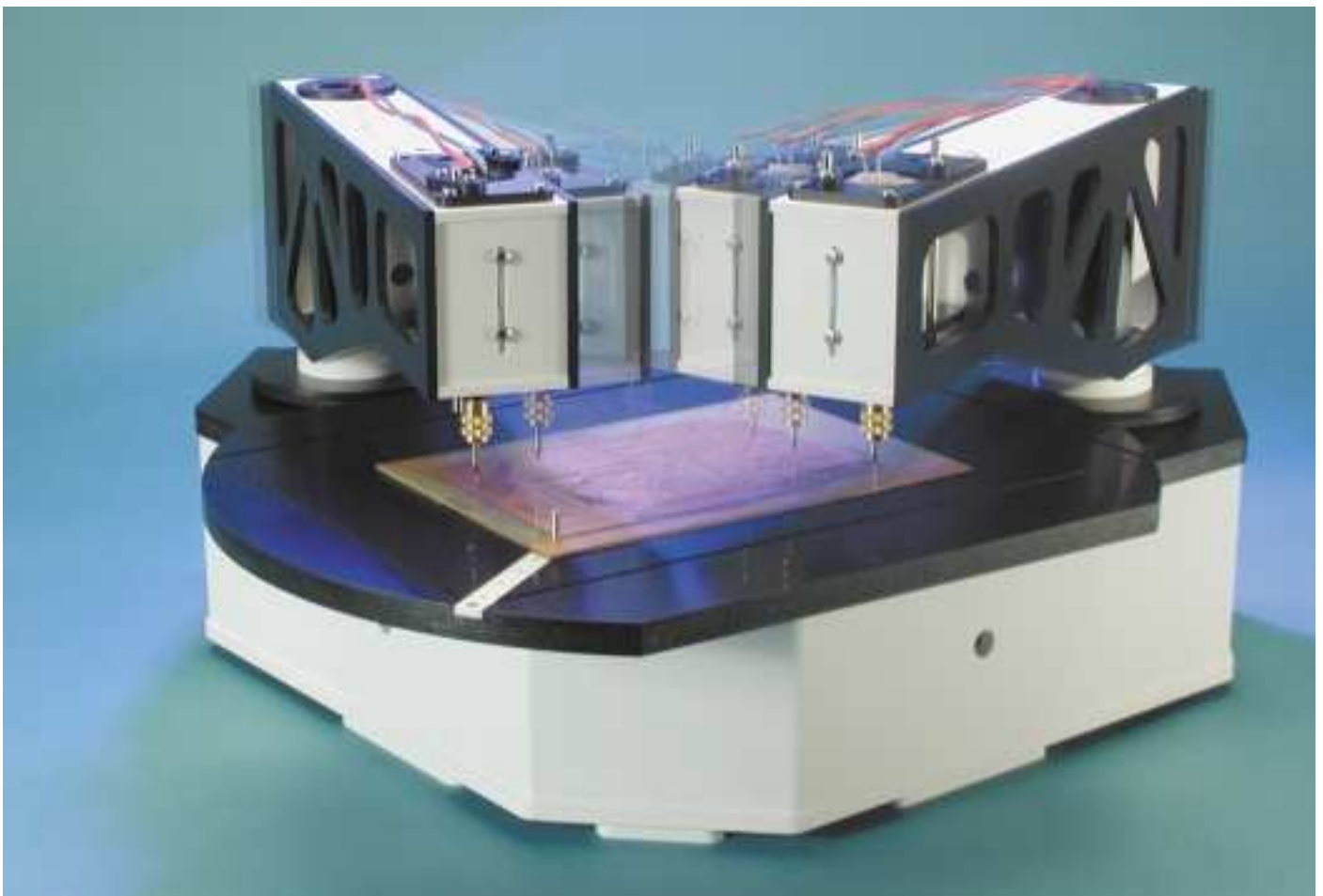


PCB Drilling Machine

Part 1: New solutions to old problems

By T. Müller (Radix GmbH)

Elektor Electronics is proud to present an unprecedented home construction project: a CNC PCB drilling machine which is economical, accurate, quick, and bristling with new ideas.



Large-scale electromechanical projects have tended to be the exception rather than the rule in *Elektor Electronics*; but these exceptions have always been well received. Many readers will have fond memories of the Plotter and Plotter Mark 2 projects, which hundreds of constructors across Europe have built and

which even fifteen years after publication still boasts a large fan club.

There is a very good reason behind the idea of presenting an electromechanical project for home construction. Rather than the fascinating and in places supremely ele-

gant and unusual design, it is the cost saving that drags the ambitious hobbyist away from the soldering iron to less familiar tools. Home construction of a plotter made a lot of sense when a ready-made commercial device was outside the financial

grasp of the electronics hobbyist.

These days, however, plotters only find specialist application, and low-cost ink-jet printers do the 'plotting' work; and no-one would consider producing a printer for home construction when mass-produced commercial devices are so cheap.

So such a project in *Elektor Electronics* must be timely and have a 'best before date' well in the future. The computer-controlled PCB Drilling Machine described in this article and the instalments to follow fits the bill. Commercially available machines with comparable specifications are around four times as expensive as the one presented here, and are hardly likely to become mass-produced commodities sold at bargain basement prices. In conventional CNC drilling machines the mechanical parts are too expensive and too complex, and cannot be made from cheap materials.

This opens the way for new ideas. The author has developed a CNC PCB drilling machine that is distinctive in two ways: first, in its economical mechanical construction, using moulded plastic parts rather than the conventional multitude of aluminium and steel components; and second because of the supremely elegant and unusual design.

We would like to describe a little of the story of the PCB drilling machine in a series of constructional articles. If you decide to construct the machine, you do not of course have to make the components yourself. Through his company, the author is offering to supply a complete kit of mechanical parts; *Elektor Electronics* (and certain component distributors) the electronics. The basic construction kit should not cost more than about five hundred pounds. More on this, and delivery details, will appear on the second instalment in the next issue of *Elektor Electronics*.

Specialist mechanical skills are not required. You will be as surprised by how foolproof construction of the machine is as we were in the *Elektor Electronics* laboratory when we first saw it run. And even if you are not planning to build the drilling machine, we hope you will still find the articles interesting!

PCB Drilling machine

Concept

CNC machine with 2.5D operation capable of vertical drilling in circuit boards up to 300x200 mm²

Construction features

- No torsion on machine, since all points where forces are applied lie at centres of rotation
- No expensive sliding components such as V-grooves, linear bearings or ball races
- Self-calibrating drill advance using magnetic drive
- Low torque about base fixing points

Functional features

- Up to four tools in use simultaneously
- No tool changing required
- Greatly improved speed moving from drilling point to drilling point
- Clear and easy access to table from above

Benefits

- Circular turntable allows fixing points outside the usual rectangular workpiece: conventional tables lose working area to fixing points
- Small footprint due to efficient use of area
- No ball races, no worm gears
- No thermal problems due to non-planarity
- Simple construction, no calibration for planarity or squareness
- No force on drilling head from trailing cables
- All heavy and sensitive components underneath or clear of working area
- Swarf cannot jam mechanism

From Idea to Circuit Board

The individual steps in developing an electronic circuit, from the idea through to the populated prototype circuit board, usually follow on from one another more or less smoothly.

First of all there is the initial design. Depending on the experience of the designer and on the complexity of the circuit, small parts of the design may first be tried out on prototyping board or by tacking components together. Then the parts are placed on a printed circuit board, leaving space for expansion, power supply, and the like.

In the past printed circuit board layout was a tedious job, involving the rubbing down of transfers onto transparent film. These days things are much easier: using a suitable software package, the tracks and pins are simply sketched on the computer and the layout can subsequently be modified and processed as desired. The layout program will allow the results to be output in many different ways: frequently the layout is simply printed out on paper,

but for higher quality (and at greater expense) the results can be photoplotted onto high-resolution film at a reprographics bureau.

The light-sensitive surface of a suitable piece of copper-clad board can now be exposed and developed. This removes the light-sensitive resist coating in those areas where the copper is to be etched away. At this point, for most electronics hobbyists and prototype builders, the nightmare begins. To develop and remove the unwanted copper chemical processes are used which employ various unpleasant chemicals. We surely do not need to explain to *Elektor Electronics* readers that the properties of the etchant are such that not only the copper on the circuit board, but also hands, tables, chairs and clothing — all can get eaten away. Some advice: if you use a colourless etchant, you only notice a splash when it is too late, and you find a hole in your clothing; use a coloured etchant and you can see the impending disaster before your clothes are ruined. A brown iron (III) chloride (ferric chloride) stain is easy to spot and can be neutralised with sodium hydroxide (caustic soda), before using a specialist stain remover.

Etching need no longer be carried out, as it was in the past, in open trays: closed etchant-spray tanks are readily available, and

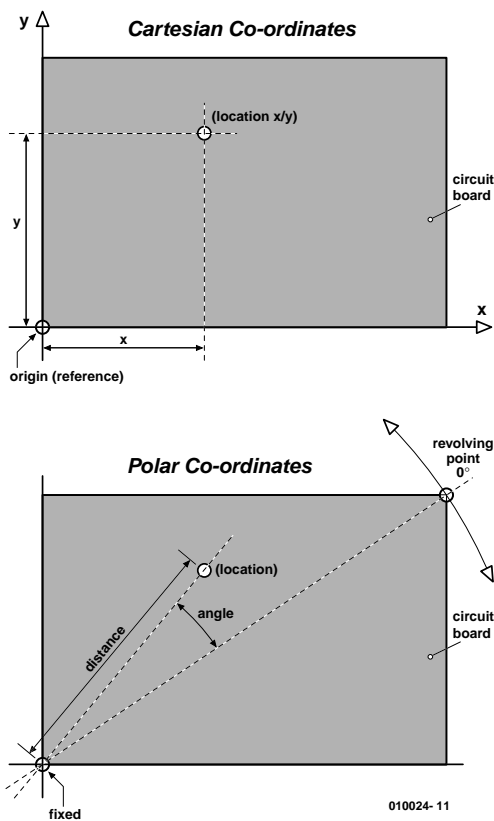


Figure 1. Cartesian and polar coordinates.

affordable. Closed systems keep the unpleasant chemicals to a defined area.

Do not forget that any remaining etchant must be washed from the newly-etched board first with water and then any still clinging to the board can be treated with developer.

