fixed.

Screw the large plastic sheet loosely to the aluminium shaft so that it has about 1 mm of play vertically.

From the other end of the shaft fit the interior part of a cone bearing and pass the shaft from the top side of the enclosure top plate into the race.

Fit the second interior part of the cone bearing to the end of the shaft now protruding under the top plate, and the neoprene ring immediately behind it. Behind the neoprene ring fit the white plastic securing washer, and the three black plastic securing pieces will fit exactly in the groove in the aluminium shaft. Now fully tighten the three screws in the large plastic sheet and check that the securing pieces are not bending the securing washer (**Figure 8**).

The parts in the pack with the long alu-

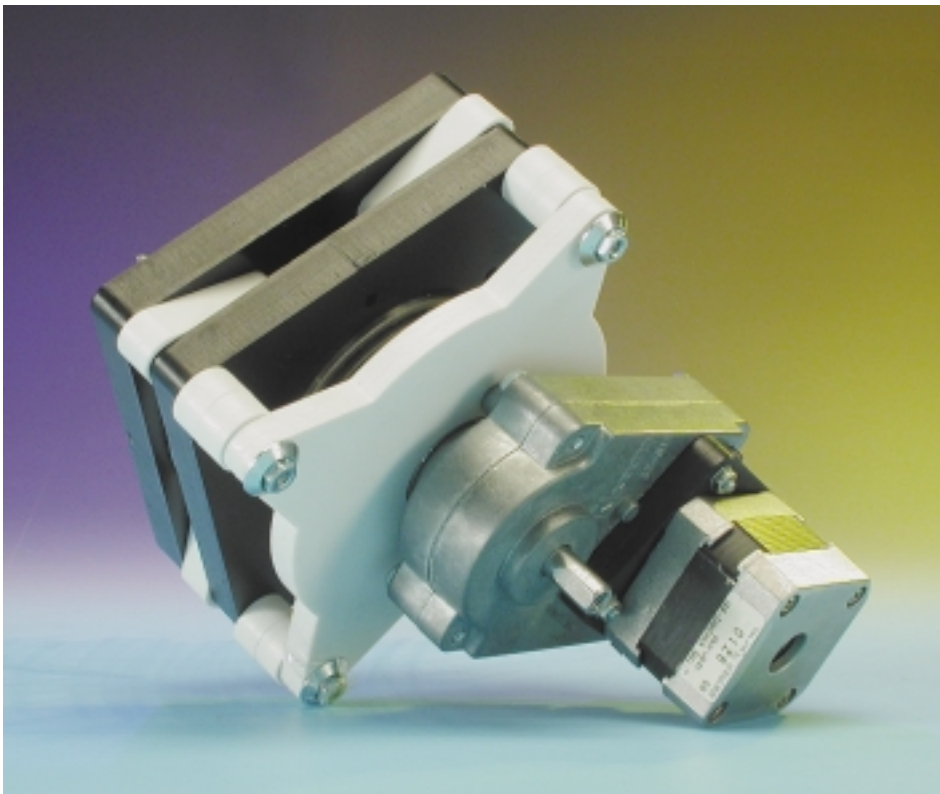


Figure 6. The motor is connected to the gearbox via a flange screwed to each, and the star-shaped drive fixing is screwed to the output of the gearbox.

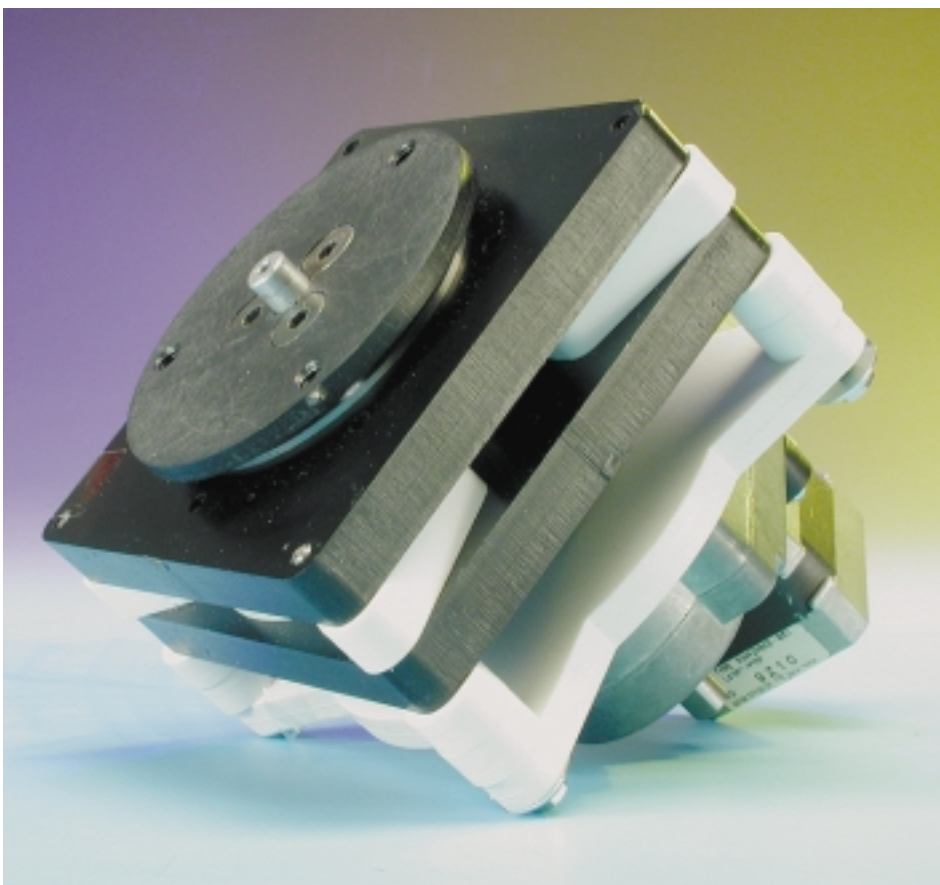


Figure 7. Completed drive unit viewed from the opposite side.

minium shaft are assembled in exactly the same way, except that here there is no large plastic sheet or attached screws.

Turn the two shafts gently to and fro until they move smoothly. In this way the bearing flanges, which are still loose, will settle into their optimal positions. You can also push gently on the bearing flanges themselves and then, by turning the shafts, find the best position. The flanges can then be fully tightened using a spanner.

Looking down on the bearings you can see four open holes, in each of which an M4 bolt (from pack 3) should be fitted. The nuts and washers from this pack will be used later for fixing the drive unit.

All the above work can of course be carried out with the top plate fixed to the rest of the enclosure. However, it is much easier with the enclosure open, and so it is only now that we glue the lid to the enclosure. Apply glue to the joints on the side pieces of the enclosure and fit the lid, ensuring it is properly seated.

All the gluing has now been done. Leave the enclosure and tool arm assemblies to dry for a day.

STEP 4: BUILDING THE MOTOR DRIVES

The two drives are built identically. They consist of a stepper motor and a gearbox, which is sealed against ingress of dirt: the gearbox should not be dismantled. To connect the motor and the gearbox we use a coupling (in pack 4) which is fixed to the motor with four M3 screws and to the gearbox with two M3 screws (**Figure 9**). Screw the star-shaped drive fixing firmly to the output of the gearbox with four M4 screws. The now complete drive unit can be fitted from below into the machine enclosure, the four bolts protruding from the bearing flange fitting exactly into the holes in the drive fixing. The whole assembly is now firmly screwed together using the nuts and washers from pack 3.

Assembly of the drive is now complete.

STEP 5: BUILDING THE DRILL LIFT UNIT

We now turn to the Z-axis drive, i.e. the lift unit for the drill. All the required components are in pack 5. First the drill motor is fixed to the longer of the two black plastic plates from the shaft side using two M3 screws. On the motor side the two long pillars are fixed to the plastic plate with M3 screws. The second black plastic plate is fitted over the motor housing from behind the motor and fixed firmly using the pillars. The two precision machined hardened steel shafts are fitted through the 4 mm holes drilled in the plastic plates and positioned so that they protrude equally above and below (**Figure 10**, here shown with drill motor but without lifting solenoid).

Before the unit can be fitted in the arm, the solenoid and the two limit switches must be screwed on. Then the unit can be lowered into the arm so that the drive shaft of the motor protrudes from under the arm. The arm assembly can now be closed off by fitting the top plate. The screws on the top plate must be tightened fully so that they do not shake loose in operation. Although the unit looks finished, the lift mechanism will wobble alarmingly in the arm. This play is reduced by fitting bearing blocks (to be found in pack 6) above and below the shafts now protruding from the arm assembly: fix these loosely at first using two M3 screws with washers (see **Figure 11**). A spring (from pack 6) is now fitted in the hole provided in the underside of the arm, and the hole closed off by fitting the cover with two spacers and screws. Check that the spring is seated properly in its recess, and push it into position if not. With the chuck attached to the motor shaft the arm assembly is complete (**Figure 12**).

STEP 6: FIXING THE ARM AND TURNTABLE

Fit the arm to the long 30 mm aluminium shaft and turn it until the 10 mm transverse hole in the shaft points in the direction of the motor spindle on the arm. The wiring will pass through this hole down into the machine enclosure. The arm is fixed to the shaft using a tapered grip, which will hold the arm absolutely

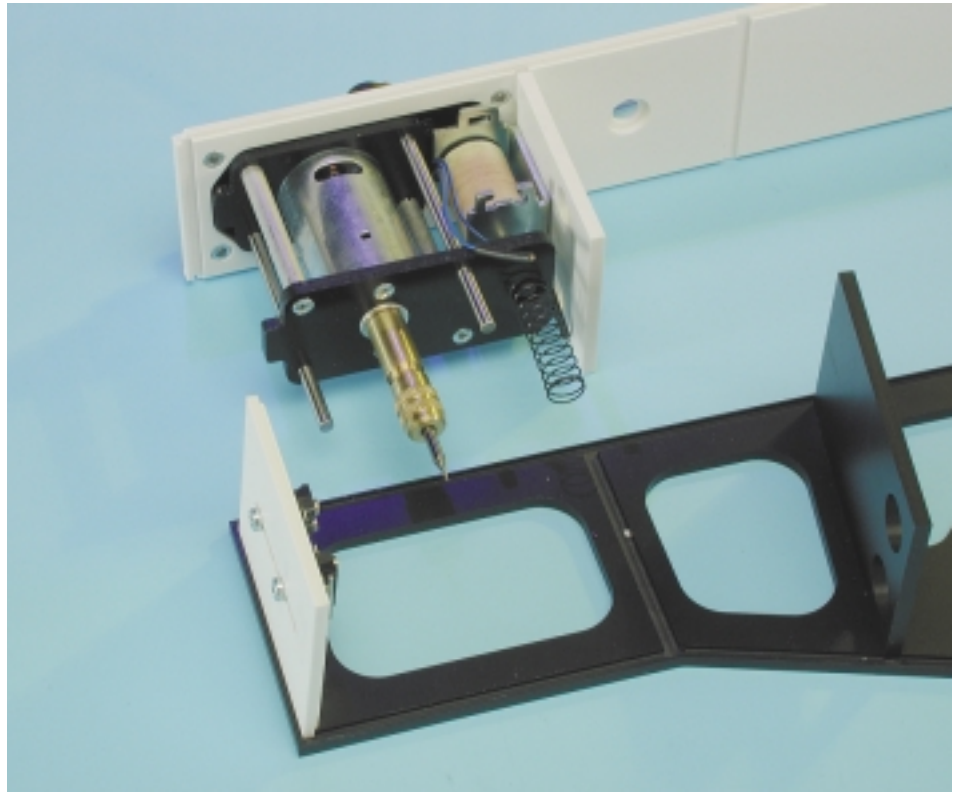


Figure 8. Fitting the drill lift assembly.

rigidly to the shaft when the shaft top plate is screwed down.

The last major component, the turntable, must be carefully and

evenly pressed down over the centring peg found at the end of the aluminium shaft onto the circular plastic plate and screwed down firmly using M6 screws. The small white reg-



Figure 9. Bearing blocks and spring hole cover on the underside of the arm. There are two identical bearing blocks on the upper side.

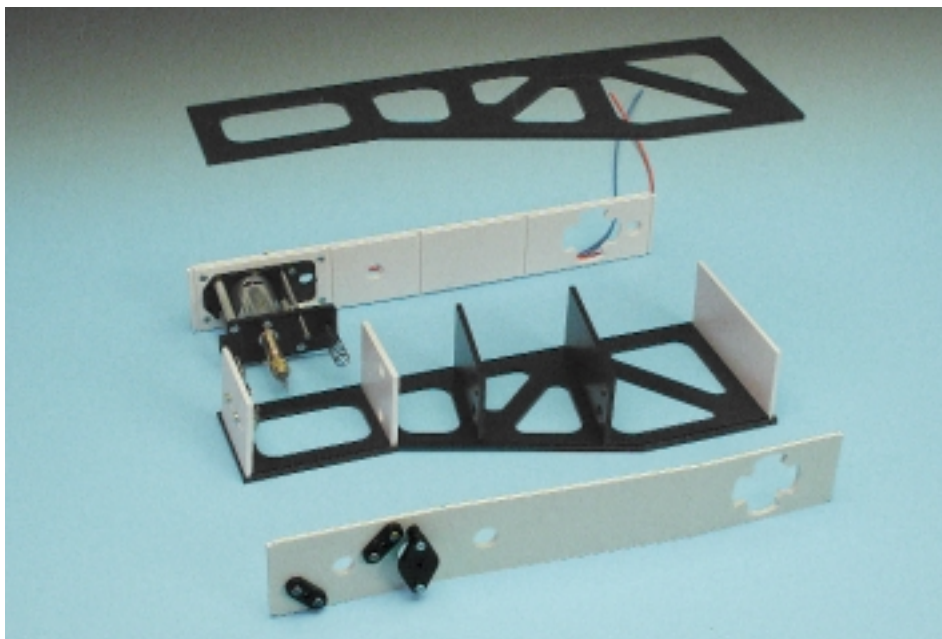


Figure 10. The parts of the arm laid out as they are to be glued. The lifting solenoid is not shown.

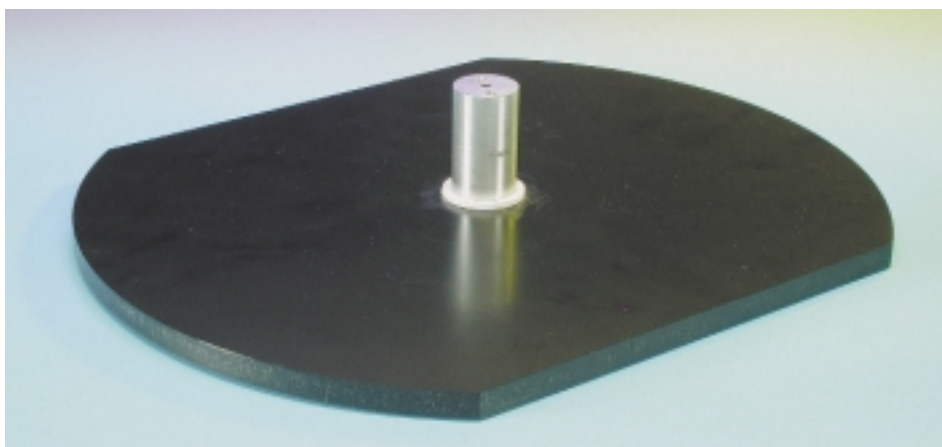
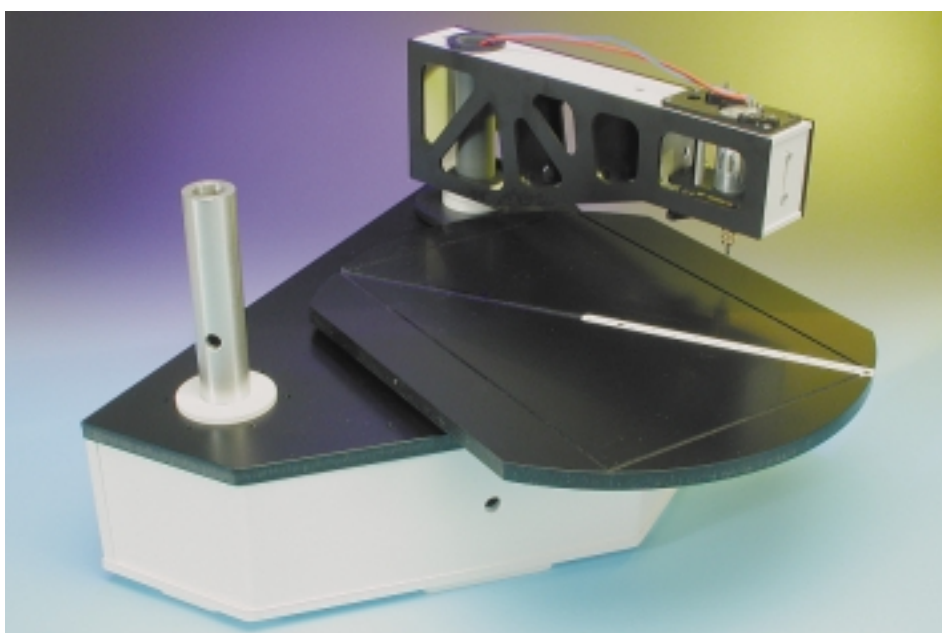


Figure 11. Turntable and axle.



istration guide, with 13 holes, is pressed down into the long slot in the table until the edge is flush with its surface.

The relative position of arm and turntable is not important as all the components are moved to the correct positions by software during commissioning of the machine: at this point the (still loosely-fitted) drill lift mechanism is also adjusted.

STEP 7: WIRING

Now we come to the point where the arm wiring must be installed. A two-conductor cable is soldered to each of the two limit switches, the drill motor and to the solenoid, and the wires are run back to the hole in the shaft. Feed the cables down the shaft until they appear in the machine enclosure directly next to the cable supports. The cables, already cut to length, are of course included in the kit of parts.

Now you can fit the connectors to the end of the cables (it is rather harder to feed the cables through the shaft with the connectors already attached!) There is no need to install the electronics in the enclosure yet: place the controller board to one side next to the enclosure while commissioning the machine. This makes things much easier if there is a fault in the circuit. When the electronics are proven to work, the board can be screwed into the machine enclosure. The PCB drilling machine is now finished and should look just like the final photograph.

(010024-4)

All that remains is to discuss the commissioning of the machine and to describe the software. We will conclude the series by describing the process for getting from PCB layout program to finished drilled circuit board.

Experiments with Piezo Ceramics (2)

applications of ceramics in electronics

By B. Kainka

Piezo ceramics are everywhere. Manufacturers have produced special ceramics optimised for different applications: one example is the thin disc-shaped piezo sound transducer used in the experiments described here.

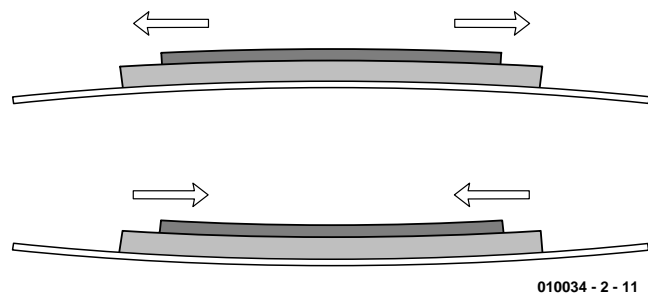
How do piezo transducers manage to make such a loud noise when the movement they make is so small? Here is the trick: the ceramic is always bonded to a thin sheet of metal. The ceramic disc stretches while the metal does not, and so a relatively large bending motion is produced (**Figure 1**). Also, the stiffness and mass of the materials are chosen so as to produce a resonance at a frequency of a few kilohertz.

Since ordinary piezo transducers (**Figure 2**) operate on the same principle as we have already explored in capacitors, we ought to be able to repeat all the experiments we described using the capacitor, and get much better results.

The transducer can be charged and then warmed with a finger. Connect a pair of headphones and you will hear a loud crackling noise. On cooling you can again hear the crackling, and again when warmed again — and so on. The experiment will now work with a pair of low-impedance (for example 32 Ohm) headphones.

A loudspeaker — and more

Using the piezo transducer as a loudspeaker is simple. If an audio frequency signal is connected, it is found that only a few millivolts are required to get an audible sound. If the diaphragm is made bigger (for example by attaching a piece of card), the sound is made louder. If you are near to a medium wave transmitter you can make a simple crystal set



010034 - 2 - 11

Figure 1. Principle of the ceramic transducer.

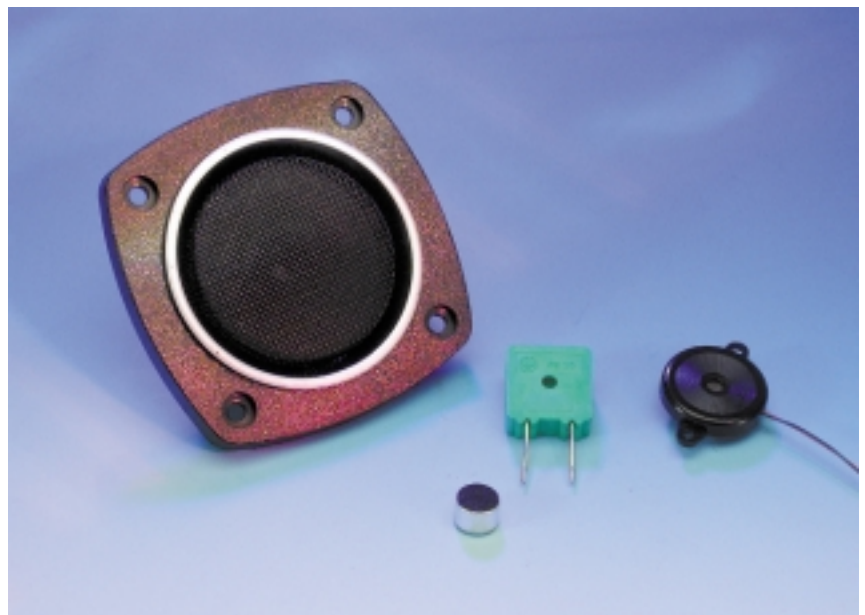


Figure 2. Practical implementation of the ceramic transducer.

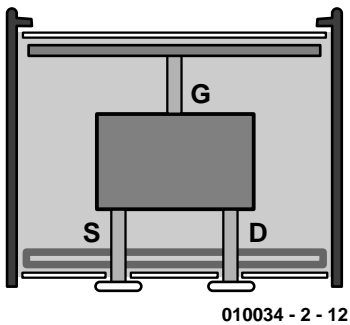


Figure 3. Construction of an electret condenser microphone.

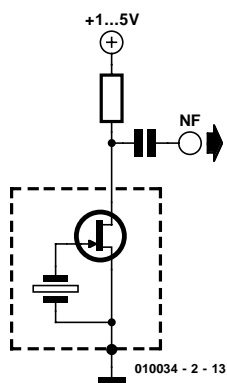


Figure 4. Connection to an external supply.

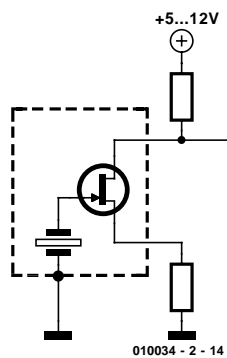


Figure 5. Connections to a passive infrared sensor.

using a piezo transducer for a loudspeaker.

As soon as the slice of ceramic is heated above the Curie temperature it loses its polarisation. It is necessary to be very careful when soldering it, to ensure that at worst only a small area is affected. If you are determined to destroy your piezo

transducer, simply dip it in hot water: the transducer will no longer work.

The transducer can be re-polarised by applying 300 V while warming it. It will work again, but not as well as before: in manufacture a much higher voltage is used.

Piezo ceramics are widely used in acoustic transducers, but they have very lots of other applications, including:

- Piezo ceramic igniters in cigarette lighters or gas lighters generate up to 10 kV.
- In glass delay lines tiny ceramic elements are used as piezo-acoustic transducers.
- Ceramic filters convert electrical impulses into mechanical ones, and vice versa.
- In video recorders ceramic elements are used for fine alignment of the head with the track on the tape.
- Ultrasound microphones use resonant piezo ceramics.

In principle, any piezo transducer can also be used as a microphone. They are used, for example, in recording sounds produced by equipment or objects. For normal use, however, the sharp resonance and low bandwidth make them unsuitable.

Electret and PIR

We are all familiar with electret condenser microphones with their particularly high sensitivity and frequency range. Here, a very thin sheet of polarised plastic is used, which forms one side of a charged capacitor. An integrated FET provides the required amplification (**Figure 3** and **Figure 4**).

The construction of passive infrared (PIR) sensors, as used in motion detectors, is very similar. A sliver of a suitable crystal (for example, lithium tantalate) is warmed by the incident infrared radiation and the voltage across it changes. Here too a FET is used as a preamplifier, and we have three connections, since both source and drain of the FET are brought out to pins (**Figure 5**).

If a microphone and a passive infrared sensor are so similar, it seems reasonable to suppose that they could be interchanged. The infrared sensor has a thick case and an optical filter, which will surely block out sound, so we must first open the case before we test its suitability as a microphone. The other way round, things are simpler. A simple ohmmeter suffices to determine whether an electret microphone makes a good heat sensor, and indeed a response to change in temperature can be detected. The effect, however, is only small, and so there is no danger of the weather affecting any audio recordings!

The experiments we have looked at so far have shown that there is a close relationship between ceramic capacitors, piezo transducers, condenser microphones and passive infrared sensors. Now a few typical applications will be described. Only piezo transducers and polarised ceramic capacitors will be employed. In general piezo transducers have a capacitance of around 50 nF and for many uses are much more sensitive than ordinary capacitors.

Measurement amplifier

A measurement amplifier for slowly-varying signals must have a long time constant. Here we are working with a 100 nF capacitor, and so we shall require a high input resistance. For a time constant of a few seconds we require an input resistor of around 10 M Ω . If we do away with the resistor and use an amplifier with a FET input, then the input voltage will vary uncontrollably. Resistors with values of many M Ω are not so common, and so instead we simply use a diode. When conducting, a diode is essentially an adjustable resistor with a very wide adjustment range. As the diode current is reduced, so the resistance increases.

Figure 6 shows the circuit of the amplifier. A type LM358 operational amplifier is run from a single power supply. It has PNP input stages and requires an input current of around 50 nA, which here flows through a type 1N4148 silicon diode. At this current the voltage across the diode is about 0.1 V, and so the DC resistance is about 2 M Ω . The differential input impedance is lower, however, at about 1M Ω . Thus, with a transducer with a capacitance of 50 nF, we obtain a time constant of around 50 ms. This means that the output voltage will be more affected by the rate of temperature change than by the temperature itself: almost exactly the behaviour of a PIR sensor.

The gain of the circuit is 28. The quiescent voltage at the output is about 3 V, and variations are observed around this average value.

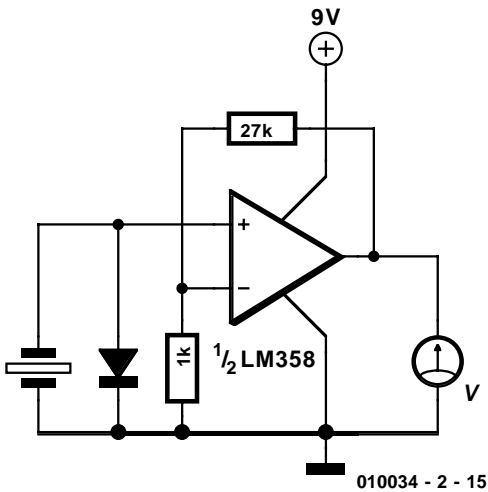


Figure 6. Sensor amplifier for temperature changes.

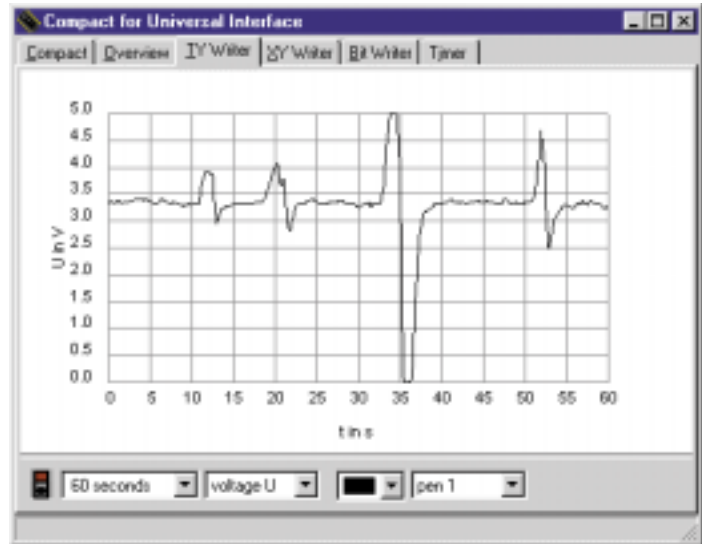


Figure 7. Temperature change readings on a PC.

These variations can be plotted using, for example, the 'Universal Interface for Windows' published in the December 1999 PC Topics Supplement of *Elektor Electronics*, along with its control software. **Figure 7** shows how the sensor responds to an approaching hand: the heat from the hand produces a clear signal.

Vibration sensor

A piezo transducer can be used as a vibration sensor. With the use of a suitable weight (say up to 100 g) it can be used to record vibrations in the ground, which are almost always present for one reason or another (**Figure 8**). Here we are interested in signals in the mid-

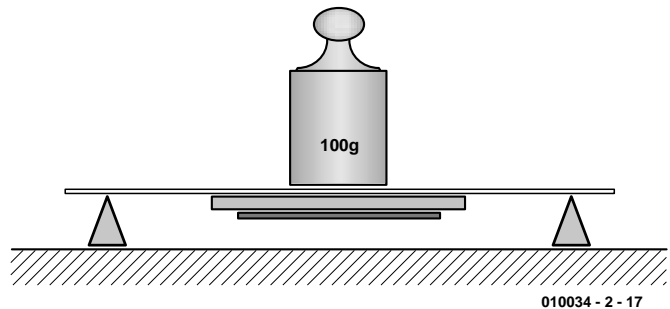


Figure 8. Use of a piezo transducer as a vibration sensor.

dle frequency range, but our amplifier has no lower cutoff frequency. By adding a capacitor to the negative feedback path as shown in **Figure 9** we can provide such a cutoff. Now we can set a desired centre voltage using a voltage divider. Using a TLC272 the circuit can be run from a single 5 V power supply.

Figure 10 shows the vibration of a tabletop lightly tapped by a finger. We have here the makings of a sensitive earthquake detector. A normal piezo transducer can serve as sensor; if we use polarised capacitors we can use rather greater weights.

ductivity (**Figure 11**). With this balance temperature sensitivity is a problem, since the sensor reacts to changes in both temperature and force. Poor construction can mean that the measurement results are affected by whether the item being weighed is hot or cold.

The amplifier must have a very high input impedance. A TLC272 CMOS operational amplifier fits the bill. In addition we need an input filter to block high frequency signals. In the circuit of **Figure 12** the charge produced is divided between the sensor and the 100 nF filter capacitor. The signal amplitude is therefore only about one third of what it would be without the filter, but this can be compensated for by adjusting the gain of the amplifier: we can in practice set any scale factor we want.

Figure 13 shows some measurements taken with our home-made balance on a time chart. Three dif-

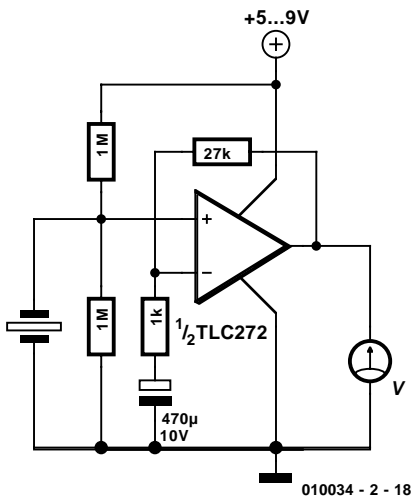


Figure 9. A measurement amplifier for the vibration sensor.

Ceramic scales

The arrangement for the force sensor is similar to that for the vibration sensor. A pan and some kind of fixing to the sensor are required, and both should, if possible, be made of a material with a low thermal con-

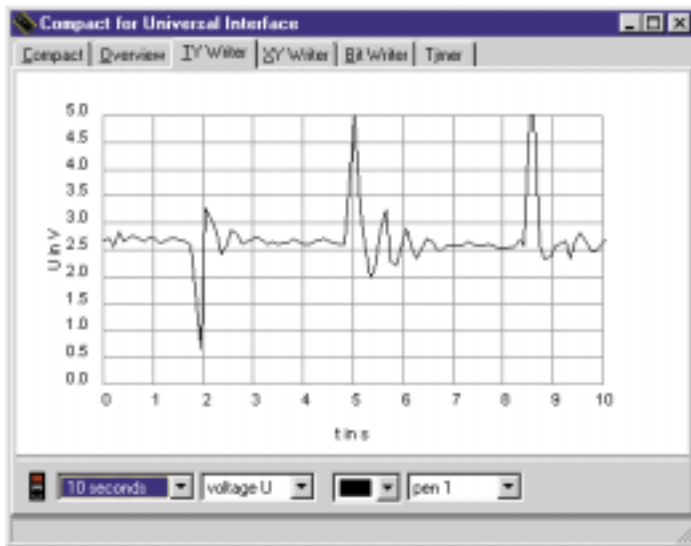


Figure 10. Measuring the vibration of a tabletop.

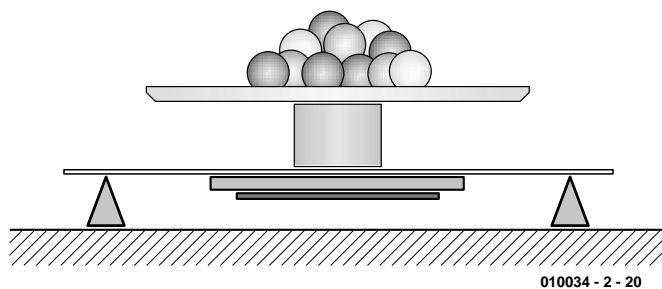


Figure 11. Experimental construction of an electric balance.

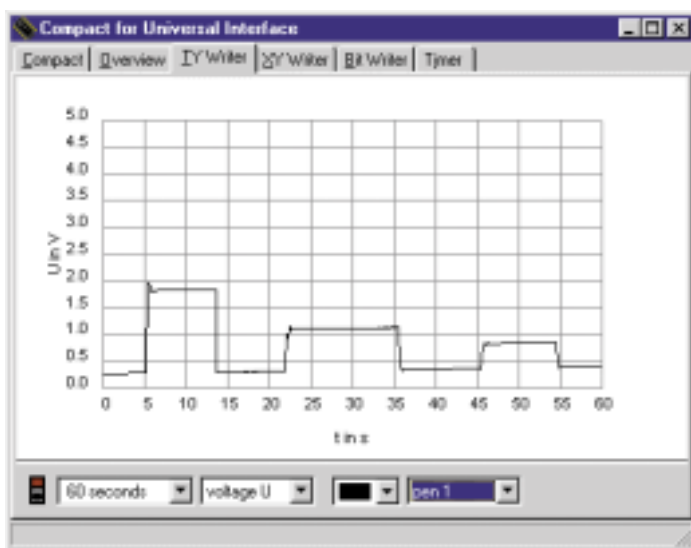


Figure 13. Static measurements with an electric balance.

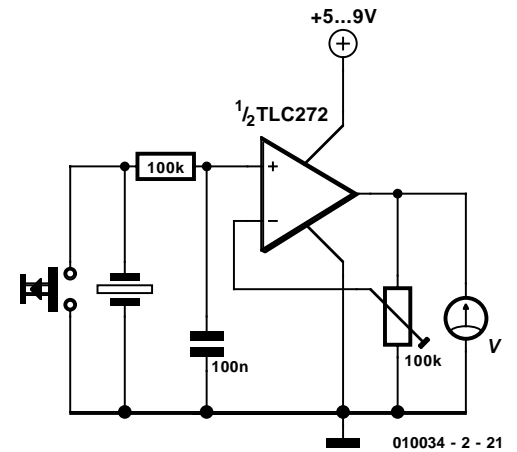


Figure 12. A measurement amplifier with a high input impedance.

ferent masses were placed on the balance and then removed. As can be seen, the measurement is basically static, but there is a discernible continuous slow rise in the offset voltage. This might be attributed to a temperature change or a leakage current in the operational amplifier. Nonetheless the balance is usable, as long as we adjust the offset voltage before each measurement. The circuit is provided with a discharge pushbutton for this purpose.

This arrangement of a piezo sensor as a balance is naturally rather experimental, and we cannot expect to achieve great accuracy. However, we can for example build a simple pair of postal scales with very little effort. The piezo transducer is suitable for masses up to around 100 g; with polarised capacitors we can weigh rather greater masses, up to several kilograms. One can imagine, for example, an arrangement where three tubular capacitors support a surface bearing the load to be weighed.

Once one has begun to try so many experiments, the possibilities start to seem endless. It is interesting, for example, to consider using a transducer in a liquid. Using a waterproof watch it has been observed that the alarm signal can be heard from one end of a swimming pool to the other, a distance of 25 m. Hence the energy transfer from the transducer into the liquid is efficient. And so we are immediately led to many new applications: what about an echo sounder? Or an ultrasound cleaning device? Or how about eavesdropping on your goldfish?

(010034-2)

In-Car SMPSU

for the OSCAR MP3-CD-player

Design by T. Giesberts

The OSCAR Stand-Alone MP3 Player described in last year's September issue appears to be very popular with our readers. So popular in fact, that a large number would also like to use it in their car, boat or caravan. Using the DC/DC-converter described here will make that easier, from the power supply perspective at least.



We really have ourselves to blame for this. In the first instalment of the series of articles describing the OSCAR, we carelessly let it slip that it would be advisable to use a special DC/DC converter to power this MP3-CD-player for mobile use. As a result, one can promptly expect the question 'when are you publishing such a converter?' Consequently, we immediately got to work and developed one. The end result is described below. It turns out that the required electronics is actually quite reasonable.

Two voltages

The OSCAR requires two (stabilised) power supply voltages: +5V/2A and +12V/2A. The exact current consumption will depend on the actual CD-ROM drive and hard disk used.

To derive these voltages from a 12 V battery appears to be a 'piece of cake', in the first instance. Anyone with even a little experience in elec-

tronics will immediately suggest a 'big' three-legged regulator for the 5 V supply. It would also appear reasonable to simply obtain the 12 V supply by direct connection to the battery.