Caustic soda is cheap and in any case does not keep well. Plain water will not do the job: it does not clean the board properly. After drying, the copper side should be sprayed with a lacquer to make soldering easier and to prevent the copper from oxidising.

A Boring Story

Next, holes are bored in the circuit board using drills of various diameters. In general, three drills are used: the commonest is 0.7 mm or 0.8 mm, used for almost all small components such as resistors, capacitors, ICs; it is even suitable for the leads of some electrolytic capacitors. A 0.9 mm or 1 mm drill is used for most connectors, square pins and the larger diodes. Larger connectors, solder pins and similar components require a 1.2 mm or 1.5 mm drill.

Usually a mini-drill is used in a simple pillar stand. Great concentration is required to ensure that the holes are drilled centrally in

the pads, so that the holes line up in straight rows and columns. Otherwise, fitting components such as D-connectors and 40-pin ICs will present a major problem.

Larger holes, perhaps 3 mm for mounting bolts or screws, are best not drilled with the mini-drill. More powerful machines, however, tend to pull away at the circuit board material, which gives an unsatisfactory appearance to the holes. It is better to drill first with a 1 mm drill and then use a hand reamer.

High speed steel (HSS) drills can be used for all except fibreglass boards. Hard metal drills are not suitable for fibreglass because the glass embedded in the plastic wears away the cutting edges of the drill so fast that it struggles through the material and does not produce a clean hole. Hard metal drills have a 3 mm or 1/8" thick spindle and a small cutter. These drills, costing upwards of two pounds each, are brittle and break at the slightest provocation. It is an expensive way to make holes.

Drilling under Software Control

The best thing about PCB layout programs is that they automatically generate lists of coordinates of points where holes are to be drilled. If a computer-controlled drilling machine is used, the circuit board simply needs to be fixed to the table and the machine does the rest. The only intervention required is to change drills, which can however be quite tiresome. The program must be stopped, the drilling head must be moved to a specific position, or at least raised away from the board, in order to gain access to the chuck. The height of the drill must be correct, to ensure that the new bit is at the same height as the old one: for this purpose rings can be marked on the drill spindles. Alternatively, drills ready-marked with rings can be bought — but all from the same manufacturer, since different manufacturers put their rings at different heights! And if, after all this effort, the new drill is only used for a few holes, the whole process will have to be repeated just a few seconds later.

Manufacturers soon realised that

this was a problem, and developed chucks which can be opened from above using a small handle. This is a great improvement over chucks, such as those frequently found on milling machines and mini-drills, which require a special tool to open them.

Such chucks make a rather expensive addition to a small drilling machine, but for the perfect solution you can expect to pay as much as a thousand pounds: you will need an automatic compressed-air tool changer with tool magazine and a compressor, and everything will work as if by magic. The control program will run a little slower, because the drilling head has to move over to the tool magazine to drop off the old drill and pick up the new one; even here, comfort has its price.

But there is still a big step from the perfect machine to a satisfactorily drilled circuit board. Have you ever considered how registration is preserved through the process of circuit board manufacture? In the PCB layout program the drilling coordinates are known exactly. The positions appear on the film and are then transferred to the circuit board — but where is the reference point?

Getting the film cleanly and accurately aligned with the edges of the circuit board is made impossible by the frequently dirty and roughly-cut base material. The circuit board is generally not rectangular, the material being cut after production using a guillotine.

A computer-controlled drilling machine requires a device that guarantees that the drill lands repeatably at the specified point. This device should be firmly fixed to the machine and in any case designed to make precise alignment easy.

There are many approaches to registration, using try-squares, pre-drilled holes (a real problem when modifying a circuit), optical registration devices, sticking the film to the circuit board, or 'intelligent' motion analysis systems using a camera, TeachIN and coordinate transformations according to reference marks, and many other wonderful methods.

In twenty years of experience developing printed circuit boards in small quantities, the author has found a much simpler solution:

Polar Coordinates!

Here a point on a surface has its position defined not by X- and Y-coordinates as in the Cartesian system, but by a length (distance from a fixed point) and an angle. Polar and Cartesian coordinates can be inter-converted without loss of information.

In the PCB layout, draw a circular pad with diameter exactly 3 mm in an unused area; alternatively, use a mounting hole. In the circuit board itself, before exposure, drill a hole with diameter 3.1 mm, in the corresponding place to the pad on the layout. Position the film over the board so that the pad is over the hole. The 0.05 mm wide crack of light around the pad allows exact centring by eye. The centre of the pad is called the reference point. If the unexposed board already has the right dimensions, the film carrying the printed layout can be rotated until its edges are also in alignment with those of the board.

In the layout, as far as possible from the reference point, a second 3 mm diameter pad is drawn. The greatest distance is across the diagonal of the circuit board, but if there is no suitable space, a different place can be chosen. The important thing is that it is far from the reference point. This second point is called the rotation point. Cartesian and polar coordinates are compared in **Figure 1**.

After exposure and etching of the circuit board the reference point will have been etched away, because of the hole previously drilled there. Each other hole to be drilled is at a known distance from the reference hole. This gives us one of the polar coordinates. All we need now is the angle, which is derived from the rotation point. At the rotation point, drill another 3.1 mm diameter hole. This hole must be made extremely accurately, because the position of all the other holes depends on it. It is best to drill first a 1 mm hole and enlarge with a small round file or reamer. The copper of the pad gives a good indication of how well centred the hole is.

All we require now on the table of the computer-controlled drilling machine is a pair of small pegs that

Cartesian coordinates: a heresy

Has anyone ever seriously wondered why CNC machines always work using the Cartesian coordinate system? Why are X-, Y- and Z-axes always used? Why, when such machines are so difficult to build? The linear guides must be absolutely parallel, because otherwise the carriage will jam. The axes must be at exactly 90 degrees to one another, or else everything goes askew. The table must be absolutely true and the whole machine must be solidly fixed to a base.

These are all disadvantages. But the greatest disadvantage is the linkage between the axes. Consider the X-Y table, the original form of hand-operated milling machine. This has two handwheels, one to move the table in the X-direction, the other to move the table in the Y-direction. There are thus two linear guides, one fixed at an angle of 90 degrees to the other, supporting one another. If the lower mechanism has play in it, this is transferred to the upper one, even if the upper one is absolutely precise. And the lower guide also has to bear the weight of the upper one.

This traditional mechanism stems from a time before computers, when positioning along the axes was controlled manually by a technician using handwheels. Technical drawings are normally marked up with XY coordinates so that the successively required positions can easily be reached by use of the handwheels. In the age of automation the technician is no longer employed and the handwheels are replaced by motors under computer control. But the coordinate system has not changed: in the human imagination everything has a length, a breadth and a height. Curious, when most human actions are polar: 'take three steps in this direction and then turn right'!

Imagine now how you would drill a circuit board by hand without a pillar drill. With the fingers of one hand you would hold the board steady and with the other hand you would hold the drill. Your drilling arm would be rather higher at the elbow than the other arm, since the mini-drill has a certain height. But you do not move your arms in the X- and Y-directions — no, you turn your drilling arm about the pivot at the elbow and turn and slide the circuit board to suit. You optimise your movements using your visual system — not perfectly, however, as sometimes you might miss a hole hidden by swarf. You do not need a firmly fixed base on which to work; your drilling arm is fixed at the elbow pivot, and what is between this pivot and the circuit board does not matter. Even a small tool between the two makes no difference. Your arms and your sitting position need not be absolutely parallel, or even anywhere near, and there are no 90 degree angles to be seen: two pivots are enough!

fit smoothly and without play in the two holes we have drilled. The peg for the reference point is fixed, while the peg for the rotation point can be slid along a line allowing for various distances between the two points. The coordinates of the fixed peg are known, as is the angle of the sliding rotation-point peg. The drilling data comprise the X- and Y-coordinates of the two points, and so the distance between the two and the angle they make with the Cartesian axes can be calculated using a simple program on the PC. After a translation and a rotation the position of all the other points can be determined, no matter how askew the film was placed on the circuit board during exposure or how the board lies on the CNC table.

This method of registration is suit-

able for the manufacture of one-off circuit boards, or for a number of different boards. For small-quantity production runs a slightly different procedure can be followed.

The Concept

The system uses two pivots, one for the workpiece (the circuit board) and one for the drilling arm. This allows any desired point on the circuit board to be brought into range on the turntable. This system has the big advantage over a linear construction that only two bearing points are needed whose exact separation is the only quantity that needs to be known. This requires no expensive specialist components: the bearings simply have to remain vertical and free of play. To align an axle to professional standards of accuracy, two so-called taper bearings are used. These can withstand enormous forces and are

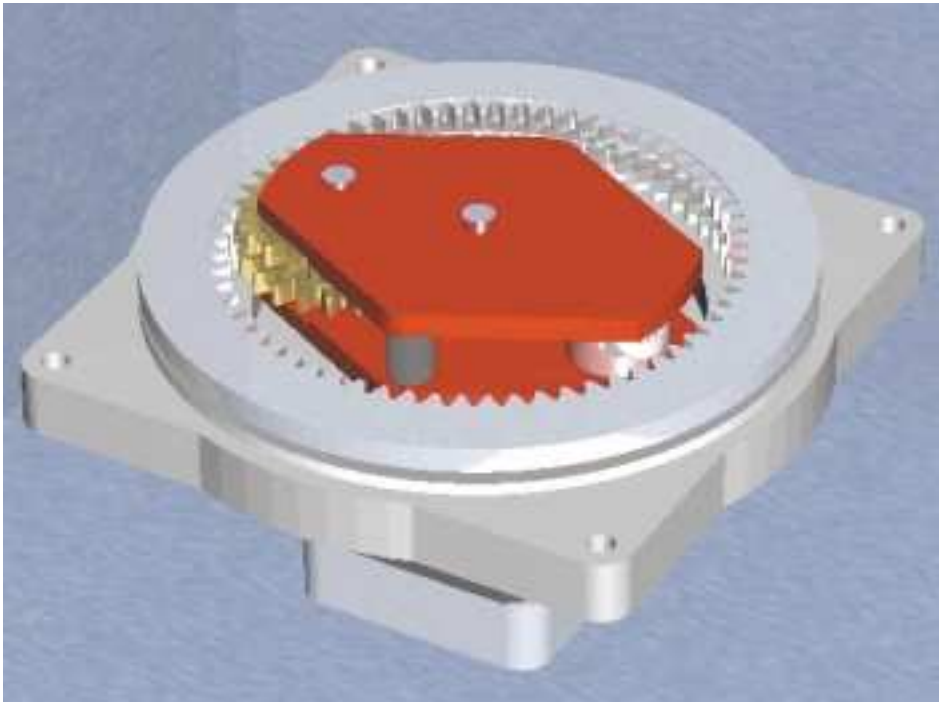


Figure 2. The well-known principle of the 'axle drive'.

expertly made to remain solid and permanently free of play. This is the main reason why our machine is so economical to build.