However, it isn't quite so easy. An 'ordinary' 5 V voltage regulator would have to dissipate a lot of power and, concerning the 12 V supply, we have to remember that the battery voltage in a car varies signif-

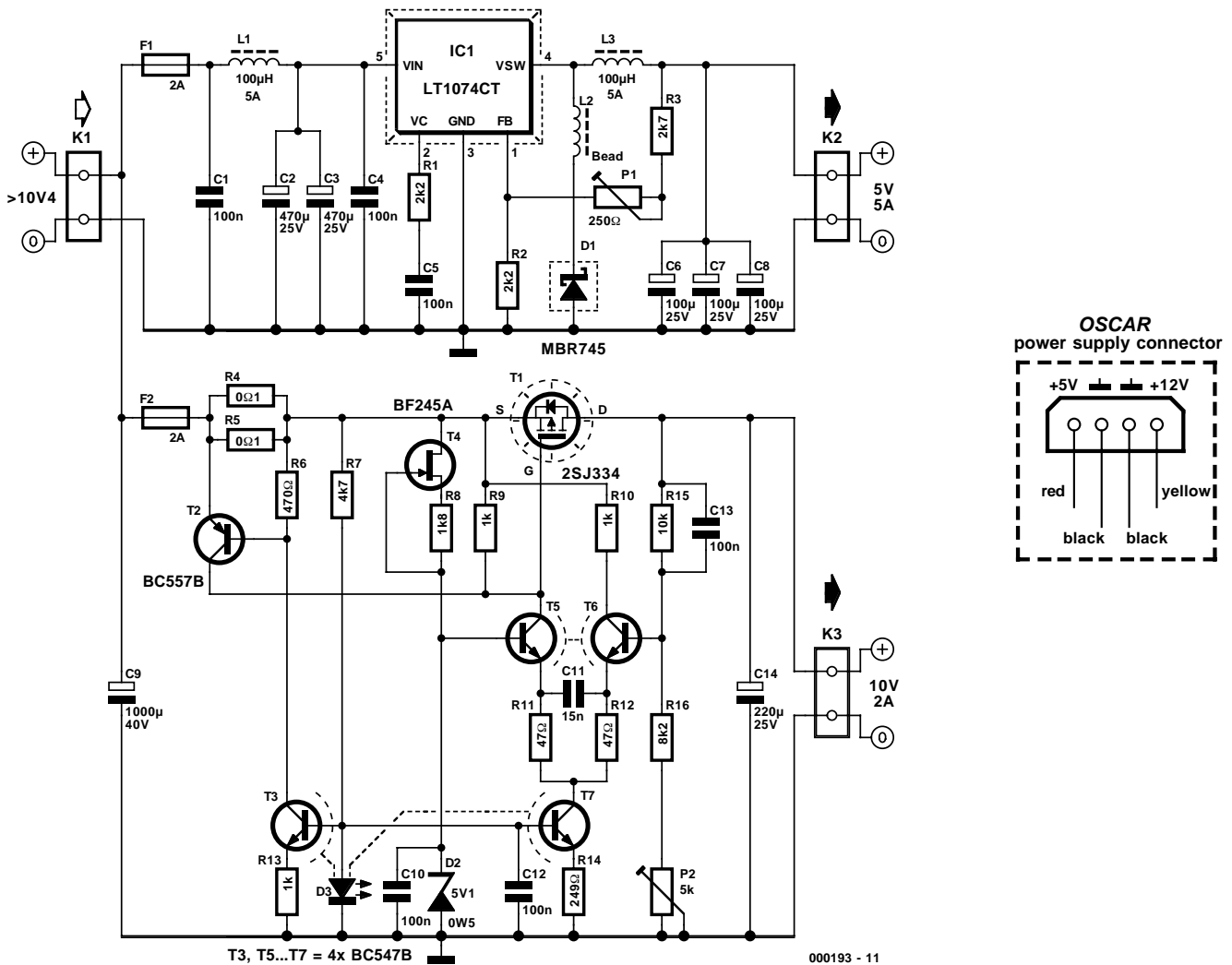


Figure 1. The schematic of the automotive power supply. P1 and P2 allow accurate adjustment of the output voltages.

icantly during use. Depending on the circumstances, the voltage can swing between about 11 V and 14.4 V. In some cases this maximum may be even higher!

The problems concerning the 5 V supply are solved by avoiding a linear regulator and using a switching step-down (buck) converter instead. Even when the battery voltage is as high as 14.4 V the dissipation will be manageable.

For the 12 V power supply a low-dropout voltage regulator will suffice. This voltage is used only for the running mechanisms and may be a lot smaller in practice. The OSCAR continued to function without any problems when the power supply was lowered to 8 V. We settled on a regulated voltage of 10 V. This is easily obtained from the battery voltage with relatively simple means. The

dissipation is low enough so that there is no need to resort to a switching regulator.

5 V/5 A

To generate the 5 V power supply we applied the LT1074CT, which has featured in *Elektor Electronics* on earlier occasions. This is an integrated switching regulator that can deliver a generous 5 A and requires very few external components. While configured as a classic ‘chopper’, a number of refinements to this IC make it possible for it to be used as a positive-to-negative converter, negative ‘boost-converter’ and negative flyback-converter. Since the LT1074CT is provided with an analogue multiplier in the negative feedback loop, it reacts to input voltage variations almost without delay,

while at the same time, the loop gain is independent of the input voltage. Furthermore, the LT1074 possesses a robust current limiting circuit, which makes the IC nearly insensitive to overloads and short circuits. The input voltage range during normal use stretches from 8 V to 60 V.

A glance at the schematic depicted in **Figure 1** reveals at the top the 5 V regulator, built around the LT1074CT (IC1). As can be seen, there are indeed very few external components. We will now examine the schematic a little closer.

The input voltage is decoupled with L1 and C2, C3 and C4 for the purpose of filtering the switching pulses. F1 prevents a short circuit on the battery in the event of a fault.

R3/P1/R2 define the output voltage. P1 permits the output voltage to be adjusted to exactly 5 V. The voltage divider is intentionally connected behind L3 in order to automatically compensate for the voltage drop across the noise suppression choke.

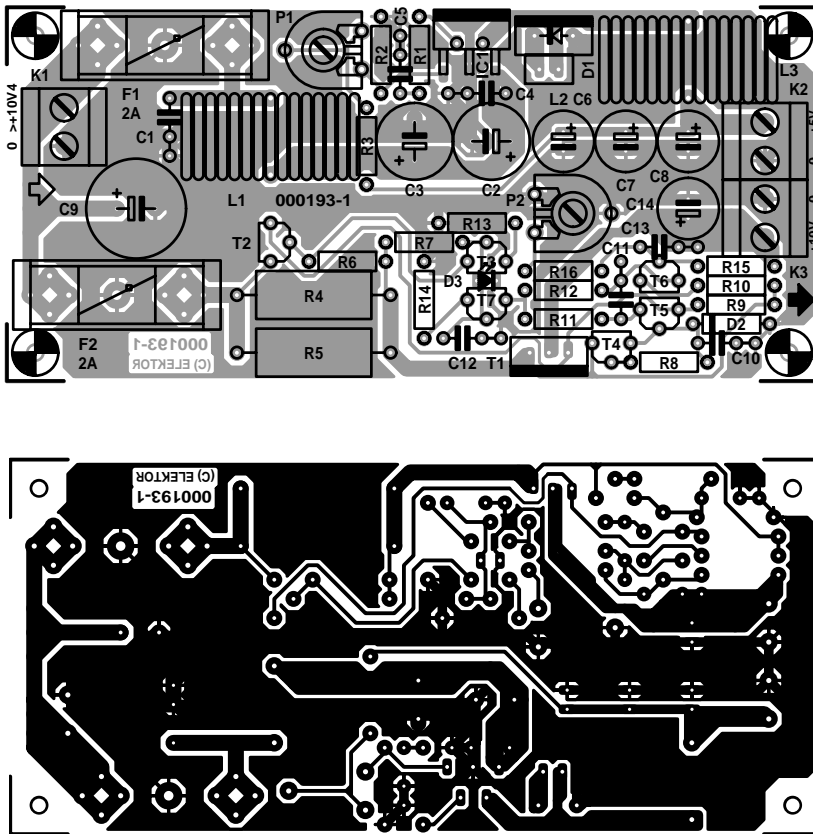


Figure 2. The printed circuit board is compact and easy to construct. Components IC1, D1 and T1 have to be cooled.

The network R1/C5 is required to frequency compensate the IC. Ferrite bead L2 may optionally be inserted to reduce possible RF emission caused by the switching of D1. This bead is not strictly necessary and may be replaced with a wire link.

L1 and L3 are two standard 100 μ H/5 A noise suppression chokes. These are easily obtained and have the advantage that they may be used with both the LT1074CT and the 2 A regulator LT1076CT. The latter is pin compatible and may be used instead if the current under consideration does not exceed 2 A.

The parallel connection of electrolytic capacitors C2/C3 and C6/C7/C8 may look a little strange initially, but has the advantage that the size of the current pulses per capacitor are reduced (longer life!) and that the total ESR and parasitic inductance are significantly smaller (smaller spikes).

Both the 10 V and the 5 V power supplies are protected with 2 A fuses against sustained overloads. If it is really necessary to obtain a current of 5 A from the 5 V power supply then the fuse needs to be increased (to 2.5 A at 12 V in), but because the load from the OSCAR is much smaller, a 2 A fuse is perfectly adequate.

10 V / 2 A

For the 10V/2A regulator a discrete solution was selected, so that it could easily be tailored to the application. Conceptually it is a low-drop regulator around a difference amplifier with current limiting to make the regulator short-circuit proof.

The voltage reference is a simple JFET current source (T1) and a 5.1 V zener diode (D2). The base of T5 forms the non-inverting input of a difference amplifier. The output voltage is fed back via voltage divider R15/R16/P2/C13 to the inverting input. P2 sets the desired output voltage, and this may be a little more or a little less than 10 V.

Considering the application, the internal gain (mainly defined by R9/R11/R12) and as a consequence the ripple rejection has been kept modest. In this configuration the output voltage at 2 A drops about 0.4 V, which is completely acceptable in this case.

The final stage consists of a sin-

COMPONENTS LIST

Resistors:
 R1,R2 = 2k Ω
 R3 = 2k Ω
 R4,R5 = 0 Ω 5W
 R6 = 470 Ω
 R7 = 4k Ω
 R8 = 1k Ω
 R9,R10,R13 = 1k Ω
 R11,R12 = 47 Ω
 R14 = 249 Ω
 R15 = 10k Ω
 R16 = 8k Ω
 P1 = 250 Ω preset H
 P2 = 5k Ω preset H

Capacitors:
 C1,C4,C5,C10,C12,C13 = 100nF ceramic
 C2,C3 = 470 μ F 25V radial
 C6,C7,C8 = 100 μ F 25V radial
 C9 = 1000 μ F 40V radial
 C11 = 15nF
 C14 = 220 μ F 25V radial

Inductors:
 L1,L3 = 100 μ H 5A, e.g., SFT12-50 (TDK)
 L2 = ferrite bead (see text)

Semiconductors:
 D1 = MBR745
 D2 = zener diode 5.1V, 0.5W
 D3 = LED, flat, red
 T1 = 2SJ334 (Toshiba)
 T2 = BC557B
 T3,T5,T6,T7 = BC547B
 T4 = BF245A
 IC1 = LT1074CT (Linear Technology)

Miscellaneous:
 K1,K2,K3 = 2-way PCB terminal block, lead pitch 5mm
 F1,F2 = fuse, 2A, with holder and cap for PCB mounting
 PCB, order code 000193-1 (see Readers Services page)

gle P-channel power MOSFET type 2SJ334. Because the drain functions as the output, the drop-out voltage is defined by the channel resistance of the MOSFET, the shunt for the current limiting (R4/R5) and the fuse. The minimum resistance of the transistor plus the shunt is less than 110 m Ω .

The MOSFET is a logic-level type that starts to conduct at 2 V and is practically fully on at 4 V. The usual application for the 2SJ334 is actually

as a switch, but this transistor also functions very well in the linear range. V_{DSS} may be up to 60 V and the drain current may be as high as 30 A DC. The SOA (safe operating area) of the 2SJ334 permits a maximum current of nearly 3 A during a short circuit with an input voltage of 14.4 V; this is the design value of the current limiting circuit.

At increasing temperatures, T2 will limit earlier and earlier, which is a desirable additional safety feature, considering the typical environment in which the power supply is intended to operate (a hot car in the summer sun).

Current limiting takes place because the collector of T2 lowers the V_{GS} of T1. The drive current for the gate is a maximum of 4 mA through R9 (defined by current source T7). By biasing the base of T2 (voltage across R6) the voltage required to turn T2 on can be significantly reduced. This has the desir-

able consequence that the drop-out voltage of the entire regulator is reduced because the resistance of the shunt R4/R5 can be made smaller.

The bias of T2 is provided by current source T3, which uses the same reference LED (D3) as current source T7 for the difference amplifier. The current for the LED is via R7 directly obtained from the input voltage. On the PCB, T3 and T7 are placed on opposite sides of D3 so that they will track each other's temperature and minimise the thermal drift of the power supply. Thermal stability was also considered while designing FET-based current-source T4: the selected current of 0.6 mA is the value at which the temperature coefficient of a BF245A is minimised.

C11 and C13 are added to improve the pulse response of the voltage regulator. C10/C12 and C14 are additional decoupling. R10 is included in the collector leg of T6 in order to

make good control of the adjustment of the difference amplifier possible.

Printed Circuit Board

A circuit board has (naturally) been designed for the In-Car Power Supply onto which the entire circuit including fuses and suppression chokes are fitted in a compact manner. **Figure 2** shows the layout of this PCB. There are almost no peculiarities regarding the assembly of the PCB. We've already mentioned the thermal coupling of T3/D3/T7. The same is true for the difference amplifier T5/T6. Good thermal coupling will improve the stability and it is therefore advisable to clamp both transistors together with, for example, a cable tie.

As can be seen, the input and outputs are along the short edges on opposite sides of the PCB. The connections for the OSCAR are in the same order as those of a PC power supply connector: +5V/0/0/+10V. For clarity, this connector is also shown in the schematic of **Figure 1**.

It is important that IC1, D1 and T1 are provided with sufficient cooling. For the prototype we used a 'box' with a do-it-yourself heatsink made from a 3 mm thick sheet of aluminium. This sheet was bent into a U-channel with the same inner dimensions as the length and width of the PCB and a height (approximately 35 mm) such that the sides protrude just above the components when the PCB is fitted inside (on plastic stand-offs). This not only ensures excellent cooling but also creates a very robust power supply module that can withstand a little abuse expected in automotive applications. **Figure 3** shows the appearance of the finished circuit board and heatsink. On the other hand, if you have no desire to bend your own U-channel, a nearly identical result can also be obtained by using two pieces of L-shaped extrusion.

The case of the LT1074CT (IC1) is used to connect the aluminium sheet to ground, by not isolating it. The case of the 2SJ334 is made from plastic and does not need additional isolation. Diode D1 also needs to be cooled and has to be electrically isolated because the cathode is internally connected to the case. This can be done using a mica washer. Use thermal paste when mounting IC1, D1 and T2 to the heatsink (apply thinly!).

A standard Y-adapter for PC power supplies can be used for the connection between the power supply, the CDROM drive and the main circuit board of the OSCAR: one male/large to two female/large. The connector at the male end can be disassembled. The pins can be pulled from the housing by pushing on the locking tabs.

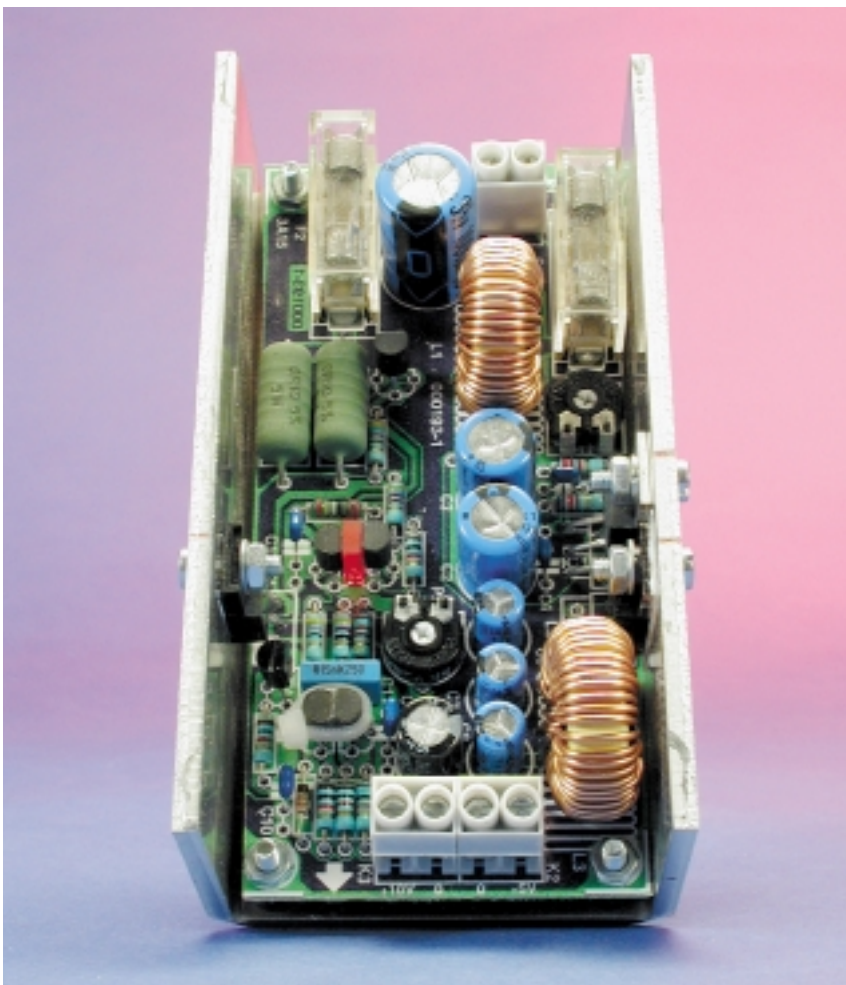


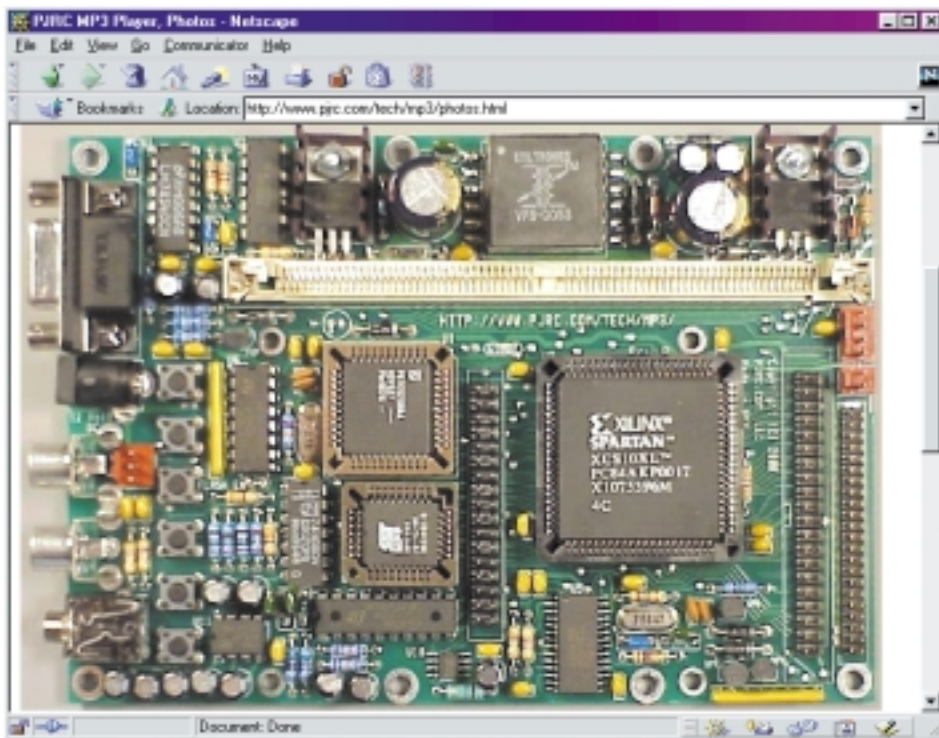
Figure 3. The do-it-yourself aluminium channel affords both cooling and protection.

(000193-1)

More MP3 Projects

DIY portables

A few months ago under this heading, we presented a number of Internet addresses for MP3 projects. Most of them are based on re-using hardware from an old PC. However, there are also other possibilities, including projects for building your own portable MP3 player.



A design based on a PC motherboard, a hard disk or CD-ROM drive, a few pushbuttons, an LCD module and some special software is not such a bad approach to making your own MP3 player, but the result can hardly be said to be compact. Nowadays, the subject of MP3 has become so popular that an increasing number of electronics engineers and hobbyists have given their attention to it, with the result that you can now find complete web-sites that are dedicated to the construction of truly portable (as opposed to 'moveable') MP3 players. A (mini) hard disk, CD-ROM drive or flash memory card is used as the storage medium.

The MP3Projects site [1] is a good starting point for searching for such projects. This site gives an extensive and up-to-date overview of DIY MP3 projects on the World Wide Web. Starting from this site, we visited a number of MP3 addresses related small portable players. This time, we have intentionally omitted the larger types of units (those using PC motherboards).

The MP3 PubliC from **MP3ar** [2] is a very nice example of an MP3 player. It involves a small printed circuit board that is slightly larger than a

computer mouse. An LCD can be stuck on top of the circuit board, and the newest version allows the connection of a CD-ROM drive as well as a hard disk drive. An accompanying PC program enables the user to download files from the PC to the hard disk. All circuit board layouts, software and source code for the project are available free of charge. This project has been created by three Argentinean electronics engineers. The estimated cost is around 130 dollars.

The **High Capacity MP3 player** van **MPRJ** [3] is a design that contains more ICs, but it certainly still deserves the name 'portable'. This player uses a normal (mini) hard disk drive, which provides enough capacity for hundreds of hours of music. Thanks to a 24-bit DAC, this unit delivers outstanding sound quality. The player can handle various types of MP3 compression, so the user can decide what level of sound quality he or she wants. MPRJ can even provide all of the components for this project. You can buy a fully assembled and tested circuit board (\$150), a bare circuit board or the individual components. The source code for the firmware is available for anyone who wants to make a few changes to how the player works.