A rather significant disadvantage ought to be pointed out. Normally, in conventional linear machines the axles are long threaded rods supported by bearings and turned by a motor. A nut is fixed to the moving part and moves backwards or forwards according to the direction of rotation of the screw. This mechanism naturally gives a high effective gear ratio. Suppose that the screw gives a linear movement of 4 mm per rotation and is driven by a stepper motor with an angular resolution of 200 steps per revolution. The linear motion corresponding to one step is $4/200 = 0.02$ mm. That is ideal for this kind of machine. Gearing is therefore completely unnecessary.

Our machine is not driven by a threaded rod, but rather works directly with angular motion. The 240 mm long tool arm has a travel of

$$240 \text{ mm} \times 2 \times 3.14 = 1510 \text{ mm}$$

(circumference of circle with arm length as radius)

A stepper motor driving this directly would take 200 steps to make one revolution; the distance corresponding to one step is therefore $1510 \text{ mm}/200 = 7.55$ mm, rather too great for a CNC machine. For a desired resolution of less than 0.04 mm we need to gear the motor down by a factor of at least $7.55 \text{ mm}/0.04 \text{ mm} = 190:1$. That is not exactly straightforward.

Gearing Mechanism

In the opinion of experts in the field there is no simple gearing mechanism that gives such a high ratio apart from a worm drive. It is mathematically impossible to achieve the ratio with two or three gears without using an unreasonably large number of teeth. At least three stages of gearing down are required with many individual arbors that must each have good quality bearings.

This difficulty, and the unreasonably large amount of play in such a mechanism (and hence problems with repeatability), would cancel out all the advantages of our idea. But you would not have this article in your hands if the problem could not be overcome even despite the firm belief of the experts; and without employing such arcane devices as flexible components fitted between gears with intricately-cut teeth, ball-bearings moving in contorted orbits on even more contorted guides, or belts with different patterns of teeth on either side. If you want to know more, there is plenty to read on this subject on the Internet: you will find ideas there considerably more curious than our drive system. We achieve the desired gearing ratio using four gears. The principle has been known for a very long time,

and there is even a VDI (German Association of Engineers) document on the subject (see box); however, practically no-one has yet realised the potential of the idea.

Our drive uses the principle of subtraction. Imagine a travelator such as those found in airports. Our travelator is circular and runs with constant speed. You are running on the travelator in the opposite direction to its rotation; the nearer your speed to that of the travelator, the lower your speed in relation to a fixed point. If you match the speed of the travelator exactly, your net speed is zero: you have a maximum 'gearing ratio', the two speeds cancelling one another out.

The situation is similar in our drive system. A spur turns within an annular gear with internal teeth: the actual value of the gearing ratio is not important. Fixed to it is a second spur gear, which turns in a second annular gear. Note that the two spur gears are fixed to the same axle. The two combinations of two gears — i.e., the ratios of spur gear to annual gear — are slightly different from one another, but the more similar they are the greater the overall gearing ratio obtained. Since the second spur gear turns at the rate determined by the first stage rather than that determined by the second, the second annular gear must make a compensating motion. In fact, we subtract the two speeds, just as in the travelator example. This explanation may seem rather complicated, but the illustration in **Figure 2** should make matters clear.

Normally more teeth are required to achieve a higher gearing ratio. Here, however, that is not the case, because it is merely the difference between the two ratios that matters. The author calls this drive system 'axle drive' and a patent has been applied for. The chances of success look slim, however, since the design has been known for a long time.

A glance at the internals of the drive system in **Figure 3** immediately suggests many possibilities. For example, there is enough space in the housing to include the parts of a motor; then one would have an incredibly small unit which would be not just a gearbox but a motor with a slowly-turning output shaft with a

very high torque. One need only think of the countless applications in vehicles, such as for electric mirror and seat adjusters, or for electric windows, where a flat drive system is required to fit between the door panels. In robots; in component handling systems; this drive is ideal any-

where where a slow but powerful motion is required.

The Drive in Action

Our design not only has constructional advantages over normal linear systems, but also works better in

many ways. What happens when the drilling head of a linear machine is at one end of its travel and must move to the opposite end? The head must be moved the entire distance. In our design, things are circular: define 360° as one 'end' of the travel and 0° as the other, and the two are the same! If we are at 360° and need to move to 10° , we do not need to

The drive system: how it works

When we started looking for a simple design for the drive system to give precisely the high gearing ratio required for our machine, we were met by shaking heads: the experts told us that high ratios could only be obtained by using multiple separate stages, each with a small ratio, multiplying together to give the total desired.

Using an ordinary planetary drive a practically-realizable gearing stage with a ratio of up to about 7:1 can be built. This is due to the fact that multiple spur wheels must turn within the annular gear and must clear one another. The diameter of the spur wheels must be less than half that of the annular gear to allow clearance, but must of course not be too small of they will have too few teeth to run smoothly. If higher ratios are required, multiple stages can be cascaded: with two stages we can obtain a ratio of $7 \times 7:1$, or 49:1. Add another stage and we have at most 343:1 (although for constructional reasons the maximum obtainable is only just over 300:1). Each stage takes up space, and a three-stage drive with a diameter of 40 mm with attached motor might easily have a total length of over 100 mm.

Further, each stage brings with it play in the mechanism and angular error, which all add together. Economical three-stage planetary drives have a play of about 3 degrees in the output shaft. Although that may sound small, in our design it would translate into an unusable 12 mm of error at the drilling head.

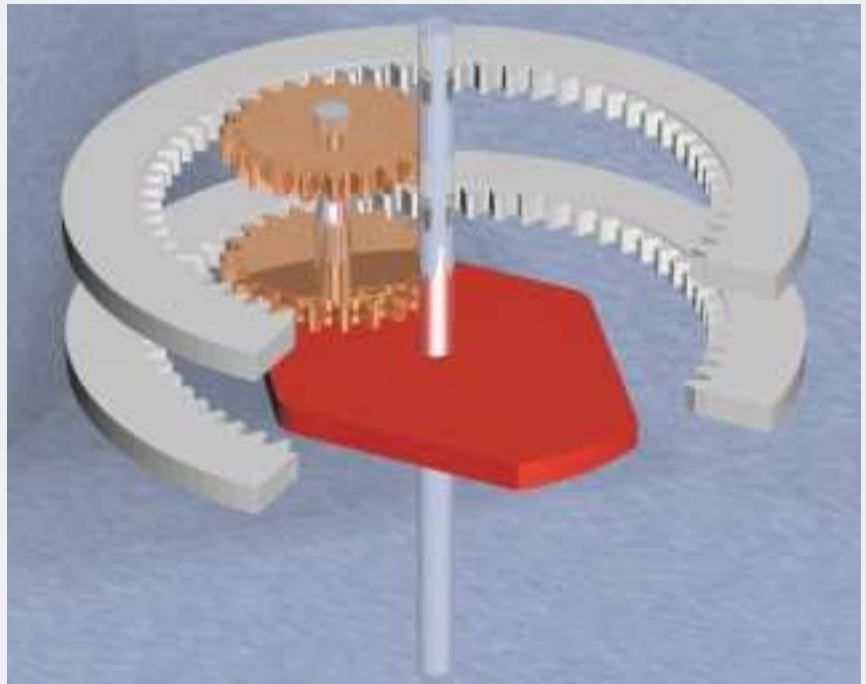
Precision drives in the middle price-bracket — say around one hundred pounds — reduce this play down to just below one degree. For our application, this is unfortunately still completely unusable.

Take a look at the figure and imagine the following sequence of motions.

We will use different tooth counts from those shown in the figure. Suppose the lower annular gear has 45 teeth, its spur wheel has 10 teeth, the upper annular gear has 50 teeth and its spur wheel has 11. These values will provide an excellent example of how the drive works.

The lower annular gear is fixed, and the long main shaft, shown passing through, is free to turn. This is the motor's output shaft, which turns the plate on which the small spur wheels are mounted.

The spur wheel is free to turn on bearings in the plate, and if the main shaft is turned then as the spur wheel runs around the inside of the annular gear, it turns on its own shaft. If the main shaft is turned through 360° , then the spur wheel will have turned on its own axis through 4.5 revolutions since its gearing ratio is



45:10, or 4.5:1. Although the two spur wheels are free to turn relative to the plate they are fixed to one another. So, when the lower spur wheel turns through 4.5 revolutions, the upper one must do so too.

Now if the upper spur wheel, with 11 teeth, turns through 4.5 revolutions, that equates to 49.5 teeth. But the upper spur wheel turning through 49.5 teeth in the upper annular gear causes a problem: the annular gear must make a compensating motion, and the size of this motion must be 50 minus 49.5 teeth, or exactly half a tooth.

Since the upper annular gear, with 50 teeth, moves by a half tooth for each revolution of the main shaft, this means that 100 revolutions of the main shaft are required to obtain an entire revolution. With this combination of 45:10 and 11:50 ratios we have therefore obtained, in one stage, a gearing ratio of 100:1. In our application we use ratios of 60:22 and 23:63, and with no other modifications to the construction obtain a gearing ratio of 231:1. The diagrams show the construction to scale and with exact teeth counts.

In the 19-page VDI (German Association of Engineers) guideline document number 2157 of September 1978, which deals with planetary drives, basic constructional rules and calculation methods are given. The exact arrangement dealt with in that document is called a 'simple planetary drive' which for technical reasons is not suitable for higher gearing ratios.

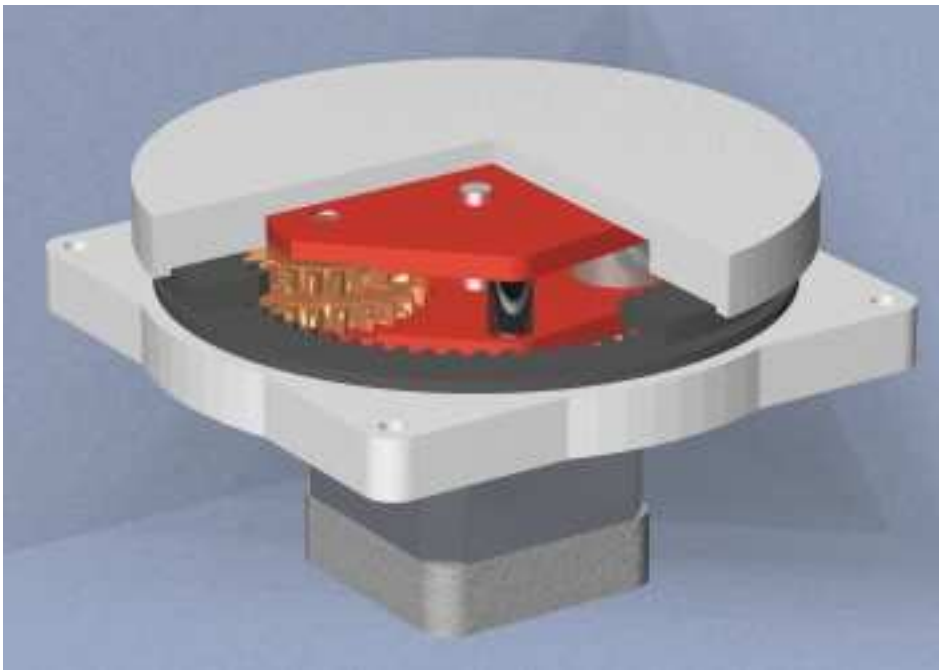


Figure 3. The 'axle drive' as used in our machine.

move through 350°, but only through 10°. For this reason our system is four times faster than a conventional CNC machine working over the same area. Also, the mass moved in our system is much less, which allows the machine to work faster still.

The construction of the machine, as can be seen in the photographs (Figure 4), aimed a

maximum stiffness and minimum weight. The arm and the table can be moved at 70 mm per second, and the stepper motors only require a simple drive circuit because they draw less than 1 A.

Naturally, our machine suffers from the problem of changing the

drill bit in the same way as conventional machines as described above. Our design does not provide for automatic tool changing. The construction of a suitable arm would be too complicated and elaborate. But here again we solve this problem in an elegant and surprising way that could not be considered for a conventional machine. There is plenty of space on the drilling table, so we simply add a second arm with a second drilling head. It would of course be possible for the two arms to foul one another, but we can easily arrange things so that one moves out of the way when the other approaches.

The machine is now even faster, since it does not need to wait idly while the tool is changed, does not need to move to a dedicated location to change the tool, and — the clincher — can work with the two arms simultaneously. And further, in order to work with three different drill sizes as is typically required (see above) we can add a third arm. This third arm is, however, not part of the basic design as seen in the cover picture and requires additional parts. The additional bearing housing can be simply screwed to the baseplate. Using three drills simultaneously in a conventional CNC machine would be out of the question.

It is possible to add a fourth arm. Since three drill sizes normally suffice for printed circuit board work, the fourth arm could perhaps carry a different tool such as a router for cutting around the edge of the board. However, the mechanics of the drilling arm are optimised for drilling, and the large sideways forces encountered during milling and routing will adversely affect its accuracy. We will leave the reader in suspense as to whether the author will solve the many problems associated with routing!

(010024-1)

In the next instalment we will be going into more detail concerning the control of the drilling machine from a PC via a microcontroller with an output drive circuit. The command transmission protocol for the microcontroller includes a couple of rather neat ideas!

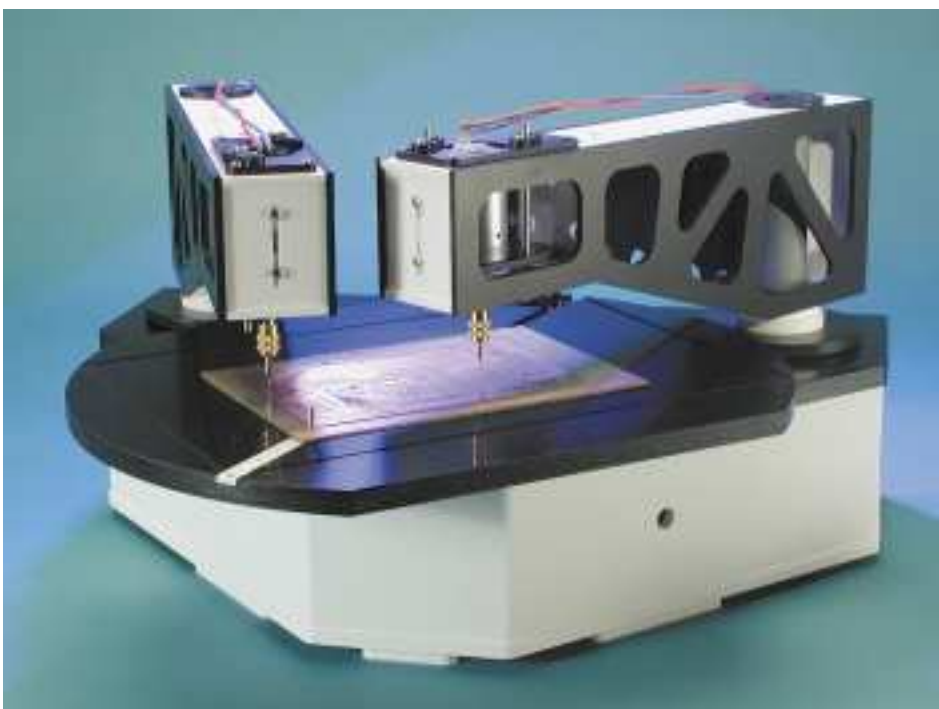


Figure 4. The drilling machine at a glance.

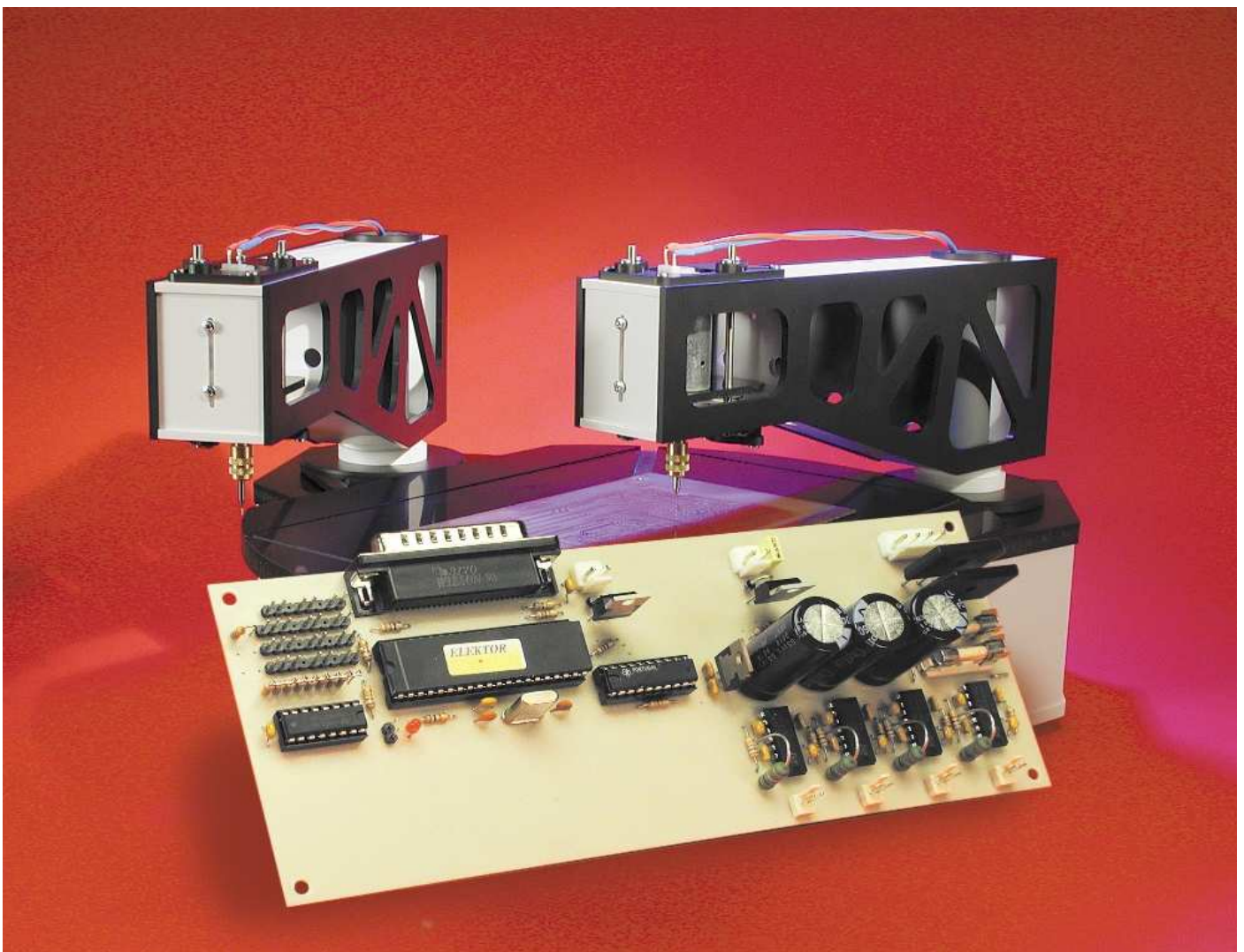
PCB Drilling Machine (2)

Part 2: The controller

Design by T. Müller (Radix GmbH)

www.radixgmbh.de

Our PCB drilling machine provides a link between a Windows PC and the motors and solenoids of the machine itself: a microcontroller integrated with a high current drive circuit.



In the previous instalment (March 2001) we described in detail the basic principles of the CNC drilling machine. The controller is just as unconventional: the machine is not driven directly from a PC, but indirectly via a special controller circuit. For trouble-free communication between the Windows PC and the controller over the Centronics interface we need to make use of several tricks that take us beyond the standard Centronics protocol.

Let's start by taking a look at the rather straightforward hardware of the controller board. The basic circuit consists on one side of a microcontroller with a PC interface and on the other the output drive circuits, connected to I/O-port pins. Of course there is rather more to the design than that.

Real-time with Windows?

How can we control a stepper motor from a PC? It goes without saying that we need an output drive circuit capable of supplying the current required by the motor, since none of the PC's interfaces can drive a motor directly. And then, in order to make the motor step smoothly, we need very accurate timing.