The design of the **MoP3-player** from Frank Hickl & Mario Becker [4] is still under development. This system is built around an inexpensive MC68HC11 embedded controller. As with many other players, MP3



decoding is handled by an MAS3507D or STA013 IC. A fairly large LCD panel (128 × 64 pixels) simplifies communications with this device. The photos show a relatively small printed circuit board, so we can rightly consider this to be a portable player. This unit currently works with a CD-ROM drive, but it is the intention that it will shortly be

able to work with a hard disk drive as well.

The YAMPP site [5] is another location where a lot of work is going on. The name stands for 'Yet Another MP3 Player'. Here you will find several different hardware designs, some of which are still under development. All of the players have connectors for an IDE hard disk drive

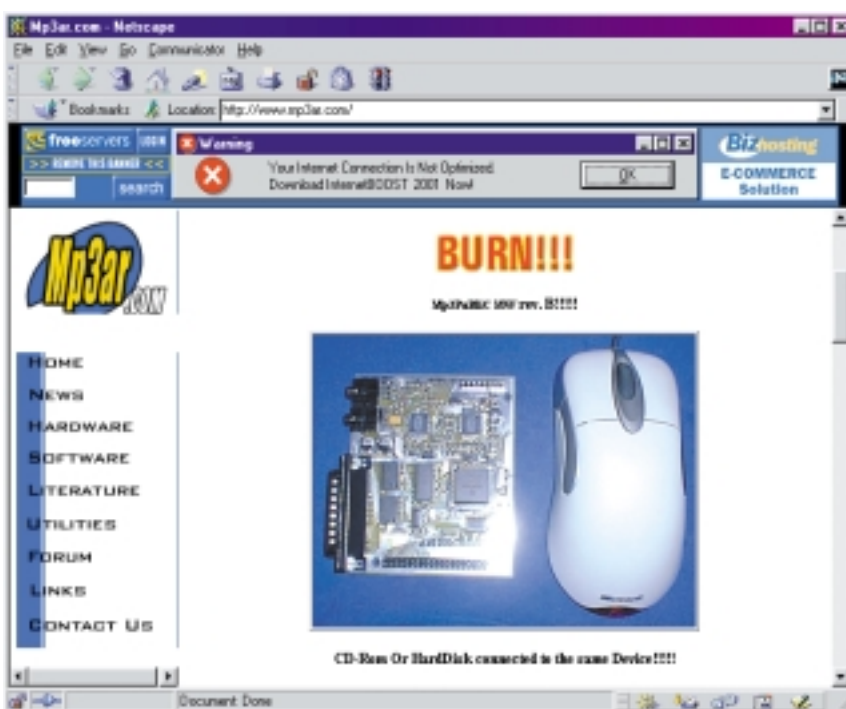
and a display. It is interesting to spend a bit of time studying and comparing the designs, since different approaches have been taken based on various processors. Here again, as with most player designs, all hardware and software information is freely available.

Audio Affinity 2000 [6], which comes from a number of electronics types in Florida, is a special project that is just getting started. The unusual feature of this project is that it supports both recording and playback in the MP3 format. Up to now, there isn't much in the way of concrete items to be seen on this site, but you might want to take a look at it from time to time in the future.

The **ELM MP3 Player** [7] is a design that looks like a commercial product, at least to judge by the prototype made by its Japanese designer. This small player uses Smartmedia memory cards to store the music data. The circuit is built around an Atmel controller and a Micronas DSP/DAC chip set.

Finally, we looked at a player that evidently comes from the final term project of a pair of students (Peter D'Antonio and Daniel Riiff). Their design, which goes by the name '**EE 476, Final Project: Portable MP3 Player**' [8], is still in the prototype phase, but its overall capabilities are quite attractive. If the two gentlemen succeed in reducing all of this to a printed circuit board of acceptable dimensions, it will certainly be an interesting player. The special feature of this player is that it uses flash memory cards.

(015065-1)



Website url's

- [1] MP3Projects:
<http://www.mp3projects.com/>
- [2] MP3 PuBliC:
<http://www.mp3ar.com/>
- [3] MPRI:
<http://www.pjrc.com/tech/mp3/>
- [4] MoP3:
<http://www.geocities.com/SiliconValley/Circuit/3150/index.html>
- [5] YAMPP:
<http://www.myplace.nu/mp3/>
- [6] Audio Affinity 2000:
<http://www.audioaffinity.com/>
- [7] ELM MP3 Player:
http://elm-chan.org/reports/mpc/report_e.html
- [8] Portable MP3 Player:
<http://instruct1.cit.cornell.edu/courses/ee476/FinalProjects/s2000/peterdan/final.htm>

32-Channel Analogue Input Card

for the serial (RS232) port

By George Vastianos

Student of Electronics Department, Technological Educational Institute of Piraeus, Greece

This project is a card with 32 analogue inputs with 12-bit resolution each for external connection to the serial port. The feature *par excellence* of the present project is the way it communicates with the computer. Read on!



Most commercially available cards for analogue voltage capturing include a microcontroller or a UART chip for the serial communication. The card described in this article does not have any of the above chips because its design is based on direct accessing of PC internal UART registers which arrange the conversion from serial to parallel.

About the serial port

Serial ports are mainly used for communication between computers or between computers and peripherals such as a modem or a mouse. The controller of the PC internal RS232 port is nearly always a UART (Universal Asynchronous Receiver Transmitter) chip fitted on the mother-

board. The chip works as a serial-to-parallel and parallel-to-serial adapter.

A computer may have one to four serial ports (COM1 through COM4). As shown in **Table 1**, each port occupies eight locations in the I/O memory map.

The basic lines used by a UART for all serial communication (transmission and reception), are called TXD and RXD. Also a group of extra lines (DCD, DSR, RTS, CTS, DTR, RI) is used to establish different types of serial communication. Some of these extra lines work as inputs, others as outputs. Each of these (except RXD) may be controlled through a bit of a UART register, see **Table 2**.

The voltage levels of the serial (RS232) port are different from TTL levels. So, for logic '1' the voltage is -12 V and for logic '0' the voltage is $+12\text{ V}$ (nominal values).

The hardware

With reference to the block diagram of the circuit shown in **Figure 1**, the inputs have been divided into four groups of eight inputs (00-07, 08-15, 16-23, 24-31) which are taken to the A-to-D Converter (ADC) units. The four ADC output lines are passed to

Technical Specifications

Number of inputs: 32
 Type of inputs: Analogue (resolution 12 bits, $V_{in(min)}=0$ Vdc, $V_{in(max)}=5$ Vdc)
 Connection: Serial Port (RS232)
 Communication: direct access of PC internal UART registers
 Power Supply: 9-15V DC
 Supply requirement: approx. 30 mA (stand-by)

Table 3. Channel allocation on Connectors K2/K3.

K3		K2	
Channel	Pin Number	Channel	Pin Number
00	33	16	33
01	31	17	31
02	29	18	29
03	27	19	27
04	25	20	25
05	23	21	23
06	21	22	21
07	19	23	19
08	15	24	15
09	13	25	13
10	11	26	11
11	9	27	9
12	7	28	7
13	5	29	5
14	3	30	3
15	1	31	1

Table 1. UART Internal I/O Addresses

Register Name	COM1	COM2	COM3	COM4
Transmit/Receive Buffer	3F8h	2F8h	3E8h	2E8h
Interrupt Enable Register	3F9h	2F9h	3E9h	2E9h
Interrupt Identification Register	3FAh	2FAh	3EAh	2EAh
Line Control Register	3FBh	2FBh	3EBh	2EBh
Modem Control Register	3FCh	2FCh	3ECh	2ECh
Line Status Register	3FDh	2FDh	3EDh	2EDh
Modem Status register	3FEh	2FEh	3EEh	2EEh
Scratch Pad Register	3FFh	2FFh	3EFh	2EFh

Table 2. RS232 Data/Control Line Addresses (I/O Map)

Pin Name	25-pin connector	9-pin connector	COM1	COM2	COM3	COM4	Bit	I/O
TXD	pin 2	pin 3	3FBh	2FBh	3EBh	2EBh	6	O
RXD	pin 3	pin 2	3FCh	2FCh	3ECh	2ECh	0	I
RTS	pin 4	pin 7	3FDh	2FCh	3ECh	2ECh	1	O
CTS	pin 5	pin 8	3FEh	2FEh	3EEh	2EEh	4	I
DSR	pin 6	pin 6	3FEh	2FEh	3EEh	2EEh	5	I
RI	pin 22	pin 9	3FEh	2FEh	3EEh	2EEh	6	I
DCD	pin 8	pin 1	3FEh	2FEh	3EEh	2EEh	7	I

the COM Driver/Receiver Unit which handles the conversion from TTL level to RS232 level. Four inputs are used on the serial port: CTS, DSR, RI, and DCD. Three outputs of the serial port (TXD, DTR, RTS) are used for the selection of the inputs we want to read (via the ADC's communication protocol). After conversion from RS232 to TTL, the group select signals are passed to the ADC select inputs.

Practical circuit

In the schematic circuit of ADC card, Figure 2, the 32 inputs have been distributed across connectors K3 and K2. The correspondence between the inputs and the connectors can be seen in **Table 3**.

All connector inputs are taken directly to the inputs of the MAX186ADCs (IC1-IC4) without pull-up or pull-down resistors for termination. This allows you, the user, to choose the best type of termination, according to the application. Each of the capacitor pairs C1-C6, C2-C7, C3-C9 and C10-C8 consists of a ceramic (or MKT) capacitor of 100 nF an 4.7-µF electrolytic type in parallel. The pairs decouple the supply voltage, suppress high- and low-frequency noise and so improve the stability of the circuit.

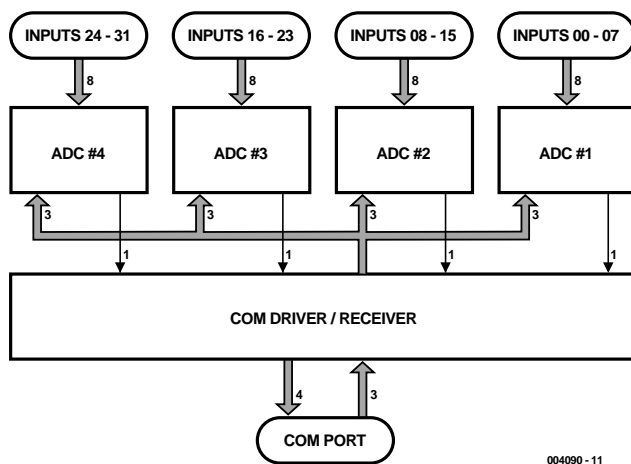


Figure 1. Block diagram of the 32-channel Analogue Input Card for the RS232 Port.

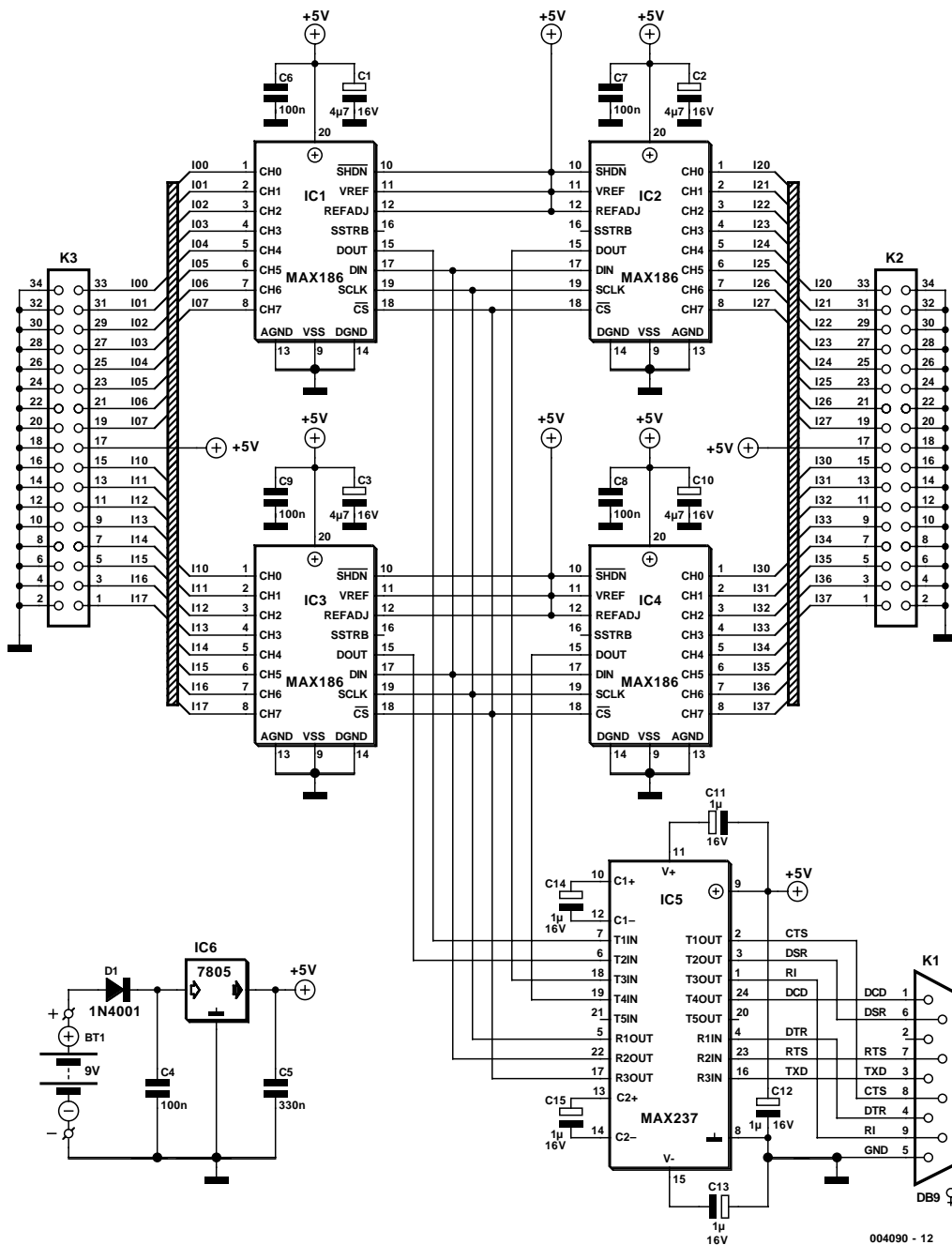


Figure 2. Practical circuit of the 32-channel ADC card.

The last unit is the COM Driver/Receiver Unit, which includes its own regulator so that the card does not need a regulated power supply to work. In the schematic diagram, the Max237 (IC5) works as an RS232 driver/receiver. It converts three channels from RS232 to TTL, and five channels from TTL to RS232. The five capacitors C11 through C15 (1 μ F) are used by the MAX237 for voltage doubling. This is necessary to allow the driver section to produce a voltage of 10 V (from the 5 V supply) for the five RS232 outputs. The MAX237 contains an

inverter in each channel. Circuit IC6, the ubiquitous 7805, together with satellite capacitors C4 and C5, steps the raw supply voltage down to a regulated line of 5 V. Diode D1 protects the circuit against input supply voltage reversal. The circuit is suitable for powering by a 9-volt battery.

Construction

The circuit is best built on the

printed circuit board with the layout shown in **Figure 3**. This single-sided PCB is available ready-made through the Publishers' Readers Services under order code **004090-1**.

Construction of the board should not present problems if you work carefully using the component overlay and the information in the parts list. There are 12 wire links on the board. Despite the fact that wire links are the simple but essential parts, we know from experience that they are often forgotten, and many reported problems with our construction projects can be traced down to absent wire links. So fit the 12 links before any other components.

The 7805 regulator is only lightly loaded by the circuit and can make do without a heatsink.

Pay attention to the orientation of polarized parts (diode, electrolytic capacitors).

We recommend using sockets for all ICs (except of course the 7805).

The software

The software for the communication with the card has been developed in QBASIC. The communication routine is called CARD32AI — its source code is available as a free download.

This routine may be called from every program written in QBASIC if you stick to the following syntax:

```
CALL CARD32AI (COMADDRESS, CHANNEL ())
```

Where COMADDRESS is the Integer type variable containing (before the call) the base address of the serial port. Probable values of this variable are &H3F8 for COM1, &H2F8 for COM2, &H3E8 for COM3 and &H2E8 for COM4.

CHANNEL () is the matrix integer type variable (with pointers from 0 to 31), which (after the call) contains the arithmetic value of each separate channel with values 0 to 4095.

A demonstration program has been

COMPONENTS LIST

Capacitors:

C1,C2,C3,C10 = 4 μ F 16V radial
 C4,C6-C9 = 100nF
 C5 = 330nF
 C11-C15 = 1 μ F 16V radial

Semiconductors:

D1 = 1N4001
 IC1-IC4 = MAX186-CPP
 IC5 = MAX237CNG
 IC6 = 7805

Miscellaneous:

K1 = 9-way sub-D socket (female), angled pins, PCB mount
 K2,K3 = 34-way boxheader PCB, order code 004090-1 (see Readers Services page and Elektor Electronics website)
 Project software: free download from Publishers' or author's website.

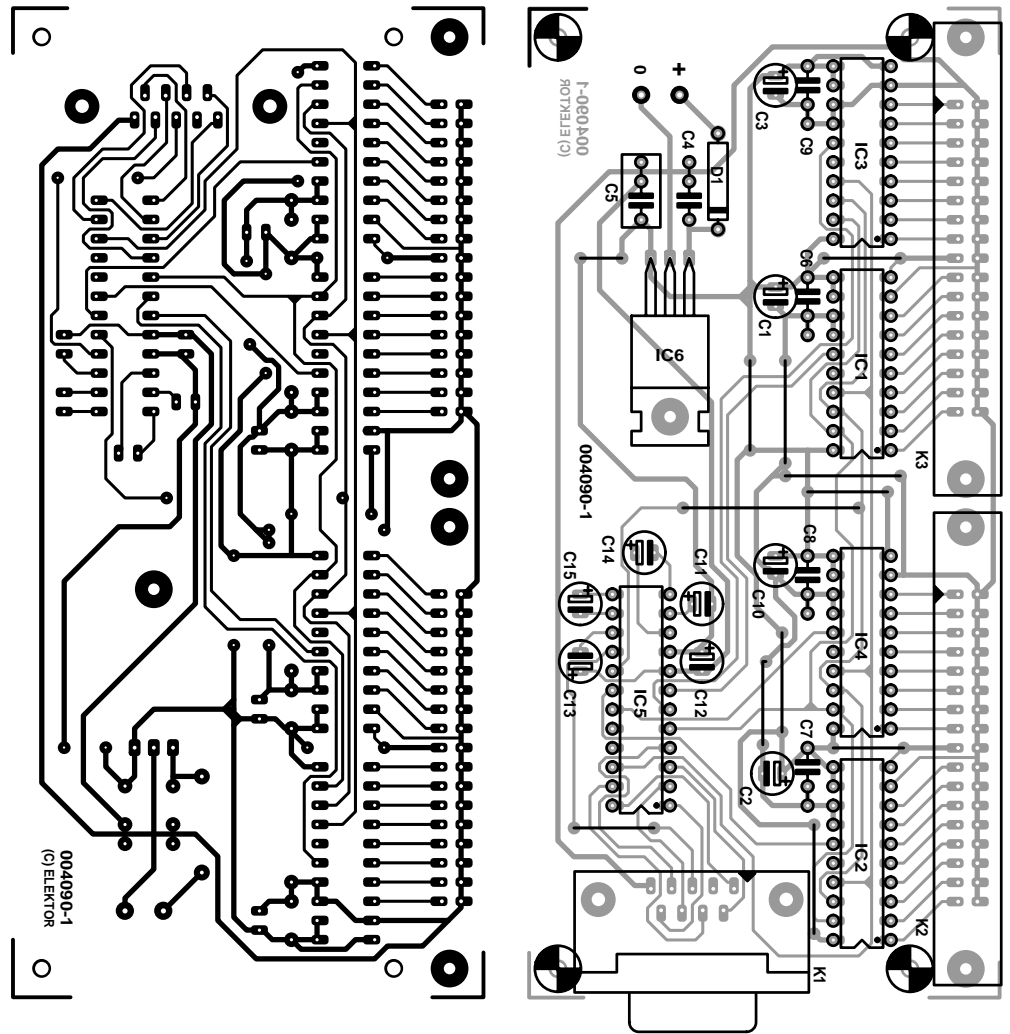


Figure 3. PCB artwork (board available ready-made).

developed for testing the card. A screenshot of the program in action is shown in **Figure 4**.