At first blush it would be ideal if we could somehow connect the output stages directly to the PC and handle practically all the control signals directly as bits from the PC. The circuit would then be very simple, consisting of just a couple of latches and an address decoder.

Unfortunately, Windows has a rather unpleasant characteristic: its operations are totally asynchronous. It is impossible to say at exactly what time an event will occur. Approximate timing is possible, but as the specified time between events gets smaller and more precise, something eventually will go wrong. Windows on its own was not designed for control applications, and indeed is not suitable for them. It may be easy to use, but from the hardware point of view it is no more suitable than older operating systems such as DOS.

First of all, modern programming languages offer hardly any commands for access to the I/O-ports; and second, multitasking prevents

A universal controller board

The controller board, with its power output stages and integrated microcontroller, has been designed not just for use with the PCB drilling machine, but also with enough flexibility to be used for many other purposes. The driver stages make it a very handy controller board for other equipment or machines that employ stepper motors and other high current devices. For this reason there are a few places in the layout where we have made additions that are not strictly necessary for the drilling machine application.

To take one example, the output stages are controlled by a GAL. The GAL is simply programmed to combine the PWM signal with a two-bit address to activate the solenoid drive, and (with an enable input) the drill motors. This function could easily have been carried out using a simple address decoder such as a 74138. However, by using a GAL we can completely reconfigure the eight output stages. To this end, you will see that we have routed extra signals from the microcontroller to the GAL, in particular signals that can be assigned special functions in the microcontroller.

Series resistors and capacitors can be fitted at the inputs to the driver stages. The power MOSFETs used in the drilling machine controller do not require these; but if you wish to use bipolar transistors in the drive stages, the spaces we have left on the board will come in handy.

direct hardware access by programs that attempt to bypass the operating system. Special drivers are required, such as `port.dll` found in the *Elektor Electronics* book 'PC Ports under Windows' (to be published soon).

Let us suppose that you have available a programming language that lets you control the port signals at will. We wish to generate a continuous square wave on one of the data pins of the Centronics port at a given frequency. In software this is a trivial matter: we simply need to toggle the bit in question regularly. In order to obtain the desired frequency, the appropriate delay must be used. If the delay is reasonably large, say 0.5 s, we obtain a perfect output signal with a frequency of 1 Hz.

If, however, we reduce the delay to say 0.5 ms and hence raise the frequency to 1 kHz, we hit a problem. The right frequency appears at the output, but only when averaged over a long period. The individual pulse lengths are a little longer or shorter in a random pattern, and some are very different from the specified period. Why is this?

Of course, modern processors are fast enough to toggle a bit 2,000 times per second. No, the reason is that Windows, as a multitasking

operating system, has other things to do while generating our output waveform. Updating the mouse position, checking for incoming e-mail, reading audio data from a CD, communicating with peripherals: all these take processor time.

Windows is event-driven, which means that the various parts of the machine can cause the operating system to carry out operations. This can be seen when printing, for example: Windows launches a process to communicate with the printer, and supplies it with new data as and when it is ready to accept it. When that happens and how much data is transferred at a time is totally undefined, but nevertheless consumes processor time. This can interfere with the timing of our square wave and cause variations in its period.

There are two ways to overcome this problem. We could shut down all other processes and disable multitasking while we generate our square wave. Then we might as well use DOS, which is what most manufacturers of this type of control software for small machines do. Under DOS, an application program can use the full power of the processor for itself. Of course, we sacrifice the ease of use of Windows, and, in spite of its power, the computer cannot be used for anything else.

There are various operating system additions available for Windows which ameliorate this situation and provide quasi-real-time facilities. These are mostly VXD or TSR pro-

grams that are loaded when Windows starts up and then run in the background. These programs run briefly, but at an exact moment in time; and they run at the highest priority to guarantee having the necessary processor power available to them.

These programs are rather critical and considerably increase the chances of Windows crashing. They also permanently consume memory and processor time, because they are always present and must always be ready to spring into action when required.

These programs are also often written using undocumented features or 'back doors' in the operating system, and so are usually only fully compatible with one particular version of the system.

Offloading the synchronisation problem

We have therefore chosen an alternative approach to driving a stepper motor under Windows. Synchronisation is not left up to Windows software, but rather transferred to a microcontroller. The microcontroller receives a control command (for example for a motor pulse), and rather than executing it immediately, waits for further timing information to indicate when the data are to be processed. In this way, as long as the overall system is fast enough, we can obtain perfectly synchronised results using Windows.

The information passed to the microcontroller to generate a square wave runs as follows: bit high in 500 μ s, bit low in 500 μ s, Repeat. This information is sufficient for the microcontroller to produce the signal independently, requiring no further intervention from the PC. The PC must deliver the data fast enough, each command arriving within 500 μ s. This should not present any difficulty for a reasonably fast PC.

To prevent the PC having to wait until the microcontroller is ready for a new command, the controller is equipped with a small FIFO memory with space for 40 entries. This allows the controller to be well decoupled from the PC. The PC can send off 40 precalculated commands for switching the various output signals to the controller and then has 40 \times 500 μ s to prepare new data or carry out other tasks.

Shortly before the FIFO empties, this state is flagged to the PC, which can then refill the FIFO with new data. This does not affect the controller, which continues executing commands one after another with perfect timing.

Using the Centronics interface

Communication with the controller could be

carried out over any PC interface. The Centronics and V24 (serial) interfaces are always available, and are particularly suitable candidates.

There are certain problems associated with the V24 interface: getting the wiring right and arranging handshaking can be full of pitfalls. Matters are also more complicated for the microcontroller, which must first convert the serial bit stream into parallel bytes. And, if the microcontroller does not contain a UART, the controller must detect and interpret the start bits itself. This consumes processor time in the controller and ultimately creates difficulties in ensuring timely execution of commands.

For these reasons, we have chosen the Centronics interface. This interface is also superior from the point of view of speed, because eight bits are transferred at a time: this also makes the data easy to process. The data are simply written to a specific I/O address and the job is done. It is also convenient for the microcontroller to receive the data in a simple 8-bit format.

Eight bits may be a convenient number, but it is too few to specify a command with timing information. Multiple bytes must be assembled together to construct a useful command so that the controller can know what to do with the data. For our application, two bytes are enough for each command, but that is too much to transfer in one go over the Centronics interface. But when the 16 bits are divided into two independent bytes, a synchronisation problem can arise.

The strobe-edge trick

How can the controller know which of the bytes is the first and which the second? Counting is one option, but not a very reliable one: a single transmission error and all successive commands will be wrongly interpreted. The error must be detected and the controller reset. Alternatively, the data bytes could include further information to allow the controller to identify which is which.

However, this also has its disadvantages. The marker bits naturally reduce the amount of real information transferred, and so we have

fewer bits available for commands.

For this reason we distinguish the bytes externally using a means already provided within the Centronics interface. Take a look at the protocol as shown in **Figure 1**. First consider normal operation, described at the top of the figure. The interface, however, will let us transfer two bytes at a time without modification. How that is achieved is described at the bottom of the figure.

The handshaking process is unchanged: we have simply changed the meaning of the signals. Two bytes are coalesced into a single packet, where the falling edge of the STROBE signal marks the first byte, and the rising edge, the second.

If an error occurs when the first byte is already in the process of being sent, STROBE will eventually rise, if only because of the pull-up resistor fitted to the controller board. The transfer is then terminated, and the pair of bytes remains together. Synchronisation problems are therefore impossible, even though the contents of the two bytes will be in error. If on the other hand we had simply counted bytes, then all subsequent bytes would be interpreted wrongly, as they would all be interchanged. If we were to continue in this manner, the controller would cease operating correctly hence the system would have to be re-initialised.

At start-up the control software sends a RESET command, which causes the controller to clear its FIFOs and bring to a halt all movements in progress in the system.

Controller functions

We have now described the main functions of the controller: accepting the 16-bit wide instructions, storing them in the FIFO, and synchronously executing the commands stored in the FIFO. Of course, the calculation of trajectories is not handled by the microcontroller, since it is not designed for trigonometric calculations and would be too slow. It would also imply programming the dynamic characteristics of the system once and for all into the microcontroller.

Instead, the movements are pre-

processed in the PC and then sent to the controller in the form of primitive instructions such as *motor 1 one step counterclockwise*. All movements are defined in the PC in this way. It is possible to compute the step information for any desired motion and then send it to the controller; and thus the microcontroller never needs to be reprogrammed.

Data format

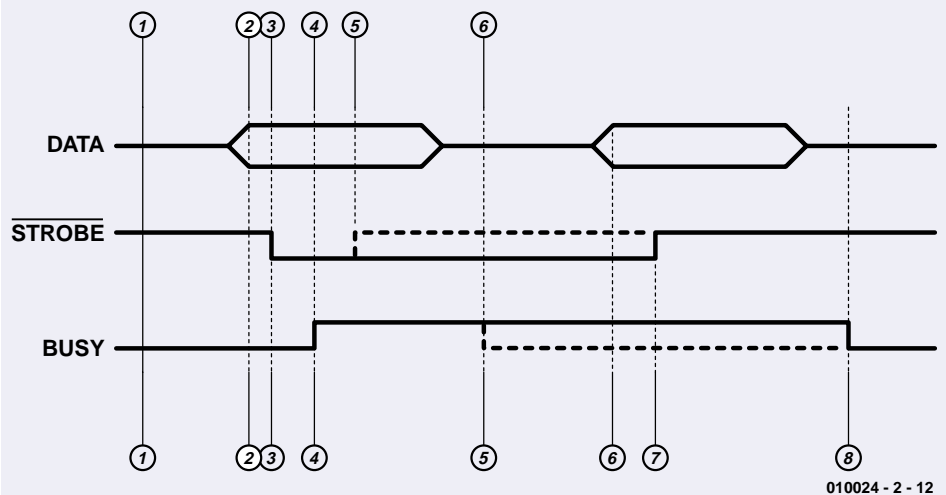
Now that we have explained how the command data are calculated in the PC and transferred over the Centronics interface to be synchronously processed by the controller, we can take a look at to the internal structure of the command data.

The two bytes are concatenated to form a word. This word contains a data value, or operand, along with an address which determines how the operand is interpreted: in other words, the particular command to be executed. The number of address and data bit is variable, the number of data bits shrinking as more address bits are required. A command with a short address field can contain a longer operand, and vice versa. This kind of addressing scheme can be processed using a sequence of branches, decoding the command using the 'divide and conquer' principle.