To change the serial port address, simply use the keys <1> to <4>. If you want to quit, just press <Esc>.

Free download

The source code of the communication routine (CARD32AI.SUB) and the demonstration program (32AICARD.BAS), with an executable version of the demonstration program (32AICARD.EXE) have been packed into a single zip file which may be downloaded from the author's website at

<http://www.robofreak.xs3.com>

or from the Free Downloads section (June 2001 items) of the Publishers' website at

<http://www.elektor-electronics.co.uk>

At the express request of the author, no floppy disks will be sold containing the project software. Readers interested in obtaining the software but not having Internet access are requested to contact the Editor in writing.

(004090-1)

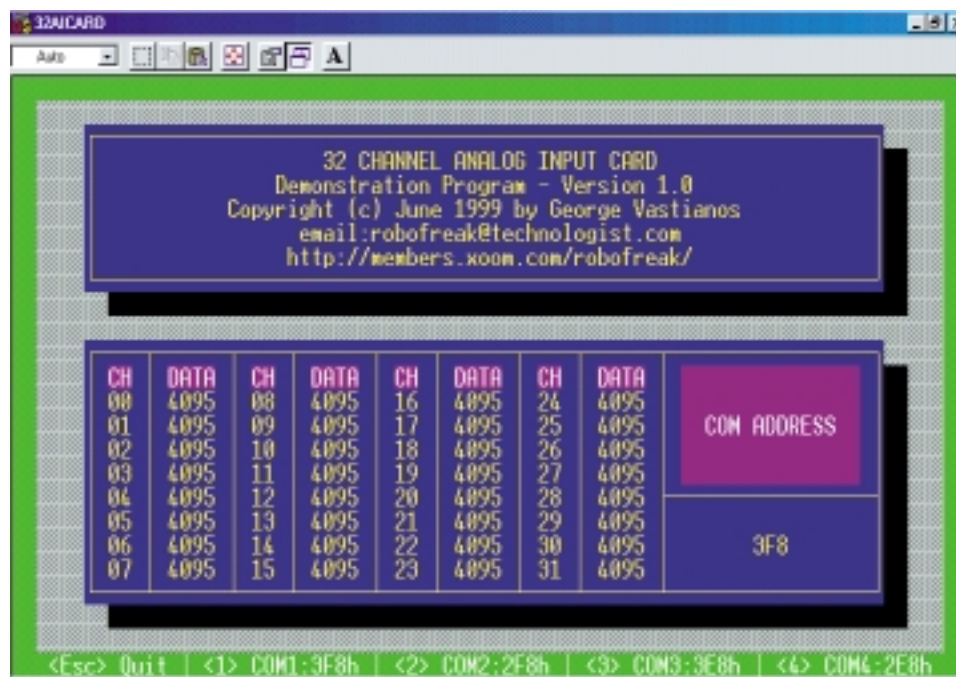


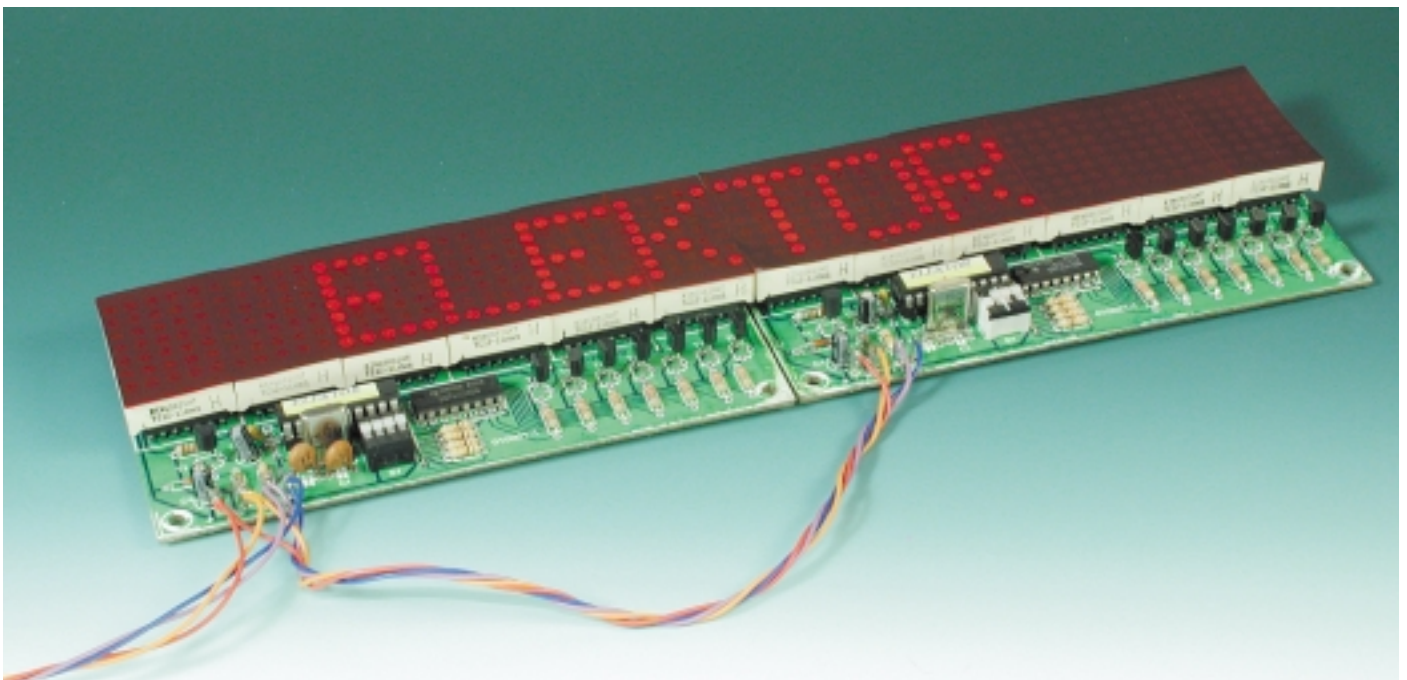
Figure 4. The demo program in action.

Modular Dot Matrix Display

PC interface to the COM port

Design by A. Köhler

Moving signs are a common enough sight today but off-the-shelf units are invariably expensive and somewhat inflexible. The six-character module described in this article can be cascaded up to 16 modules long and has a character set that can be customised to suit your application.



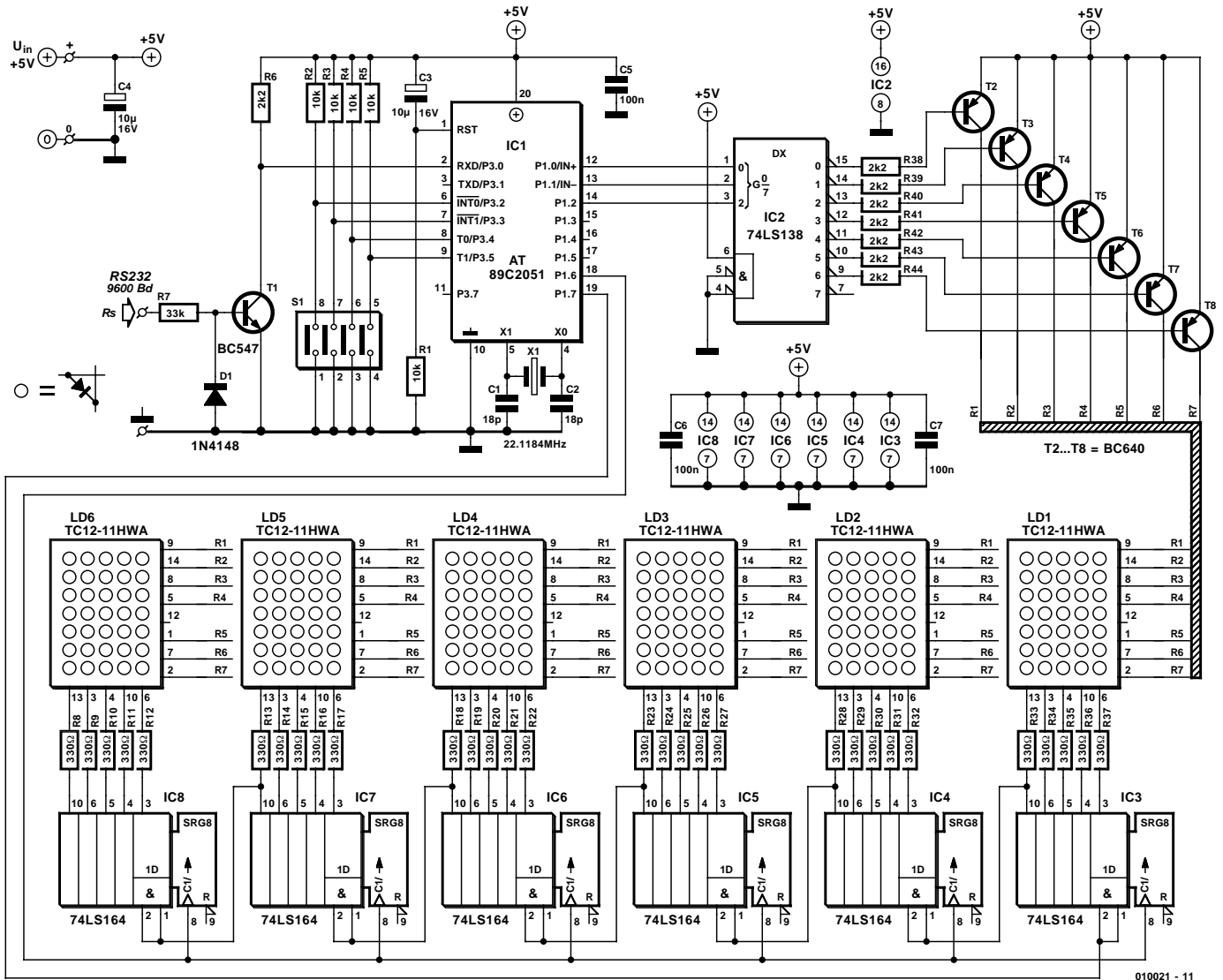
The dot matrix moving sign module presented here has a six character display area. A maximum of 16 of these modules can be cascaded together to produce a 96-character display. The PCB layout for the module can accommodate 5×7-Matrix displays with a distance between the rows of pins of 33.0 mm (30 mm character height) but larger or smaller displays may also be fitted by using adapter

boards, if the LED matrix is electrically compatible, of course.

The big advantage of building this moving sign yourself is that the cost of the finished article is much less than that of a ready-built unit. Most of the ICs used in the design are widely available and relatively cheap. The passive components are

all standard and an AT89C2051 with Flash-memory is not the most expensive microcontroller. The main item from a cost point of view is the type of LED matrix displays that are used.

The microcontroller that controls the display module has a character set of 95 characters. The pre-programmed controller is supplied with



010021 - 11

Figure 1. The circuit diagram of a dot-matrix module.

the standard character set containing punctuation marks, symbols, numbers and the alphabet (in upper and lower case) but there is also the possibility to program your own special characters and implement simple graphics.

Display Multiplexing

A separate 5-V mains unit supplies power for this module. Each displayed character is made up of a 5x7 or 8x8 LED matrix. Assuming that each LED requires a current of 10 mA, this gives us a worst case current consumption of 350 mA or 640 mA for each character. Multiplying this value by six for the six characters of each module may cause fear for a hefty power supply. In

order to reduce the current required for the module — and to simplify the hardware and wiring necessary to drive the LEDs — this design uses multiplexing. This technique switches on the driver transistor for each horizontal row of LEDs in the matrix sequentially. If the lines are switched quickly enough then your eye will not see the flickering and they will appear to be continuously lit. The trade-off is that because each row of the character is only on for one seventh (in practice less) of the time that it would be if it were continuously lit, the display will not appear as bright. In practice the LED driver must also be blanked while each new row of information is shifted into the display shift registers. To offset the reduction in dis-

play brightness the LED current has been increased.

Looking at the circuit diagram shown in **Figure 1** it can be seen that the column information for the display LEDs is supplied by the outputs of the six shift register ICs type 74LS164. Only five outputs are used on each shift register. The seven display matrix rows are selected by the three outputs of the microcontroller (P1.0 to P1.2) which control a one-of-eight decoder type 74LS138 (IC2). Two port lines from the microcontroller control the shift registers. Port 1.7 is the data signal while P1.6 produces the clock signal. The reset inputs to the shift registers are not used — in this application it would have the effect of turning on all the LEDs in the matrix.

The display process begins by the controller selecting the 'invalid' line 8 output of the display line driver (IC2). The effect of this is to turn off all the LEDs. This is necessary

to blank out the passage of serial data through the shift registers which would otherwise cause all the LEDs to flicker dimly. The microcontroller will take the ASCII code of the character to be displayed from its internal memory and uses this value to access its corresponding display pattern in a character generator. The controller will then take the first line of the display pattern and send it out serially to the display shift registers. Any bit in the pattern that is '1' will turn the LED off, any bit that is a '0' will turn the LED on. The controller also generates the clock to transfer this serial data into the shift registers. This process is repeated for the first line of the other five display characters. After this the transistor driving the first row (R1) is enabled. All of the shift register outputs that are low will cause these LEDs in the first line to light for a short period until the row driving transistor is turned off. The same process is now repeated for the second row up to the seventh when the whole process is repeated. This is repeated until the PC sends a new character.

The circuit design

Looking at the details of the circuit we can see that the row addressing is performed by the 74LS138 (IC2) with active low outputs. PNP Transistors T2 to T8 are switched on when one of the outputs of IC2 goes low. Each transistor collector is connected to the anodes of a complete row of LEDs while the outputs of the shift registers (IC3 to IC8) are connected via current limiting resistors to the cathodes of each column of LEDs. These LS series shift registers should be capable of sinking 8-10 mA on their outputs.

The LED dot matrix displays suggested for this design (Kingbright TC12-11 range) certainly produce the neatest finished product and also guarantee close matching of the LED's brightness. The Kingbright TC12-11HWA is stocked by Conrad Electronics and can be ordered from www.conrad-electronics.com. Unfortunately they are not really intended for multiplex operation. The light intensity of the LEDs are rated at 560 - 1400 μcd which is quite low. Even if the drive current is increased to 40 mA we only achieve a doubling of the brightness. The value of the current limiting resistors R8 to R37 can be reduced to 220 Ω if the display is not bright enough for your application. RS components stock the excellent hyper red matrix type TC12-11SRWA currently priced at £6.14 each and they can be ordered on-line from www.rswwww.com from their Electro-mail subsidiary. Kingbright also produce LED matrix's in the same series with different

colours but these may be difficult to obtain in small quantities:

TC12-11	Light intensity μcd ($I_F=10\text{ mA}$) min - typ.	colour
HWA	560-5600	Red
YWA	2200-5600	Yellow
SGWA	3600-9000	Green
EWA	3600-9000	Super red
SRWA	5600-14000	Hyper red

Another alternative is to make up your own LED matrix using individual LEDs. It is advisable to use 'low current' types as these will emit the same light intensity at 2 mA as a standard LED at 10 mA. The main disadvantage of this method is the time and patience needed to solder all the LEDs to form an accurate matrix.

Each display module has its own address. A four-way DIP switch (S1) defines the binary coded address of the module (0 to 15) and is read by the controller at port pins P3.2 to P3.5. The serial data line from the PC is connected in parallel to each module and this switch setting is used by the controller on each module to ensure that only the correct part of the message is used by that module. The LSB switch position of this address is between pins 1 and 8 of S1 and each contact is closed when the slider on the top of the switch is nearest IC1 (assuming that you have mounted S1 the right way round).

The RS232 interface is a little unconventional in that it does not use the ubiquitous MAX232 interface chip. In this application data flows in one direction only so a simple transistor is used as the most economical solution. Input resistor R7 is used to limit the input current when the PC drives the data line high. Diode D1 limits the negative input voltage at the transistor to approximately minus 0.7 V when the data line is driven low. Transistor T1 inverts the signal on the data line and applies it to input port 3.0 of the microcontroller.

This design operates without any handshaking protocol between the PC and the display module, but it is still necessary to connect the unused pins of the COM port to the correct signal levels so that the PC knows that the module is always ready to receive data. Some PCs are fitted

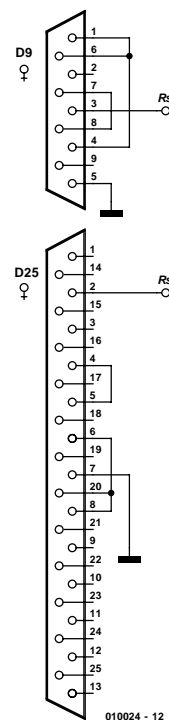


Figure 2. Wiring details of the COM port connector.

with a 25-pin connector while others have a 9-pin. For those with a 25-pin connector it is necessary to link pin 4 with pin 5 and also to link pins 20, 6 and 8. For those with 9-pin connectors it is necessary to link pin 7 with pin 8 and also to link pins 1, 4 and 6 (see **Figure 2**).

Communication from the PC to the module takes place at 9600 baud, with no parity bit, 8 data bits and one stop bit. The microcontroller clock is supplied by a standard 22.1184 MHz crystal (X1). Other crystals can be used but this would require a change to the initialisation routine of the microcontroller.

The complete module is powered from a single 5 V power supply. The LS-TTL chips used in the circuit consume almost 200 mA in their quiescent state. The LEDs will take approximately 10 mA each and because of the multiplexing only 30 LEDs (maximum) will be lit at any one time this gives a current of 300 mA. Adding this value to the 200 mA required for the control circuitry we arrive at a current supply of 500 mA at 5 V.

Figure 3 shows the layout of the PCB. We have not tried to cram everything into the smallest space,

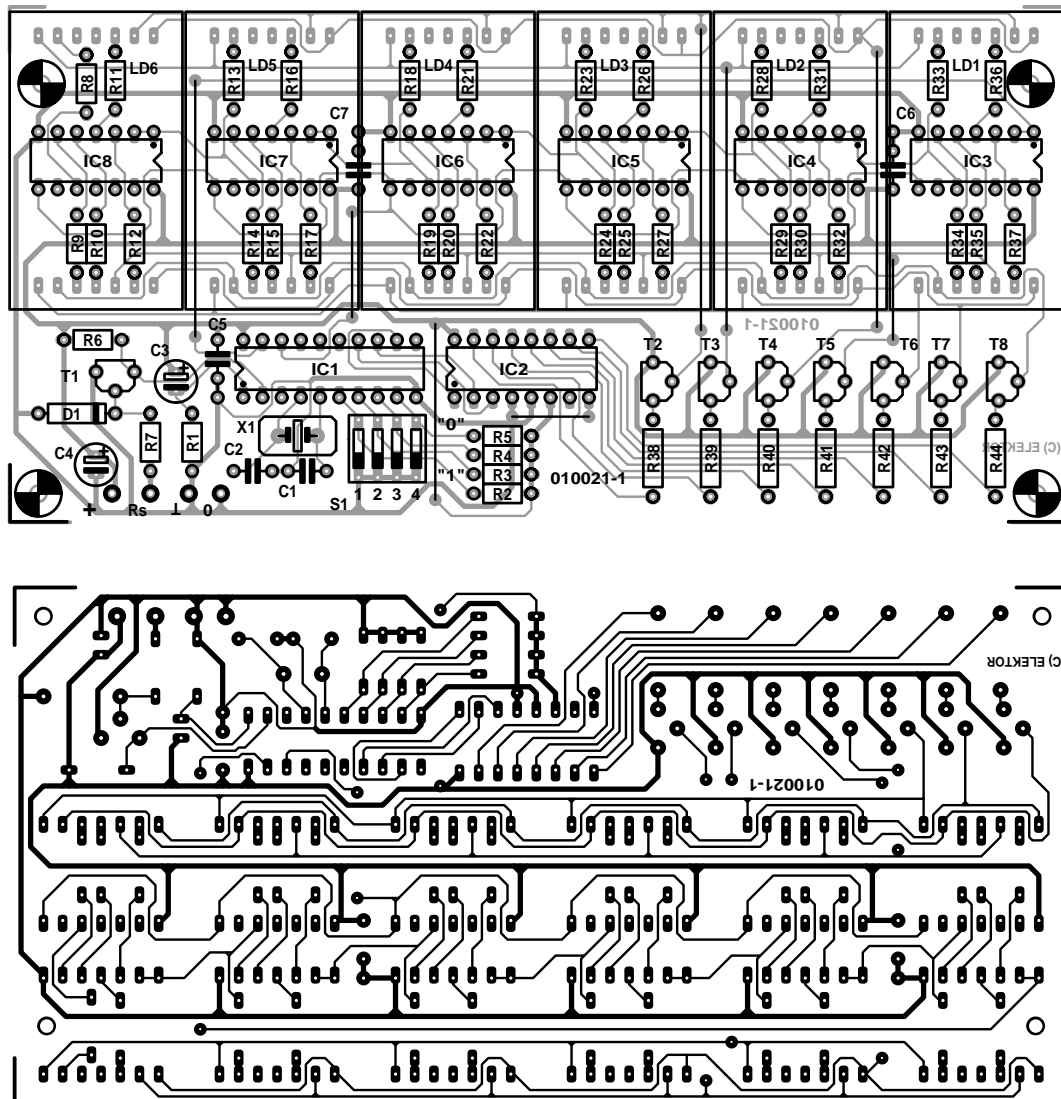


Figure 3. PCB Layout and component placement.