This works as follows: the first address bit determines whether we are dealing with one particular type of command (when the bit is a '1') or not (when it is '0'). In the second case we still do not know what type of command we are processing, so we examine the second bit. If this is a '1', then we have decoded a second class of commands, while if it is '0', we proceed to examine the third bit, and so on. Of course, as we continue to examine bits in this way, there remains less and less space for the operand data.

The advantage is that we only need one address bit for the first command type, and so we have a full 15 bits available for the operand data. For commands whose address field is, say, three bits long we have 13 bits available. Note that a three-bit address field allows at most eight different commands to be separately decoded.

1. Quiescent condition:
STROBE from PC to microcontroller is high. BUSY from microcontroller to PC is low.
2. The PC wishes to transfer data as usual, and puts the data byte on the eight data lines.
3. The PC takes STROBE low. The connected microcontroller (or peripheral device) sees the low state and knows that the data are ready to be read.
4. The microcontroller takes the data byte and sets BUSY high in acknowledgement.
5. The PC takes the STROBE signal high again.
6. When the microcontroller has processed the byte, it indicates this state by taking the BUSY signal low. This completes the transfer of a byte. The system is ready to transfer another byte, the control signals (BUSY and STROBE) being once more in their original states.



010024 - 2 - 12

1. Quiescent condition:
STROBE from PC to microcontroller is high. BUSY from microcontroller to PC is low.
2. The PC wishes to transfer data using the special mode, and puts the data byte on the eight data lines.
3. The PC takes STROBE low. The microcontroller sees the low state and knows that the data are ready to be read.
4. The microcontroller now takes the data byte and sets BUSY high in acknowledgement.
5. The microcontroller processes the byte immediately and leaves the BUSY signal high.
6. The PC observes that the BUSY signal is high and therefore that the first byte has been accepted. It then puts the second byte on the data lines.
7. The PC indicates that the second byte is present by taking STROBE high again. This causes the microcontroller to accept this byte also.
8. When the microcontroller has finished its processing, it sets BUSY low again.

Figure 1. The original Centronics protocol (above) and as modified for the drilling machine (below).

The variable-length address field also allows a higher priority to be assigned to more common or more urgent commands. For a rarely used command, or one that initiates a lengthy action, we can allow the address decoding to take a little longer.

The commands which are most commonly used, and which sent to the controller at the fastest rate, are without doubt the stepper motor commands; they also require a long operand, which is provided by the highest priority command format. The format of the commands for sending stepping pulses to the motors is set out in **Table 1**.

The second command class is the drilling command. Three address bits can be used for this command without causing any difficulty

because its execution time is much longer than the motor step commands, and because it requires a shorter operand. The meaning of the various bits in the drilling command is set out in **Table 2**.

These two commands suffice for basic operation of the PCB drilling machine. In fact there is a range of other commands which are less relevant to our discussion and which are not described here.

Brawn and brains

At first glance the circuit diagram in **Figure 2** may look like an impenetrably

complex piece of electronics, but on closer inspection we see that the brains of the circuit are actually very simple. The intelligence is concentrated in a type PIC16C64 microcontroller from Microchip, clocked at 20 MHz. This microcontroller boasts the magnificent total of 33 individually addressable I/O-ports, but only 2 K of EPROM program memory and 128 bytes of internal SRAM, here used for (among other things) the FIFO memory. The PIC16C64 is equipped with peripherals such as a real-time clock, timer/counter and capture/compare inputs. In this application we do not use the 3-wire

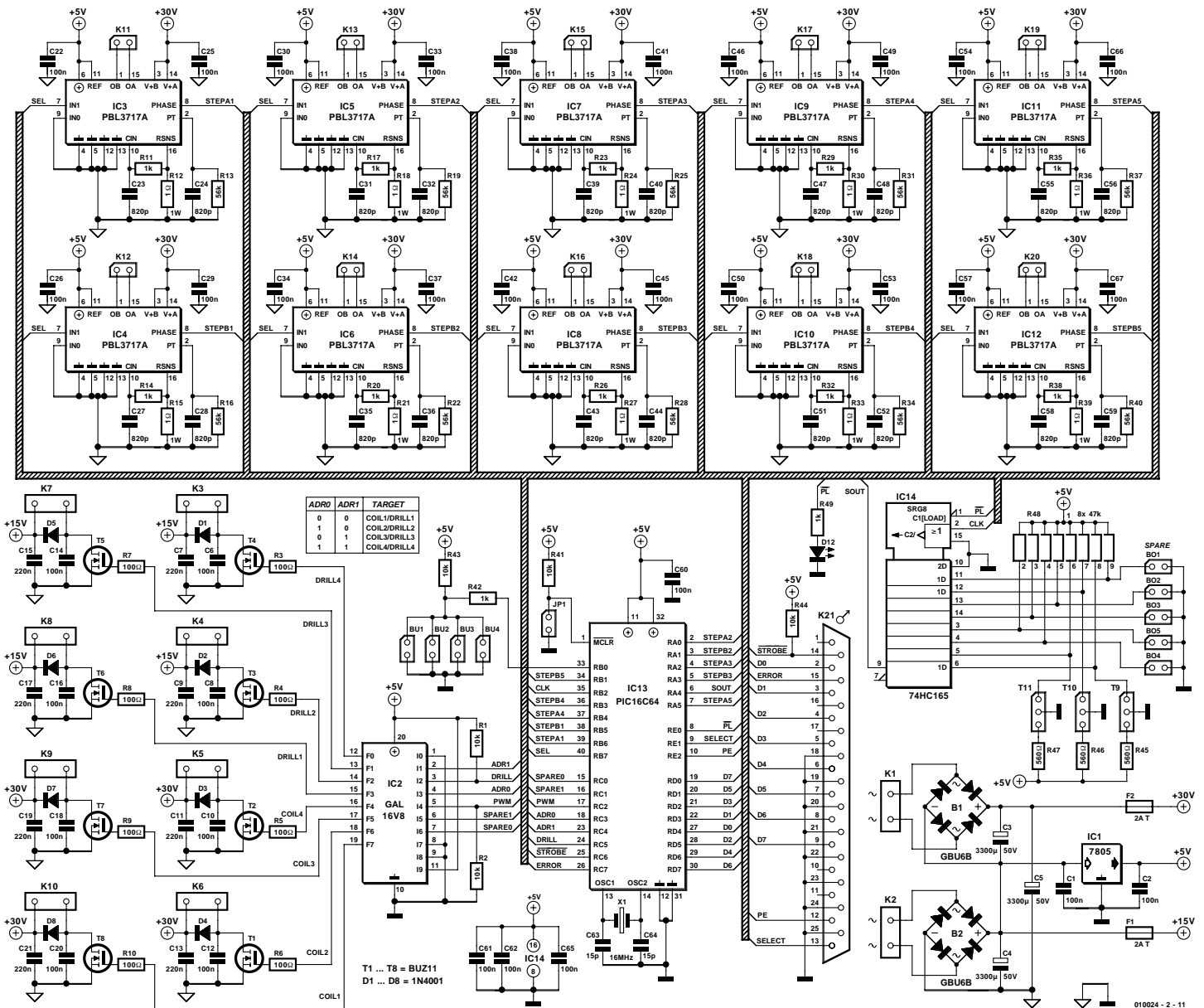


Figure 2. The drilling machine controller board: lots of brawn, and a little bit of brains!

Table 1: Motor pulse command

MSB	Byte1	Byte2	LSB
1	Z Z Z	D T T T	Z Z Z Z

Addressing: bit 7, byte 1 = '1'.

Z (11 bits): 'Time'

The time value specifies how long AFTER the execution of the motor pulse command the controller must wait before executing the next instruction. If the value is zero, then the next command (assuming it is also a motor pulse command) is executed immediately and synchronously with the present one.

T (3 bits): 'Target'

These three bits select the stepper motor drive stage affected by the command. The order runs from Target 1 ('000'), the output stage driving connectors K11 and K12, through to Target 5 ('100'), the output stage driving connectors K19 and K20.

Target addresses 6, 7 and 8 ('101', '110' and '111') are reserved.

D (1 bit): 'Direction'

This bit specifies the direction in which the motors are to turn. D= 1 specifies clockwise rotation, D= 0 counterclockwise.

Table 2: Drilling command

MSB	Byte1	Byte2	LSB
0	T T T	1 1 B B B B B D	D D D D

Addressing: bit 7, byte 1 = '0'; bits 2 and 3, byte 1 = '11'.

T (3 bits): 'Target'

These three bits select the output stage for the drilling command. The machine can be fitted with up to four arms, and each of these arms has its own drill and a dedicated driver stage on the controller board. The targets are numbered from left to right, starting with Target 1 ('000') at connector K10, where the solenoid for the first arm is connected. The corresponding drill motor is connected to K8. Target 4 ('011') is driven from connectors K5 (solenoid) and K3 (drill motor). Targets 5 to 8 ('100' to '111') are reserved.

D (5 bits): 'Drilling force'

The force applied to the drill by the solenoid can be controlled by this five-bit data value. The value sets the amount of energy delivered to the solenoid, and can be set to any number in the range 0 to 31.

B (5 bits): 'Braking control'

When the drilling head is withdrawn the solenoid cannot simply be switched off, since this would cause long-term wear on the mechanism. To prevent undue stress on the motor guide spring, the magnetic field in the solenoid is gradually reduced, so that the drilling head is withdrawn gently.

In principle this value can be set in advance for a particular weight of drilling head. Here, however, we have the possibility of optimising the value. If, for example, you use a more powerful drill motor or a 3-jaw chuck instead of the collet chuck, the weight of the drilling head may be significantly different.

synchronous serial SPI/I²C bus.

The modified Centronics interface uses port D (data signals), RC6 (active-low STROBE), RE1 (SELECT) and RC7 (ERROR). The BUSY signal from the microcontroller is not wired to the standard connection on pin 11, but to the PAPER EMPTY signal (PE) on pin 12. There are several reasons for this: first, it prevents the PCB drilling machine from springing into action when a print job is started under Windows, and second, it allows a printer to operate in parallel with the machine. Further, pulses on the STROBE and BUSY signals appear to affect the internal interrupt processing of Windows: and that is best left well alone.

R44 is the pull-up resistor referred to above that pulls STROBE high in the event of a system crash, thus deactivating the signal.