COMPONENTS LIST

Resistors:

R1-R5 = 10kΩ
 R6, R38-R44 = 2kΩ
 R7 = 33kΩ
 R8-R37 = 330Ω

Capacitors:

C1,C2 = 18pF
 C3,C4 = 10μF 16V radial
 C5,C6,C7 = 100nF

Semiconductors:

D1 = 1N4148

LD1-LD6 = TC12-11SRWA (Electro-mail) or TC12-11HWA (Kingbright, Conrad Electronics)

T1 = BC547

T2...T8 = BC640

IC1 = 89C2051-24PC, programmed, order code **010021-41**

IC2 = 74LS138

IC3-IC8 = 74LS164

Miscellaneous:

S1 = 4-way DIP switch
 X1 = 22.1184MHz quartz crystal Disk, order code **010021-11**

sockets. Sockets should however be used for all the remaining IC's and the LED matrix chips themselves. As always, make sure that you give the finished board a thorough visual inspection after you have soldered all the remaining components to the PCB.

The Controller Software

The microcontroller mounted on the display module requires a program to receive the display information from the PC and to control the hardware. The software is available as a free download from the Elektor Electronics web site at www.elektor-electronics.co.uk. Along with the fully commented Assembler source code there is also the hex file ready to send to your device programmer. The controller is available pre-programmed from Elektor Electronics for those constructors who do not need to 'tweak' the code in any

Figure 4:

```

REM Demo program for driving dot matrix display
REM using LEDs in 5*7 Matrix
REM Text is shifted from right to left
REM until the readout is cleared again
REM Then a pause of 1 s and repeat loop if no
REM key activity detected
REM -----
REM Init interface
REM -----
OPEN "COM2: 9600, N, 8, 1, CS, DS" FOR OUTPUT AS #1
PRINT #1, CHR$(12);
REM -----
CLS
LOCATE 10, 10
PRINT "Demo program for Running Text"
LOCATE 12, 10
PRINT "Stop by pressing any key"
Z$ = "Circuit developed for Elektor by Andreas Koehler."
REM -----
REM Output loop
REM -----
M2: V = 0
M3:   FOR ZA = V + 1 TO V + 12
      PRINT #1, MID$(Z$, ZA, 1);
      NEXT ZA
V = V + 1
  FOR ZT = 1 TO 50000
    NEXT ZT
    PRINT #1, CHR$(12);
  IF V < LEN(Z$) THEN GOTO M3
  SLEEP (1)
  A$ = INKEY$
  IF A$ = "" GOTO M2
END

```

Figure 4. BASIC Program to produce a running message.

way or do not have the means to program a chip themselves.

The controller software begins by initialising the stack pointer, the received serial data register and the timer register. The PBE routine is performed once during initialisation and reads the value of the DIP switch to find out the address of the module. An arithmetic and logic calculation is performed to work out the position of the characters in the received data stream that are intended for this module. The next five characters are loaded similarly into the memory. The LOE routine clears the display by sending out a series of 1's in the data stream that will turn off the outputs to the shift registers and also put an invalid address on the input of IC2. This has the effect of turning off the entire row driver transistors.

The counter register that is used to unload the serial data from the PC is reset and the registers holding the display characters are cleared to ensure that they do not contain any random information. After resetting the hardware the display will have a blank display and the module will be ready to accept display characters over the serial interface from the PC. When ASCII characters arrive at the controller, the SERIN interrupt routine will be started. The first character is temporarily saved to a register and after incrementing the receive character counter the character is checked to determine if it is the control charac-

ter 0Chex (form feed). If it is then we jump to LOE where the display will be cleared and the receive character counter reset. Another test is performed in SER0 to check if the received character is greater than 20hex any character less than this is invalid and would cause the character generator table pointer to point to a program code address rather than the character generator table resulting in a invalid display pattern. Other methods of character generation are too processor intensive and cause the display to flicker. SER1 tests if the received character is intended for this display position and checks that the address of the module calculated in the PBE routine, if there is no match then the routine will be exited and only the character counter will be incremented. The character will only be loaded into the register to be displayed if it is intended for this position and less than 20hex.

After receiving and checking the ASCII code for the character it will be converted into the correct display pattern. A program loop calculates the information of each display row from the ASCII character and the character generator table and stores it in register A. The ANZ routine shifts the character row information serially into the shift registers and also generates the clock. This process is repeated for each of the six display registers R2 to R7.

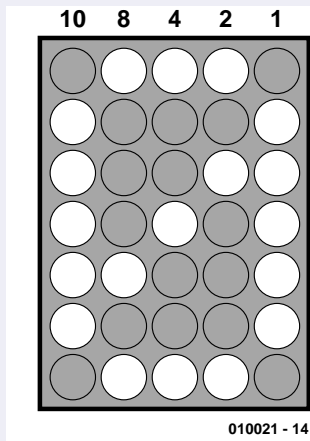
When a complete row is loaded to all of the shift registers, the line driver transistor will be switched on by sending the binary code corresponding to the line number to IC2. After this display period, the 'invalid' address code of 08 is sent to IC2 this will switch off all of the transistors. The next line will now be sent to the shift registers and this will continue for all of the seven rows of the display. The whole process is then continually repeated from the start.

The PC Software

If we take a look at the other end of the RS232 cable we can see that we need a PC running some sort of terminal program. This type of program will convert the characters to be displayed into ASCII and send them serially over the cable. Your PC will

Customise your symbols

In order to design your own symbols it is necessary firstly to have a copy of the controller program code (punzelek.a51). The procedure is relatively simple, all you really need is a simple device programmer and an Assembler program for the controller. The Assembler program is freely available as a download from: <ftp://www.atmel.com/pub/atmel/asmb51.exe> <ftp://www.atmel.com/pub/atmel/mlasm51.exe> The figure shows a 5x7 dot matrix displaying the character 0. Starting from the top line, if you add up the value of each LED in each line which is **not** lit and add this value to 0E0hex this produces the seven values for the seven lines of the displayed character that you will find in the character generator table at the end of the punzelek.a51 program code:



```
DEFB OF1H, OEEH, OECH, OEAH, OE6H, OEEH, OF1H ; 0
```

This process is a bit of a slog but it certainly improves your hexadecimal calculation!

instructive to write your own routine. **Figure 4** shows an example program written in BASIC, this outputs the character string contained between the quotes in the line beginning Z\$ via the COM2 port of the PC. The message runs from right to left over two cascaded display modules. The part of the program named 'output loop' contains a counter value (V) which indicates how many character positions there are in the module. The value in this output loop reflects the number of modules that are cascaded, this value is always a multiple of 6 and in this example with two modules the figure is 12:

```
FOR ZA = V + 1 TO V + 12
```

For three modules this value would be 18, and so on.

The value of ZT in the next part of the program is used to introduce a time delay that will control the speed of the characters moving across the display. A value of 50000 for this delay gives a reasonable appearance with our PC but if you are using a newer, faster machine it will be necessary to increase this value perhaps by a factor of ten or even one hundred.

Finally, we should repeat that the DIP switch setting is only read at power-on.

(010021-1)

be behaving just like a computer terminal rather than a stand-alone PC (hence the name of this type of pro-

gram). A proprietary program such as HyperTerminal could be used but it is often more interesting and

CONSTRUCTION GUIDELINES

Elektor Electronics (Publishing) does not provide **parts and components other than** PCBs, front panel foils and software on diskette or IC (not necessarily for all projects). Components are usually available from a number of retailers - see the adverts in the magazine.

Large and small values of components are indicated by means of one of the following prefixes:

E (exa) = 10 ¹⁸	a (atto) = 10 ⁻¹⁸
P (peta) = 10 ¹⁵	f (femto) = 10 ⁻¹⁵
T (tera) = 10 ¹²	p (pico) = 10 ⁻¹²
G (giga) = 10 ⁹	n (nano) = 10 ⁻⁹
M (mega) = 10 ⁶	μ (micro) = 10 ⁻⁶
k (kilo) = 10 ³	m (milli) = 10 ⁻³
h (hecto) = 10 ²	c (centi) = 10 ⁻²
da (deca) = 10 ¹	d (deci) = 10 ⁻¹

In some circuit diagrams, to avoid confusion, but contrary to IEC and BS recommendations, the value of components is given by substituting the relevant prefix for the decimal point. For example, 3k9 = 3.9 kΩ 4μ7 = 4.7 μF

Unless otherwise indicated, the tolerance of resistors is ± 5% and their rating is 1/8-1/2 watt. The working voltage of capacitors is ≥ 50 V.

In **populating a PCB**, always start with the smallest passive components, that is, wire bridges, resistors and small capacitors; and then IC sockets, relays, electrolytic and other large capacitors, and connectors. Vulnerable semiconductors and ICs should be done last.

Soldering. Use a 15-30 W soldering iron with a fine tip and tin with a resin core (60/40). Insert the terminals of components in the board, bend them slightly, cut them short, and solder: wait 1-2 seconds for the tin to flow smoothly and remove the iron. Do not overheat, particularly when soldering ICs and semiconductors. Unsoldering is best done with a suction iron or special unsoldering braid.

Faultfinding. If the circuit does not work, carefully compare the populated board with the published component layout and parts list. Are all the com-

ponents in the correct position? Has correct polarity been observed? Have the powerlines been reversed? Are all solder joints sound? Have any wire bridges been forgotten?

If voltage levels have been given on the circuit diagram, do those measured on the board match them - note that deviations up to ± 10% from the specified values are acceptable.

Possible corrections to published projects are published from time to time in this magazine. Also, the readers letters column often contains useful comments/additions to the published projects.

The value of a resistor is indicated by a **colour code** as follows.



color 1st digit 2nd digit mult. factor tolerance

color	1st digit	2nd digit	mult. factor	tolerance
black	-	0	-	-
brown	1	1	×10 ¹	±1%
red	2	2	×10 ²	±2%
orange	3	3	×10 ³	-
yellow	4	4	×10 ⁴	-
green	5	5	×10 ⁵	±0.5%
blue	6	6	×10 ⁶	-
violet	7	7	-	-
grey	8	8	-	-
white	9	9	-	-
gold	-	-	×10 ⁻¹	±5%
silver	-	-	×10 ⁻²	±10%
none	-	-	-	±20%

Examples:

brown-red-brown-gold = 120 Ω, 5%
yellow-violet-orange-gold = 47 kΩ, 5%

CAD in Europe (1)

a review of schematic drawing and PCB layout programs

By Guy Raedersdorf

If there's any field where developments come at a lightning pace, it's without doubt the realm of computer programs, especially those for the PC. The number of programs that are intended to make life easier for the electronic hobbyist just keeps on growing. We thought it would be a good idea to review several European programs for generating schematic drawings and PCB layouts.



First, a preliminary remark: in our opinion, the subject of simulation warrants an article of its own, which will perhaps appear within a few months. This is why we limit our attention in this article to programs for generating schematic diagrams and PCB layouts. Our adventure started with sending e-mail

messages to all of the publishers of these types of programs that we were aware of and that we considered including in our review. We expected to be able to relax for a few days after that, but positive responses came streaming in from

the next day onwards.

It is not our intention to make any sort of comparative study followed by a classification and the awarding of points. Instead, we want to present the various products to you and describe the specific features that attracted our attention in the short time that we used them. In the case of products that are distributed via the Internet, we also want to tell you where to find them.

It is humanly impossible to summarise all the features of a program in a few column-inches of text. Consequently, we must ask you to forgive us for a certain amount of 'superficiality'.

In the light of the number of responses that we received, we feel compelled to split this article into two parts. The first part, which is the article you are now reading, is dedicated to three programs that we received in complete versions, to wit:

- Multisim + Ultiboard
- Eagle Layout Editor
- Electronic Design Studio

The second article of the series deals with programs that we received as demo, evaluation or limited versions that we were able to access via the Internet.

Multisim and Ultiboard

Multisim

- Source: Electronics Workbench
- Version tested: Multisim 2001
- Medium: CD-ROM + serial number
- Language: English
- Installation: two steps
- De-installation: clean and easy
- Documentation: one thick, carefully prepared manual (in English)
- Price: from 399 ex VAT (personal version) to 4995 ex VAT for the Multisim Power Pro version with Verilog (Ultiboard is available as an option for both versions).

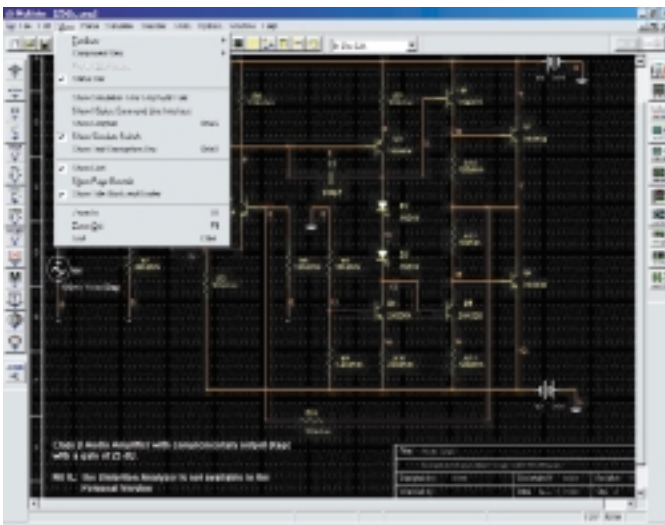
Ultiboard

- Source: Electronics Workbench
- Version tested: 5.63
- Medium: CD-ROM + serial number + dongle
- Language: English
- Installation: simple
- De-installation: quick and clean
- Documentation: 1 extremely thick, well-prepared manual

Internet addresses:

www.electronicworkbench.com

www.interactiv.com



The first programs that we received were the Multisim and Ultiboard duo from Electronics Workbench.

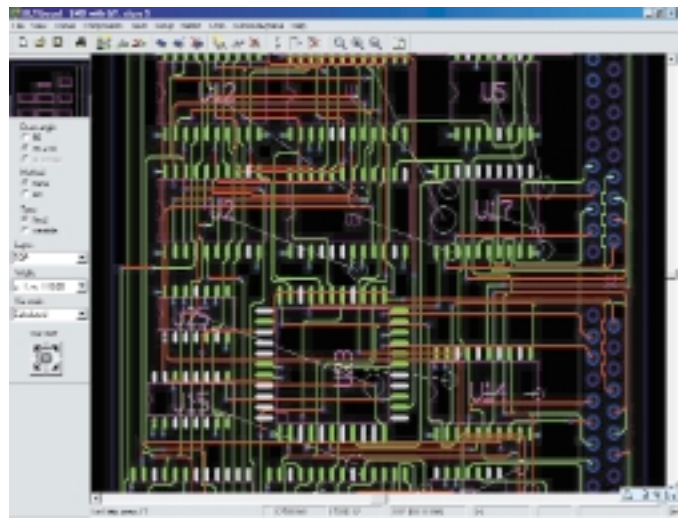
Multisim is a complete development tool that also includes elements falling within the scope of this article, namely a schematic drawing package with a very extensive component library. These programs also include other elements that fall outside our current scope, namely SPICE (a complete analogue and digital simulation program with a schematic drawing package and a VHDL/Verilog simulation package, FPGA/CPLD analysis and RF capability), post-processing options and problem-free coupling to layout development programs such as Ultiboard. As can be seen from the screen dump, there is a familiar schematic drawing environment with menus, Multisim toolbars and a component bar on the left-hand side. Placing components is simple — they are auto-

matically aligned to the grid — and making connections is child's play. Actually, generating schematic drawings for electronics projects is the axis about which everything else revolves.

The (relative) complexity of Multisim is related to the fact that it is in principle a simulation program that has been equipped with schematic drawing facilities and many other tools that are more specifically intended to be used for simulation.

You should not forget that Multisim, like many of the competing programs, allows you to modify existing components and even to create new components. The standard library contains 16,000 components. It would go too far to describe them in detail.

Next, let's look at **Ultiboard**, which is the PCB layout program. This program works with a dongle.



The installation of Ultiboard takes almost no time at all. It is not even necessary to restart the computer.

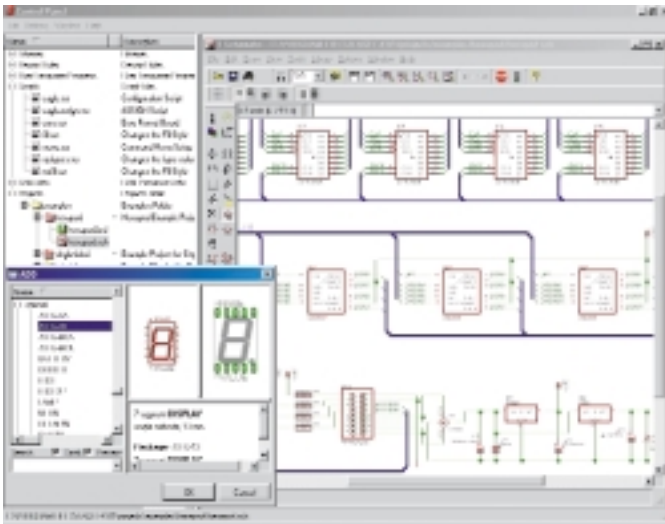
Ultiboard is a very complete layout program. There is a link to the schematic drawing program, which normally includes a simulation model and footprint for each component. This considerably simplifies the use of this powerful pair of programs. In conclusion, if all you want to do is to generate schematic drawings and PCB layouts, there are simpler solutions, but if you also want to check your schematics using simulations, you will need the Multisim module of the Multisim + Ultiboard duo.

Eagle Layout Editor

- Source: CadSoft (Germany)
- Version tested: 4.01
- Medium: CD-ROM + key disk
- Language: English
- Installation: simple
- De-installation: no problems
- Documentation: two manuals (in English) totalling 250 pages.
- Price: from 96 DM (incl. VAT) for the Light version to 2723 DM (incl. VAT) for the Professional version.

Internet address:

www.cadsoftusa.com



In case you are wondering where the name 'Eagle' comes from, it is an acronym for 'Easily Applicable Graphical Layout Editor' — that sounds promising!

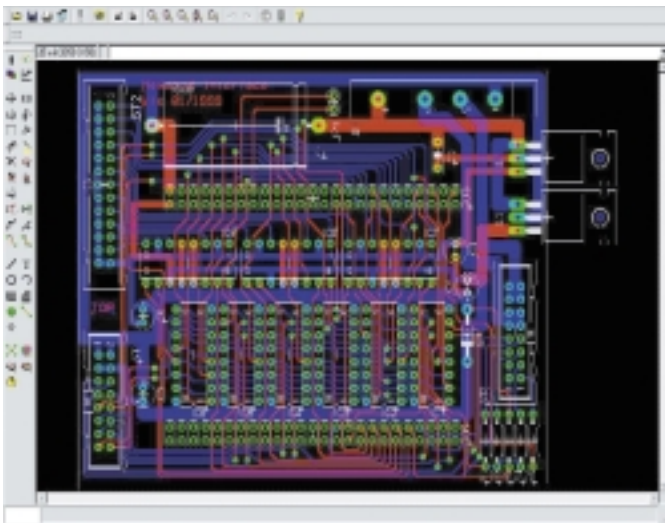
A few special features of Eagle:

Updated component management with a browser for viewing the various components.

A choice of different footprints, depending on the function and technology of the component.

Very up-to-date documentation, in the form of both printed material and PDF files.

A scripting language is included, to allow a number of routine procedures to be recorded.



The menu for the various functions can be seen as a tree structure on the left-hand side of the screen dump.

A stroll through the libraries is impressive. The Atmel library, for instance, has more than 50 entries for footprints and components. When you select a component, you can thus immediately specify the footprint belonging to that component.

You should not let anything hold you back from getting your hands on this program.

The tools for the schematic drawing and layout programs are extremely complete.

The final thing that we should mention is that **Eagle 4.0 Light** is available as freeware for non-commercial use. The program can be used with the accompanying freeware key for evalua-

tion or personal use.

We can conclude that **Eagle** is very user-friendly; with this complete program, you will be able to generate very nice schematic drawings and layouts.

Electronic Design Studio

– Version tested: EDS2 2.0D Advance

– Medium: CD+ key disk

– Language: English

– Installation: easy

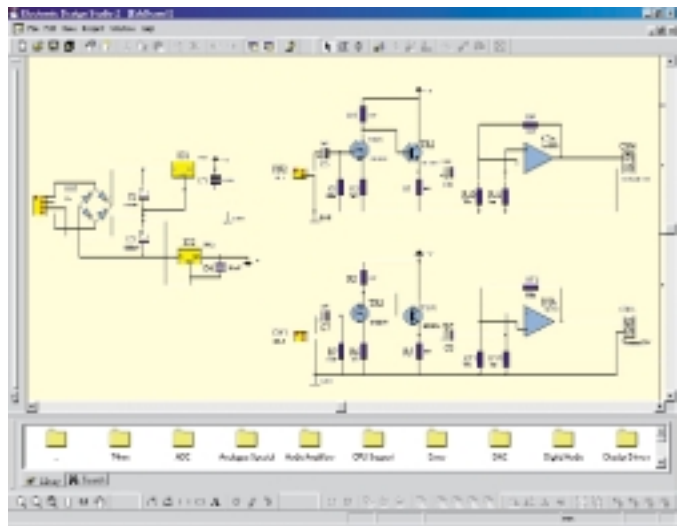
– De-installation: faultless, but don't forget to delete the key!

– Documentation: two manuals (in English) totalling 130 pages

– Price: from £199 ex VAT for the Standard version to £349 ex VAT for the Advance version.

Internet address:

www.quikroute.co.uk



The product range of Quikroute Systems Ltd consists of four versions of their schematic drawing and layout program: **EDS2 Lite, EDS2, EDS2 Advance and EDS Ultra**.

The differences relate to the features of the layout module. The three simplest versions can handle 32 layers, while there is no limit on the number of layers in the Ultra version. The Advance and Ultra versions have a 'rip-up and retry' function, a router that can handle multiple tracks with different widths at the same time and some other handy features, such as the creation of copper planes. The Lite version does not have the routing and simulation features. The resolution of the first three versions is 1 μm (1 micron), while the Ultra version has a resolution of 1 nm (0.001 μm).

As with many of the other programs presented here, Electronic Design Studio can lay claim to being a modular, integrated electronics CAD system.

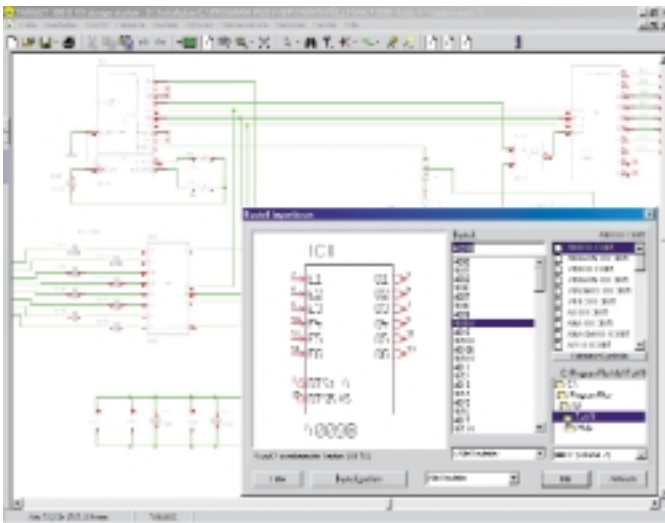
In the screen dumps, you can see that this is a professional-level schematic drawing and layout program. Just have a look at the 3-D form of the components on the printed circuit board.... The CAD/CAM printout driver also supports True-Type fonts. The schematic drawing printouts are tidy and certainly do justice to the new millennium.

With the exception of the Lite version, it is possible to work



with several sheets, each of which can be up to 2 m by 2 m (!). Components can be easily placed on the document, and numbering is automatic. EDS also automatically generates a netlist.

The three most powerful versions have a Viper autorouter. EDS automatically generates the printed circuit board layout from one or more schematic drawings or a netlist. The components are placed on the circuit board, and the interconnections are visible as a 'rat's nest'. The user can reposition the components and do manual routing, automatic routing or both together.



Although it is outside the scope of this article, we would like to mention that all versions except the Lite version have a simulation function. The package contains a coupon that can be redeemed for a Eurocard format film made from a Quickroute design.

EDS2 on the Internet: the version that we downloaded from the Quickroute requires a serial number, which you can obtain via e-mail or fax.

Important: before reformatting your hard disk, don't forget to move the key back to the diskette. The manual has an excellent description of how to do this. Make sure that the diskette is not write protected, since otherwise the program will refuse to start and an error message will appear.

With the description of the last of these four programs, we have nearly arrived at the end of the first article.

In order to avoid any misunderstanding, we should point out that the schematic drawings and circuit board layouts that can be seen in the screen dumps were not generated by the author. They are examples taken from the samples and tutorials delivered with the various programs.

Conclusion

It has not escaped our attention that over the years, certain collaborative arrangements have arisen between previously independent companies. Some examples are Ultimate (known for its Ultiboard and Ulticap programs), which has joined with Electronics Workbench, and OrCAD, which has merged with Cadence. There is no doubt that we can expect to see other 'marriages' in the future.

It is also entirely possible that even as you read this article, a new version of one of these programs will have already appeared, since everything moves super-fast in the ever-changing CAD world.

In conclusion, we would like to make one general remark. We see ourselves surrounded by programs that are steadily becoming more complex. This means that regardless of which program we are dealing with, we must take the time to read the documentation and to practice with simple circuits before tackling more complicated tasks. You certainly should not regard time spent in looking at examples (which are usually contained in a 'Tutorial' subdirectory) as being wasted.

(000148-1)

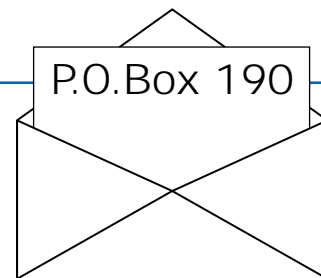
Next month's instalment will introduce the rest of the schematic drawing and printed circuit board layout programs, which we received in the form of demo, time-limited or evaluation versions.

In random order, these programs are:

- Spicycle™
- Easy-PC for Windows
- OrCAD Capture & OrCAD Layout
- Layo PCB
- Target 2001! V8.2F
- Proteus VSM
- Protel 99SE
- p-cad® 2000
- Ranger XL
- CIA04
- CAD PACK
- WINSCHEM & WINTYPON
- Windraft
- Winboard
- VUTRAX
- Edwin

A whole collection, as you can see!

We can only answer questions or remarks of general interest to our readers, concerning projects not older than two years and published in *Elektor Electronics*. In view of the amount of post and email received, it is not possible to answer all correspondence, and we are unable to respond to individual wishes and requests for modifications to, or additional information about, *Elektor Electronics* projects.



Missing DLLs?

Every month a number of readers inform us that certain system files are not included on diskettes supplied through our Readers Services, or in zip files downloaded from our website. In most cases, the problem is caused by the Windows Explorer which by default 'hides' system files. The solution is simple:

- Launch Windows Explorer
- Open the View menu
- Select 'Folder Options'
- Select 'View'
- With Hidden Files, tick the box 'Show All Files'

Having done this you should be able to see DLLs and other system files when using the Explorer.

IC Availability

Dear Editor — I have recently started to build your GBDSO and USB Audio DAC projects (October and December 2000 respectively, Ed.). Unfortunately I have been unable so far to obtain the following ICs: MC33182d, PCM2702E and OPA2353µA. Can you point me in the right direction?

R. Smith (by email)

Reportedly the MC33182d is hard to find. Fortunately, a near equivalent exists, the MC34182d which is a standard stock item with Farnell, Leeds. The problems with the PCM2702E and OPA2353mA were solved recently by C-I Electronics having obtained a certain quantity for one-off reselling.

Contact C-I Electronics by fax on +31 10 4861592 or by email dil@euronet.nl.

PC Audio

Dear Editor — Referring to your excellent Audio ADC2001 published in the March 2001 issue, I

would like to add a tip that may be useful for PC-audiophiles like myself. I was fortunate to be able to test a number of soundcards with S/PDIF inputs and outputs. My conclusion was that some of these are less than perfect.

The first card I got my hands on was a Terratec EWS64 which appeared to work just fine although it employs an ISA slot. However, having made a recording, I found that the output signal was 3 dB lower than with playback (I refer to digital signal levels only), and this proved very hard to correct using the software. More seriously, a level difference of 0.5 dB was measured between the left and right channel.

The next card was an Audioworks 32 from Emagic. Although the signal quality was beyond reproach, the priority assigned to the drivers is far from ideal. It is not even possible to open the Windows Notepad without causing dropouts in the recorded or played back signal. The Emagic helpdesk kindly advised me to use the AISO drivers. Unfortunately Windows applications supporting these drivers are few and far between.

The last card was a Di02448-man from M-audio. This card I found came up to all requirements. Moreover it is relatively inexpensive for a professional product, and 100 per cent stable. Not a single recording or playback error was noted, not even when scanning in a photograph in Adobe PhotoShop. The latest drivers also enable excellent MIDI reproduction. More info on this card on www.midiman.com.
Mark Wuyts, Netherlands

Electronics Online: Overclocking (June 2000)

Dear Editor — I enjoyed the informative and interesting article on overclocking in the May

2001 edition of *Elektor Electronics*. I am an overclocker myself, my first overclock was quite some time ago, overclocking a Pentium 1 166 MHz up to 233 MHz, and since then I've overclocked nearly every computer I've owned. I am currently working on a water-cooled, Peltier-assisted rig that will cool my motherboard chipset, CPU, and graphics card and RAM, in an attempt to get above the current spec I have, which is an AMD T-bird 1-GHz at 1.33 GHz. I hope to get as high as 1.5 GHz with water-cooling.

I cannot stress enough the importance of care and quality when overclocking. The current Athlon T-Bird CPUs are very delicate. The CPU core is ceramic and is very small. As a result the core can be cracked easily if too much force is applied to the chip. Taking heatsinks on and off regularly to replace them, or adjust the multiplier settings on the L1 Bridges is enough to crack the core, and as these are very expensive chips, inexperienced readers of *elektor* could easily do damage to them.

Another thing I feel I should add is that AMD CPUs are HOT, very hot. These things at default voltage dissipate more heat than the average 60 W light bulb, in an area that is only about 1 cm square. It is an absolute MUST that a heatsink should be on these when running. Even booting into bios without a heatsink is enough to fry the CPU, so even if you are 'just checking' to see if your updated L1 modifications have worked, a heatsink has to be on. This applies to processors running at their default speed, not just overclocked ones.

Another point I must add is that when buying new heatsinks, most come with a TIM, or thermal interface material, glued to the heatsink base. These are OK if you are running at default

speeds and voltages, but when overclocking they are a nuisance. I suggest removing them carefully with a razor blade and instead using a silicon or silver based heat transfer paste, as used with power transistors. This not only increases the heatsink's efficiency enough to drop its effective temperature by a few degrees, but also prevents from happening what I have seen on a few websites, where the TIM becomes so hot it melts like plastic and destroys the CPU.

I would suggest as another good overclocking website www.hardocp.com. The humour is a little strange, but the articles are informative. The website www.tomshardware.com is good too. As a final note, it is vital to keep track of your processor's temperature. The BIOS on most motherboards lets you do this, but the excellent **Motherboard Monitor** software available at <http://mbm.livewiredev.com/> is vital for overclocking. It allows you to view temperatures from windows, generates log files, and can shut down the computer if any serious problems occur.

That's all I have to say, except that I urge new overclockers to trawl the web, read every article they can on overclocking, and ensure they know what they are getting into before they try.

I am very pleased with the recent articles on audio circuitry by the way. The Crescendo Millennium was good to see. I was considering building the original version a few years ago, and now I'm saving up to get the components for the new model. I also enjoyed the PPP Valve Power Amp in this month's edition (ref. June 2001, Ed.). Thanks for your time.

Conor Fitzpatrick (by email)

(a.k.a. a freezin' fitz in the overclockers part of the web)

Metronome/Diapason

based on a PIC16F84 microcontroller

By C. Salmeron

This article describes a metronome and a diapason (electronic tuning fork) combined into a single unit. The circuit design philosophy was to keep the hardware as simple as possible, and to provide features comparable to those of high-end commercial units. This has been achieved by using a microcontroller chip PIC16F84.



The PIC16F84 was programmed to take care of all functions, including the generation of beat *ticks* in metronome mode, the digital synthesis of pure tones (sinewaves) for the full range of notes (from C₂ to B₇) in diapason mode, and the driving of the LCD display module.

In metronome mode, the circuit produces beat *ticks* whose sound is short and sharp, imitating that of a mechanical metronome. A

useful feature is emphasising the first beat in each bar. This is done acoustically, by changing the pitch of the beat, and visually, by the lighting of a LED. An interesting feature is that the unit can provide irregular rhythms of 5/4th, 7/4th and 8/4th, as well as no rhythm — only regular beats imitating a mechanical metronome.

In diapason (tuning fork) mode, any note from C₂ to B₇ can be generated and two tuning scales can be chosen: fifths or *equal temperament*. The tone generated is quite pure with low harmonic content and is crystal controlled, for stability and better accuracy in frequency. Also, the circuit does not need any adjustment or pre-tuning: it is always on tune.

Circuit description

The heart of the circuit (see **Figure 1**) is a PIC16F84 microcontroller, from Microchip (IC1). As many of you will be aware, the PIC16F84 microcontroller is a complete computer on a chip with 1 K (words) of Flash EPROM, 68 bytes of data RAM, 64 bytes of data EEPROM, a clock circuit, 13 I/O pins, a real time clock/counter, and several other useful features. For a full description of this microcontroller and its instruction set, see the Microchip data book or look at www.microchip.com. You can download the free datasheets of this microcontroller as well as a good collection of application notes and software examples.

After developing the first version of the circuit, and after realising that there was enough free program memory in the microcontroller for

Main features

Metronome mode

- Rhythm variable between 40 and 240 beats per minute. Crystal controlled, giving high precision in time measure.
- Beats sound is short and sharp, imitating that of a mechanical metronome.
- The first beat of each measure has a different sound (higher pitch) and the led flashes.
- In the display each part of the measure is shown numbered. The numbers shift across the display to give a visual picture of the bar.
- Sound and LED can be turn on/off as desired, in independent way. Turning them on or off does not alter the internal time measurement.
- Different rhythms can be selected:
 - Regular rhythms from 2 to 7 beats per bar
 - Irregular rhythms of 5/4th, 7/4th and 8/4th
 - No rhythm: only regular beats imitating a mechanical metronome
- All control information is permanently shown in the display.
- Headphone output and volume control

Diapason mode

- Any note, from C₂ to B₇, can be generated. Selection by octaves or incremental, note by note.
- Two tuning scales can be chosen: by fifths (F) or equal temperament (T).
- The sound generated is pure with no harmonic content. Crystal controlled for precision in frequency and stability.
- Display shows selected note and tuning scale.
- Headphones output for better quality at higher volume settings. Headphones also recommended for very low notes i.e., C₂ to B₂.

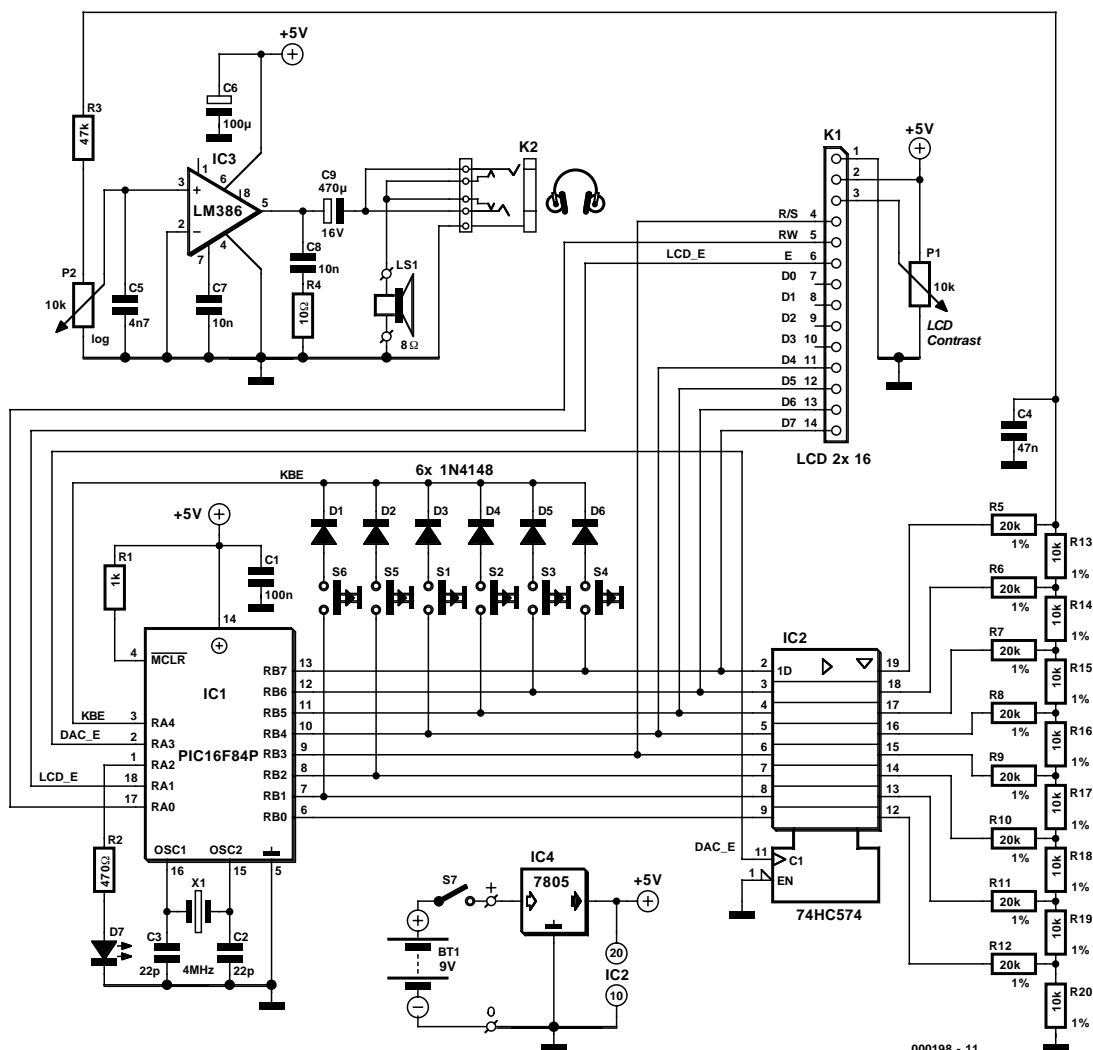


Figure 1. Circuit diagram of the Metronome/Diapason.

additional functions, it was decided to improve the quality of the sound produced by the piezoelectric sounder originally used as the sound producing device. To any musician, the *clicks* produced by a mechanical metronome are more pleasant than the short whistles produced by a piezoelectric transducer. Also, a good diapason should produce pure sinewaves, not corrupted by harmonics; this was quite far from the sound produced by the piezoelectric transducer excited by a squarewave. On the one hand, the square-wave used to excite the piezoelectric transducer was found to have a high content of odd harmonics, especially the third. On the

other hand, due to the fact that piezoelectric transducers have strong peaks in their frequency response, the volume of the sound varies greatly depending of the note produced, and, for some notes, the third harmonic prevails over the fundamental. So it was decided to develop a circuit with better audio characteristics. The result is presented in this article.

Apart from the microcontroller IC1, the rest of the circuit consists of an LCD module, six push buttons, a digital-to-analogue converter (built

around IC2), and a small audio amplifier Type LM386 (IC3). The discrete DAC is built around a 74HC574 and an R-2R resistor ladder network.