The microcontroller determines the configuration of the drilling machine via the drilling head limit switches connected to **Bo1-Bo4**. If a drilling arm is fitted and the head is in the 'up' position, the corresponding switch is closed. If an arm is not equipped with a limit switch, the corresponding contacts remain open.

OptSpare is an optical input that, like the switch input Spare, is not used in the present design. **OptBase** is an optical input used to calibrate the reference positions of arm 1, arm 2 and the rotating table. **OptAdd** does the same job for arms 3 and 4. How exactly the calibration is carried out will be described in a later article in this series. The LED can be used to indicate that the machine is operating: it lights during drilling and during the calibration process.

It may seem that 33 I/O-ports are plenty, but unfortunately are not enough for all the signals just described. We therefore use a shift register type 74HC165 (IC14) to combine the bits into a single byte read serially by the microcontroller over a single port bit RA4 (SOUT). A parallel load pulse loads the bits into an intermediate register where they are buffered before being clocked out using the CLK signal (RB2).

The final item of information read by the controller is the 'stop' signal from the solenoid limit switch. These switches are connected to BU1-BU4. Since the current must be switched off quickly, it is not feasible to read these signals out slowly via a shift register. Instead, the stop signal drives the interrupt input RB0 low. Since only one drilling head is actuated at a time, the four limit switches can be connected in parallel.

The rest of the circuit consists of the four identical output drive stages for the solenoids (T1, T2, T7 and T8), the drill motors (T3-T6) and ten identical integrated stepper motor

drivers type PBL3717A from ST Microelectronics. The datasheet for this IC can readily be found on ST's website at www.st.com.

First let us consider the drill motors and solenoids. These are not driven directly, but rather via a programmable logic device type GAL 16V8. You can read the reasons for this in the box *A universal controller board*. Address signals ADR0 (RC3) and ADR1 (RC4) select from the at most four motors and solenoids in the order shown. The level on DRILL (RC5) selects either the motor (when RC5 is low) or the solenoid (when RC5 is high). In this way we make sure that in the case of a failure in the circuit the drill will not turn and the drilling head will be up. When the circuit is switched on, the PIC's ports are in a high impedance state, and resistors R1 and R2 hold the signals at suitable levels to ensure that the motors and solenoids are safely turned off.

The controller activates the selected motor/solenoid pair using port pin RC2. The pin does not just turn on and off, but provide a current control using a PWM signal, different for the solenoid and for the motor.

The solenoids are controlled using a rather complicated current waveform. The timing depends on the instantaneous position of the drilling head (which is determined from the two switches on the head guide), the value specified in the drilling command, and on whether an error situation arises due, for example, to a timeout resulting from a jammed or dirty guide. The important fact is that the PWM control value is calculated at each point in time to actuate the guide solenoid as smoothly as possible.

The solenoid coils are rated for continuous operation at 12 V, and they are driven

(although only momentarily) from the same +30 V supply as the stepper motors. This produces an enormous magnetic field to force the drilling head down. As soon as the appropriate Bo switch opens, the PIC can deduce that the solenoid has moved and it then proceeds to regulate the current to the value specified in the drilling command using the PWM signal. Although the solenoid is initially overdriven, in continuous operation it is only driven at effectively 12 V with an 80% duty cycle. This is within the specified limits. Overall, with typical activity, the duty cycle is only a few percent, and in practice the coils only get warm to the touch.

The drill motors are controlled during switch-on and running using a PWM ramp that reaches 100% duty cycle during drilling.

The stepper motors are activated using the STEP signals. Each PBL3717A integrated driver controls one motor winding, and two drivers (STEPAx and STEPBx) are allocated to each motor. The drivers are responsible for producing the drive pulses and for converting the +5 V logic levels to +30 V, the operating voltage of the stepper motors. The microcontroller activates the driver via the level (respectively from STEPAx and STEPBx) at the PHASE input, and a current then flows from OA to OB. Inputs IN0 and IN1 have a special function: they allow control of the output stage drive current.

This is normally achieved using a

retriggerable timing circuit for each output stage. If no more motion is required of the motors, the pulses stop and the current is reduced in the motors. Here, this is realised via the signal from RB7 (pin 40) connected to the IN1 pins of all the PBL3717s. When motion is required, all the motors are driven with full current (even those which are not to move), by setting IN1=0. Shortly after the motion stops, IN1 is set to 1 again, and the current drive is reduced to 19%.

Why do we increase the current to the motors not involved in a motion? Partly because in this way the power supply is loaded to the same extent independent of the motion, and partly because, as a result of various resonances in the system, those motors driven at 19% of the maximum current can be joggled out of position. Although this is rather unlikely, it would lead to errors almost impossible to track down.

(010024-2)

In the next instalment of this series we get down to the construction of the controller board as well as the nuts and bolts of the project. The various parts of the PCB drilling machine will be brought together, with illustrations — and lots of glue and grease!

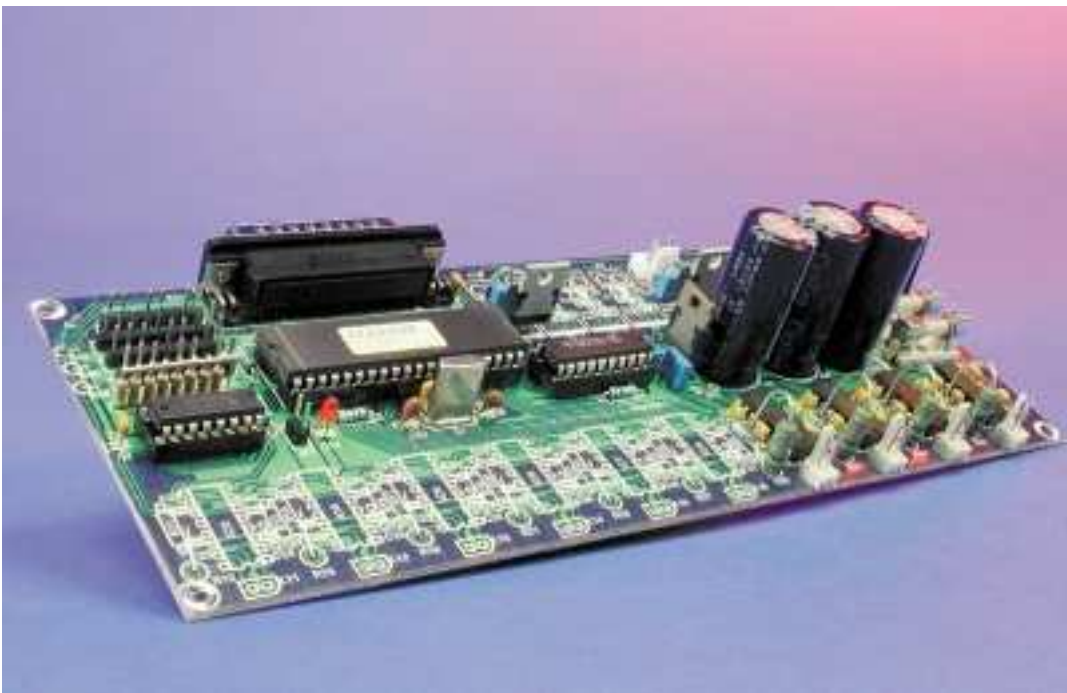
PCB Drilling Machine (3)

Part 3: PCB construction

Design by T. Müller (Radix GmbH)

www.radixgmbh.de

The controller board contains all the electronics for the plate motor and four drill arms. As a matter of course, the board population is in accordance with the actual number of drill arms you have in use.



All the components of the circuit can be built onto a single-sided printed circuit board. The PCB and component layout are shown in **Figure 1**. The power supply (as far as the transformers) is also built onto the circuit board: it provides a +30 V supply for the solenoids and stepper motor drivers, +15 V for the drill motors, and +5 V for the logic, regulated by IC1. The layout is very straightforward: along one long edge of the board we have the stepper motor drivers, each with a pair of terminal posts (K11-K20) for connection to the stepper motor windings. Since comparatively

high currents flow through the ICs, they should be soldered directly to the board without sockets.

Along the other long edge we find the terminals for the jumpers and switches, the 25-way sub-D connector, the output stages for the solenoid drivers and drill motors (connected via the small PCB connectors K3-K10), and finally the power connectors K1 and K2. The microcontroller, the GAL and the shift register are in the middle of

the board, and naturally good quality IC sockets should be used for them. One toroidal mains transformer with 2·12 V/ 80 VA secondary windings is used for each pair of drilling arms. If four arms are fitted, then the windings of the two transformers should be wired in parallel to connectors K1 and K2.

Fitting the components should present no difficulties. One point to note is that although all the ground connections are linked, the two marked heavy gauge wire links on the circuit board next to the fuses must be soldered in. This is to avoid long current paths over the circuit board and the

interference problems this causes. The ground plane is split between the sub-D connector and the nearest corner of the circuit board: with the type of drilling motor used this gives the best EMC performance. With other types of motor it may be necessary to bridge the gap.

The reset input JP1 can be brought out and used as a kind of emergency switch. Using this switch will certainly confuse the system, however, since the controlling PC