One interesting thing to note is the way the bus RB0-RB7 is shared between different devices: the keyboard, the LCD unit and the DAC converter. To share the bus, the devices connected to it must be capable of being disabled when other devices use the bus. The LCD module and the DAC converter both have *enable* lines, useful for this control. For controlling the keys, each key

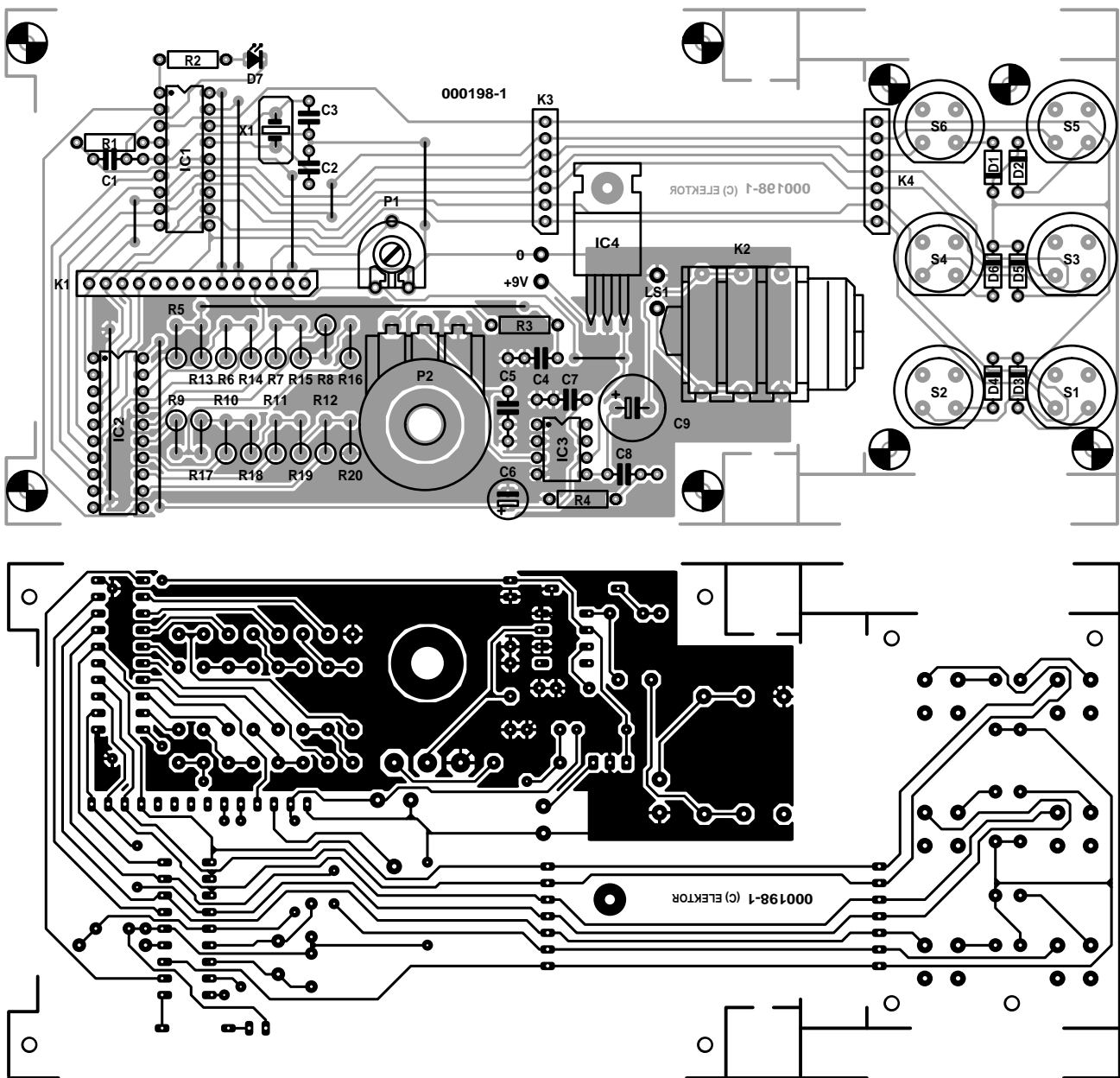


Figure 2. Copper track layout and component mounting plan of the PCB designed for the Metronome/Diapason (board available ready-made).

is connected in series with a diode (D1 through D6) to a line labelled as KBE (keyboard enable). The diodes play a double role. On one hand, if two keys are pressed simultaneously, the diodes prevent a short circuit between the corresponding I/O pins of the microcontroller. On the other hand, when the line KBE is pulled up to high level the keys are electrically isolated, avoiding interference with the other devices connected to the microcontroller bus.

An important practical observation is concerning the loudspeaker.

For this design, a miniature loudspeaker is used. The volume and quality of the sound will vary considerably between one unit to another. This was noticeable not only in the intended frequency range in diapason mode but, more surprisingly, in the quality of the beats produced in metronome mode: while with some loudspeakers the sound was like a pleasant short hit, in others it was a barely audible click. So we recommend experimenting with different loudspeakers you may have available.

An optional headphones output is provided by socket K2. Useful for practising without disturbing the neighbours!

A final observation about the circuit is the low-pass filter behind the DAC. It is just a simple RC network (in fact, the combination of C4, R3, P2 and C5). As already mentioned, one design constraint was to keep the physical dimensions of the circuit as small as possible. Due to this constraint some quality in the audio part had to be sacrificed as a result of this simple filter. Nevertheless, you should take into account that when in metronome mode, to imitate the *click* sound of a mechanical unit, the loudspeaker is driven by a very

COMPONENTS LIST

Resistors:

R1 = 1 k Ω
 R2 = 470 Ω
 R3 = 47k Ω
 R4 = 10 Ω
 R5-R12 = 20k Ω 1%
 R13-R20 = 10k Ω 1%
 P1 = 10k Ω preset
 P2 = 10k Ω logarithmic potentiometer

Capacitors:

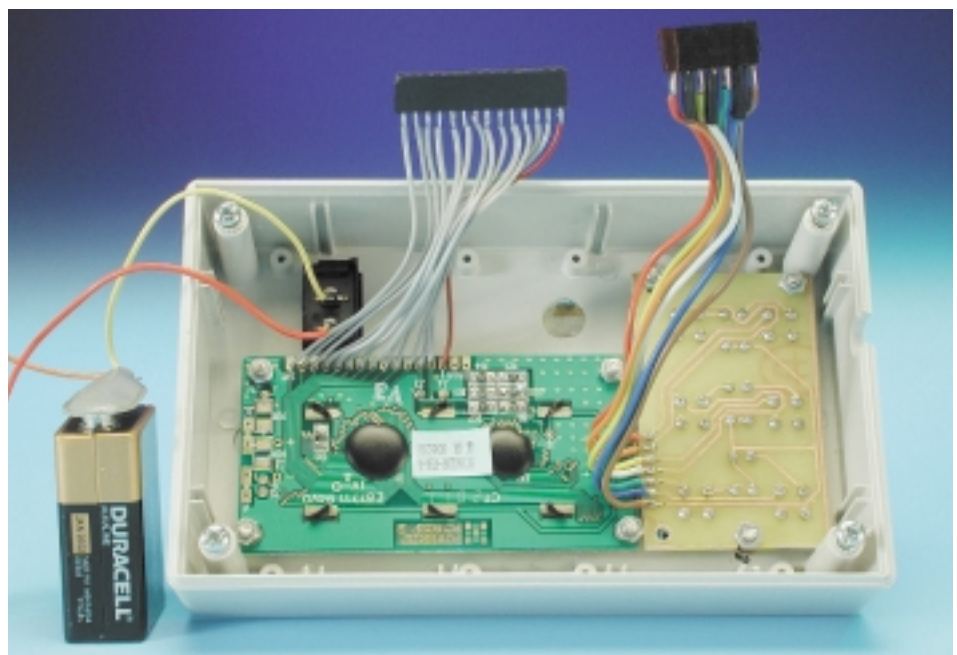
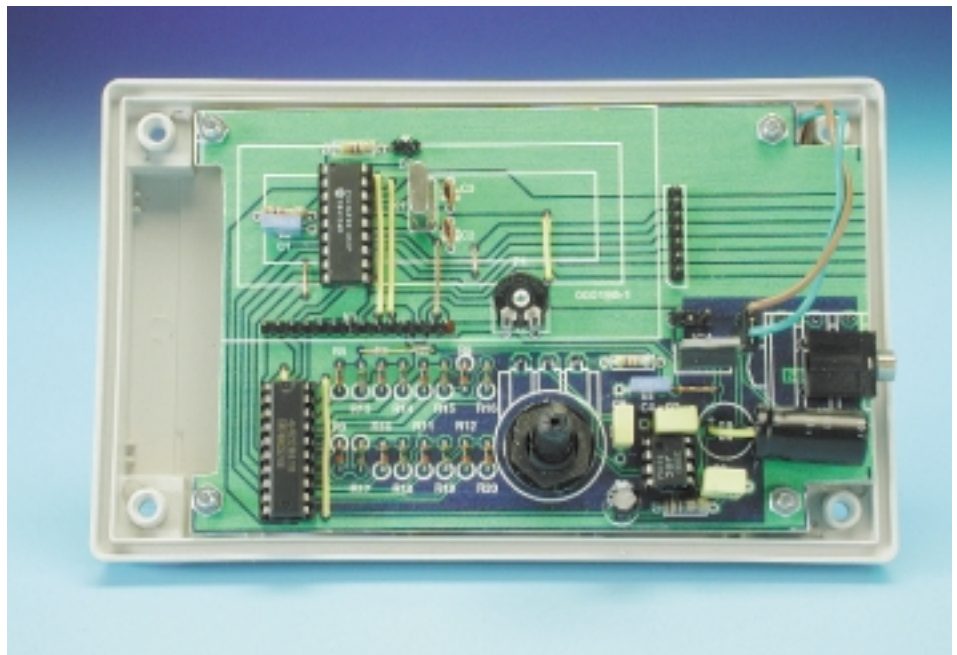
C1 = 100nF
 C2,C3 = 22pF
 C4 = 47nF
 C5 = 4nF7
 C6 = 100 μ F 16V radial
 C7,C8 = 10nF
 C9 = 470 μ F 16V radial

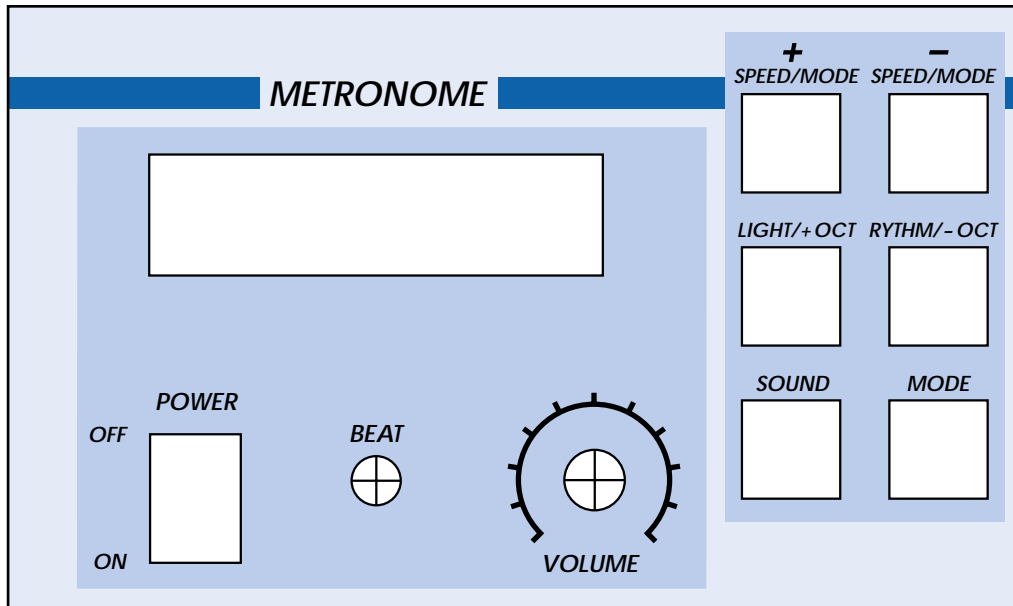
Semiconductors:

D1-D6 = 1N4148
 D7 = LED, red, high efficiency
 IC1 = PIC16F84-04/P, programmed, order no. **000198-41**
 IC2 = 74HC574
 IC3 = LM386
 IC4 = 7805

Miscellaneous:

S1-S6 = pushbutton, Multimec CTL4 with cap D6K0
 K2 = headphones socket, stereo, PCB mount, for 6.3mm dia. jack plug
 LCD with 2 lines of 16 characters, e.g., PCF1602
 Battery clip for PP3 (9V) battery
 Enclosure: Retex, dim. Approx. 145 x 90 x 45mm (Conrad Electronics # 54 15 59)
 PCB, order code 000189-1
 Disk, all project software, order code **000189-11**





000198 - F

Figure 3. Suggested front panel layout at true scale.

short square wave burst (in fact, three or four cycles) to force the loudspeaker to vibrate quasi-freely, as if its cone has been hit with a stick. To get this effect it is very important that the driving signal has steep edges, which implies high harmonic content. Therefore, the cut-off frequency of the filter should be high enough to get a pleasant sound in metronome mode. Consequently, the filter was designed to strike a compromise between two opposing requirements: those of the diapason (to avoid harmonic content) and those of the metronome (to get a signal with a steep slope).

Voltage regulator IC4 reduces the battery supply voltage to +5 V.

Program description

The operation of the program that controls the PIC is quite simple. The source code is fully commented and readers should not have any problem understanding its operation. All project software is available either on diskette (Publishers' order code **000198-11**) or as a free download from our website.

Metronome mode

After initialising all the necessary data and displaying a welcome message for one second, the program enters into the main loop in metronome mode. In this mode the microcontroller must attend quasi-simultaneously to the execution of four tasks: 1) produce ticks sounds and flash the LED at relevant times;

2) update the information on the display; 3) scan the keyboard and process the function associated with key pressed, if any, and 4) update time counters. This fourth task is the most critical in metronome mode, as the main function of a metronome is to keep time count accurately. The other tasks are less critical, and a delay of a few milliseconds in attending to them is irrelevant. In this program, everything is synchronized using the timer facilities of the microcontroller. Timer0 interrupts are enabled and an interrupt is produced every 256 μ s. An interrupt service routine (ISR) then takes control every exactly 256 μ s and, in this routine, different time counters are incremented; also, the program checks if a tick sound is due for producing. If do, a flag is set. When the ISR routine finishes its work, control is returned to the instruction to be executed when the interrupt occurred. The other three tasks are executed in succession inside the main loop.

Diapason mode

In diapason (tuning fork) mode only two tasks need to be executed: producing a sine wave of the desired frequency and checking for key activity.

The technique used to produce a continuous sine wave is based on direct digital synthesis (DDS). In DDS, a counter (phase accumulator) is used to step through a sine look-up table, which is then fed into a digital-to-analogue converter (DAC).

In this program, the phase accumulator is a 16-bit variable called nPhase (implemented as two variables nPhaseH, nPhaseL). With each loop iteration the value in nPhaseInc is added. The variables nPhase and nPhaseInc can be regarded as having an integer portion *i* and a fractional portion *f*, i.e., *iiiiiii.fffffff*, where the integer portion represents steps of 1/256th cycle. This integer portion is used as an index into the sine table and the value obtained from the table is then output to the DAC.

On each loop iteration the keyboard is read to verify if any new key is pressed. If so, control is transferred to the 'read keyboard' routine, interrupting the production of the sine wave. The port B interrupt facility was not used as it is only available for four lines (RB0-RB3) and this application requires control over six keys. Although testing the keyboard on each cycle increments the loop time, the resulting sampling frequency is still high enough

(> 20 kHz).

The phase increments necessary to generate sine waves of the appropriate frequencies are stored into one look-up table. This table was computed with the formula:

$$\text{Phase increment} = 256 \times f_{\text{required}} \times T_{\text{sampling}}$$

The factor 256 accounts for the effective domain of the sine function (256 steps = 2π radians). As the phase accumulator has a decimal part of 8 bits, the value 65536 is used in the formula, instead of 256. Since the sampling frequency is 25 kHz, as will be explained later, this gives a sampling period of 40 μ s. In computer-speak, the formula becomes:

$$\text{Phase increment} = (\text{int})(65536.0 * 40.0e-6 * \text{Frequency})$$

The phase table was generated by the program PHASETBL.C leaving the output in a file: PHASETBL.EXE > PTBL.INC. The output file PTBL.INC was then merged into the source file. The phase table is, in fact, two tables, as the frequencies to generate depend on the tuning mode. One table is for 'equal temperament tuning' mode and the other for 'tuned by fifths' mode.

Once the phase has been calcu-

lated, the corresponding sinewave amplitude is looked up in the sinewave table. This table contains the values (8 bits) of a sine wave in 256 steps. The table is scaled so that value range [-1.0; 1.0] is mapped to [00h; 0FFh]. In order to save space, only values for the range 0 to 90 degrees are stored — a full sine wave table would take up 256 words. The 8th bit represents the sign, so absolute values are stored as seven bits. Values are generated using the formula:

$$127 * \sin(\text{phase} * 2 * \pi / 256)$$

where phase is in the range 0 to 63. The sine table was generated by the program SINETBL.C leaving the output in a file: SINETBL.EXE > STBL.INC. The output file STBL.INC was then merged into the source file.

Returning to program operation, when the program enters into diapason mode all interrupts are disabled and, after some initialisations, the program enters a loop. In this loop four tasks are executed in sequence:

- update the sine wave phase. This takes 6 μ s;
- calculate the sample value by looking up in the sinewave table (14 μ s);
- send the sample value to the DAC

(5 μ s); and finally

- verify if any key is pressed and if not, repeat the loop (15 μ s).

The total loop time is always fixed, unless a key is pressed, and it takes 40 μ s. This implies that the sampling frequency is 25 kHz, high enough for the maximum sound frequency to produce (B₇, nearly 8 kHz) whilst obviating the need for strong low-pass filtering after the DAC.

When a keyboard activity is detected, the sinewave production is suspended and the key function is processed.

Construction

The PCB artwork for the project is given in **Figure 2**. Construction of the metronome is fairly easy as no adjustment is necessary apart from the desired contrast of the LCD. Depending on the enclosure you intend to use, the keyboard section may or may not be separated from the main board. If you employ miniature resistors, these may be mounted horizontally rather than vertically as indicated by the component overlay.

Check the orientation of all polarised devices, particularly the diodes connected to the keys. For the display, any general-purpose 2-line, 16-character display having the Hitachi HD44780 chip should be suitable. The HD44780 is widely used between display manufacturers. If the LCD you employ has a backlight, then this should be connected extra. No provision has been made for adjusting the frequency of the crystal oscillator, as any error should be negligible. Nevertheless, should you like to adjust it, simply substitute one of the capacitors C2 or C3 by a small trimmer of 40 pF.

A suggested fascia for the Metronome/Diapason is shown in **Figure 3** (not available ready-made).

Operation

When the unit is switched on, the welcome message is displayed for about one second. Next, it goes into Metronome mode with ternary rhythm and a tempo of 120 beats per minute. The user can then change the settings as desired by pressing the keys. The key functions should be easy to learn once used a couple of times. Some labels on the keys have two names: the first one is applied when in metronome mode and the second when in Diapason mode. The functions in Metronome and Diapason mode are explained in **Table 1** and **Table 2** respectively.

(000198-1)

Table 1. Functions available in Metronome mode

KEY	FUNCTION
+	Increment tempo (bpm). Maximum is 240 bpm.
-	Decrement tempo. Minimum is 40 bpm.
Light/ -Oct	Switch LED on / off.
Rhythm/ +Oct	Cycle through different rhythms: 1, 2, 3, 4, 5, 6, 7, 5/4th, 7/4th, 8/4th.
Sound	Switch the sound on / off.
Mode	Change to Diapason mode, equal temperament tuning.

Table 2. Functions available in Diapason (tuning fork) mode

KEY	FUNCTION
+	Increment note. Highest is B ₇ .
-	Decrement note. Lowest is C ₂ .
Light/ -Oct	Decrement octave, e.g., if current note is G ₄ , change to G ₃ .
Rhythm/ +	Oct Increment octave.
Sound	Switch sound on / off
Mode	From equal temperament tuning, change to tuning by fifths. From tuning by fifths mode, change to Metronome mode.