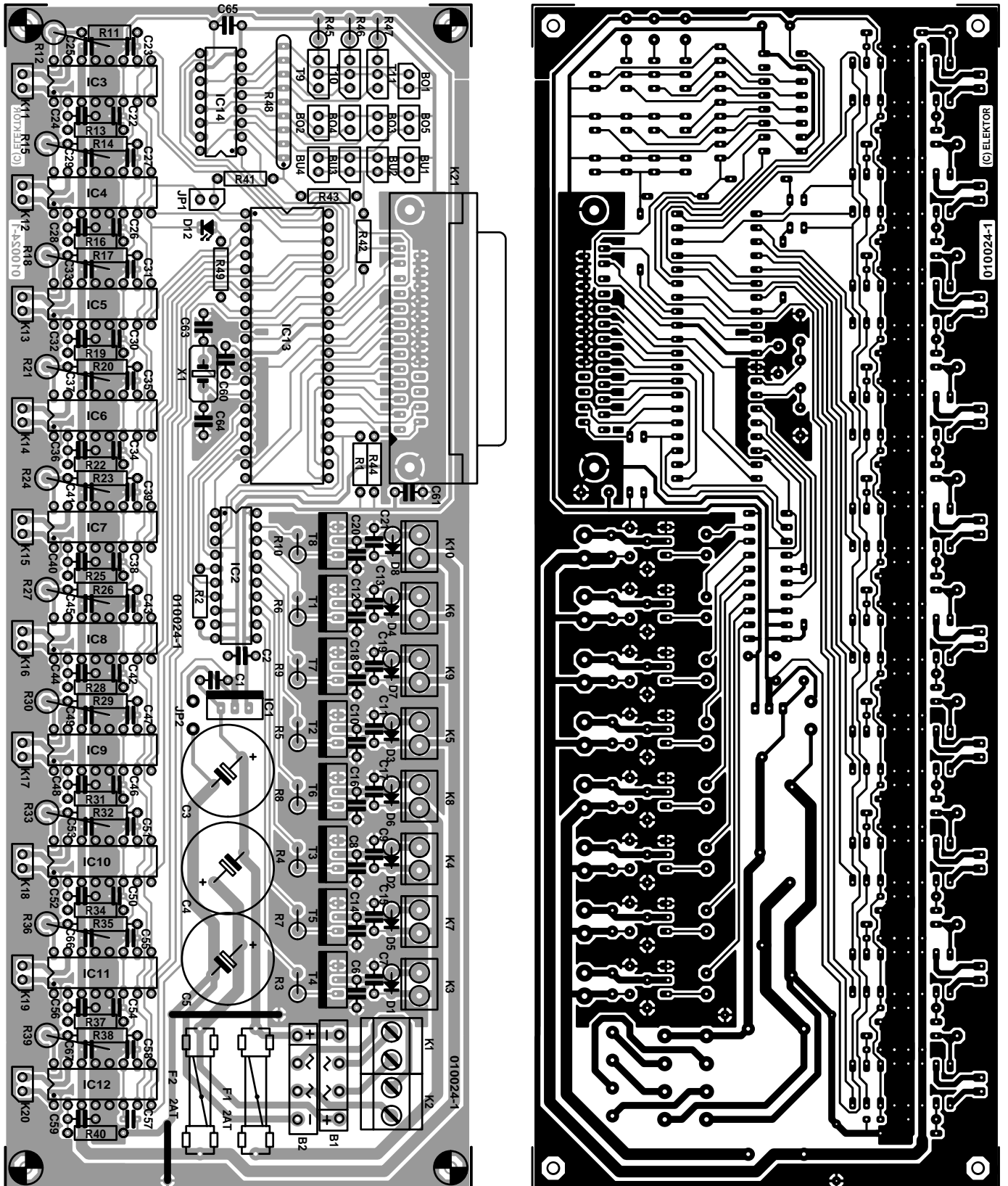


Figure 1. Track layout and component mounting plan of the controller board.

will not know when it has been operated; but in any case an emergency switch is not a bad idea.

(010024-3)

In the next instalment of this series we get down to the nuts and bolts of the project. The various parts of the PCB drilling machine will be brought

together, with illustrations — and lots of glue and grease!

Compact PCB drilling machine

After discussions with several readers during the HobbyTronic Fair in Dortmund, Germany, and several e-mail enquiries and telephone calls, we have decided to describe briefly once more the operation of the PCB drilling machine, along with its advantages and capabilities.

Basic features of the PCB drilling machine

Our project is a CNC machine which carries out operating commands calculated with the help of a computer. Conventional CNC machines are constructed using linear components, such as worm drives, ball races, linear guides and many other — not inexpensive — specialist components. The construction of a linear machine appears very straightforward: simply select the required components in the appropriate sizes, fit them together, and set the computer going. But the devil is of course in the detail: all the parts must be precisely mounted, parallel, true, and free of play, or else things will grind and jam.

Our PCB drilling machine is, as far as function and aims are concerned, exactly like an X/Y/Z-machine. It can handle circuit boards up to 200 mm by 300 mm with an accuracy of 0.03 mm, which is entirely adequate for our application. Because of its lightweight construction it can move extremely quickly, at almost 80 mm/s. In construction it is fundamentally different from linear machines. Everything is reduced down to two rotating, vertical axes. The only specialist components required are good-quality ball bearings

to support these axes without play. In appearance, the machine resembles a record player: in the middle is a rotating table, on which the workpiece — the circuit board — is fixed. Instead of a cartridge we have a drill, which can be moved up and down. And in exactly the same way as the tone arm can reach any point on a record, the drill can get to any point where drilling is required on the circuit board.

As well as the rotating table, there is space for more tool arms, which can all operate practically simultaneously on the same circuit board. The advantages are that the job gets done quicker, and no tool changes are required.

The tool can be a drill, which can be used for drilling holes for the leads of components, or a milling tool, with which the conductors can be milled away. Drilling data is taken from one's preferred circuit board design program in Excellon format: all programs are capable of outputting this format. Milling data is sent to the machine in the HPGL plotter control language, allowing any desired cutting path to be followed. The machine can be connected directly to the printer port of an ordinary PC without extra hardware.

The machine is not designed for heavy-duty milling operations. If you want to produce milled aluminium front panels, you will have to look elsewhere. The key feature of the machine is its high speed, which can only be obtained with a lightweight construction. It is sturdy enough for working on printed circuit boards, but not sturdy enough to withstand the high sideways forces that arise when milling metals. The machine has to be small and light enough so that it finds a place not just in the workshop, but actually on the experimenter's bench

COMPONENTS LIST

Basic version, w/o tool arm

Resistors:

R1,R2,R41,R43,R44 = 10k Ω
 R11,R14 = 1k Ω
 R12,R15 = 1 Ω , 1W
 R13,R16 = 56k Ω
 R42 = 220 Ω
 R45,R46,R47 = 560 Ω
 R48 = 8-way resistor array 47k Ω
 R49 = 1k Ω 5

Capacitors:

C1,C2,C60,C61,C65 = 100nF 5mm lead pitch
 C3,C4,C5 = 3300 μ F 50V radial
 C22,C25,C26,C29 = 100nF 2.5mm lead pitch
 C23,C27 = 820pF 2.5mm lead pitch
 C24,C28 = 820pF 5mm lead pitch
 C63,C64 = 15pF

Semiconductors:

B1,B2 = GBU6B
 D12 = LED, red, 3 mm dia.
 IC1 = 7805
 IC2 = GAL 16R8 (programmed, order code **010024-31**)
 IC3,IC4 = PBL3717A

IC13 = PIC16C64-20P (programmed, order code **010024-41**)
 IC14 = 74HC165

Miscellaneous:

K1,K2 = 2-way PCB terminal block, lead pitch 5mm
 K11,K12 = 2-way pinheader, 0.1in. lead pitch
 K21 = 25-way sub-D plug, angled pins, PCB mount
 F1,F2 = fuse, 2A (T) time lag with PCB mount holder and cap
 BU1-BU4,BO1-BO4,JP1 = 2-way SIL pinheader
 JP2 = jumper
 X1 = 20MHz quartz crystal
 PCB, order code **010024-1**, see Readers Services or www.elektor-electronics.co.uk

COMPONENT LIST

(external components)

1 mains transformer for 2 drill arms: toroidal 2 x 12V 80VA
 T9,T10,T11 = SFH309-F4 or LTR4206E
 IR-LED: TSIP4400
 Stepper motors type KH2JM2-851

COMPONENTS LIST

(for Tool Arm # 1, replicate for other arms)

Resistors:

R8,R10 = 100 Ω
 R17,R20 = 1k Ω
 R18,R21 = 1 Ω , 1W
 R19,R22 = 56k Ω

Capacitors:

C16,C20 = 100nF 5mm lead pitch
 C30,C33,C34,C37 = 100n, 2.5mm lead pitch
 C17,C21 = 220nF 5mm lead pitch
 C31,C35 = 820pF 2.5mm lead pitch
 C32,C36 = 820pF 5mm lead pitch

Semiconductors:

D6,D8 = BUV27-200
 T6,T8 = BUZ11
 IC5,IC6 = PBL3717A

Miscellaneous:

K8,K10 = 2-way pinheader, lead pitch 3.96mm
 K13,K14 = 2-way pinheader, lead pitch 2.54mm

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-

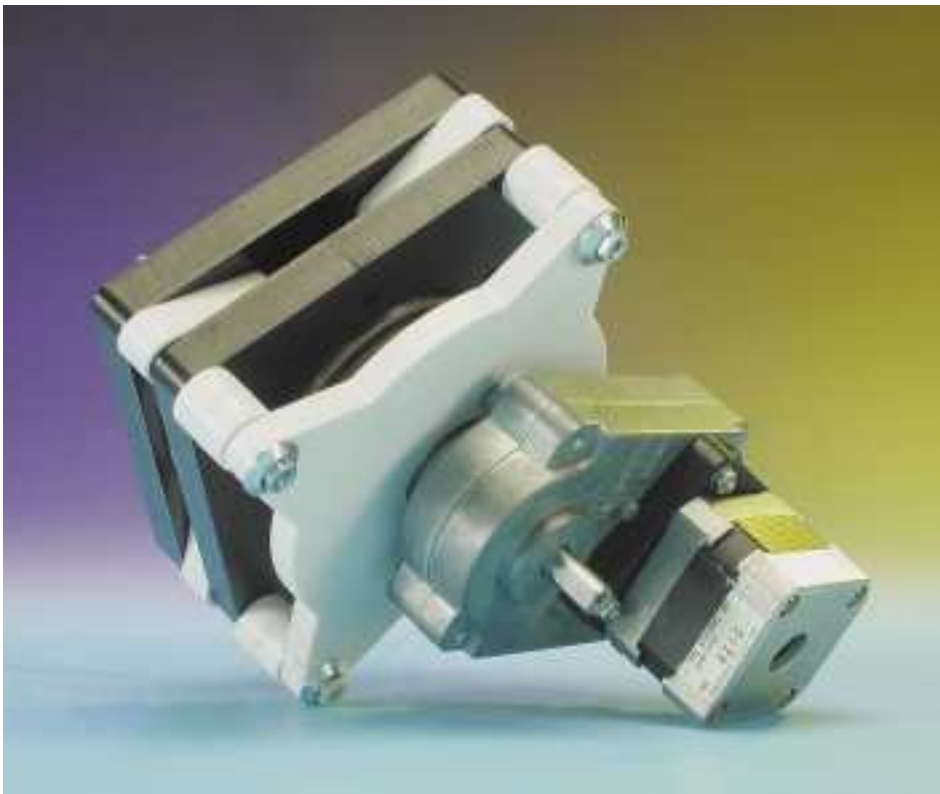


Figure 6. The motor is connected to the gearbox via a flange screwed to each, and the star-shaped drive fixing is screwed to the output of the gearbox.

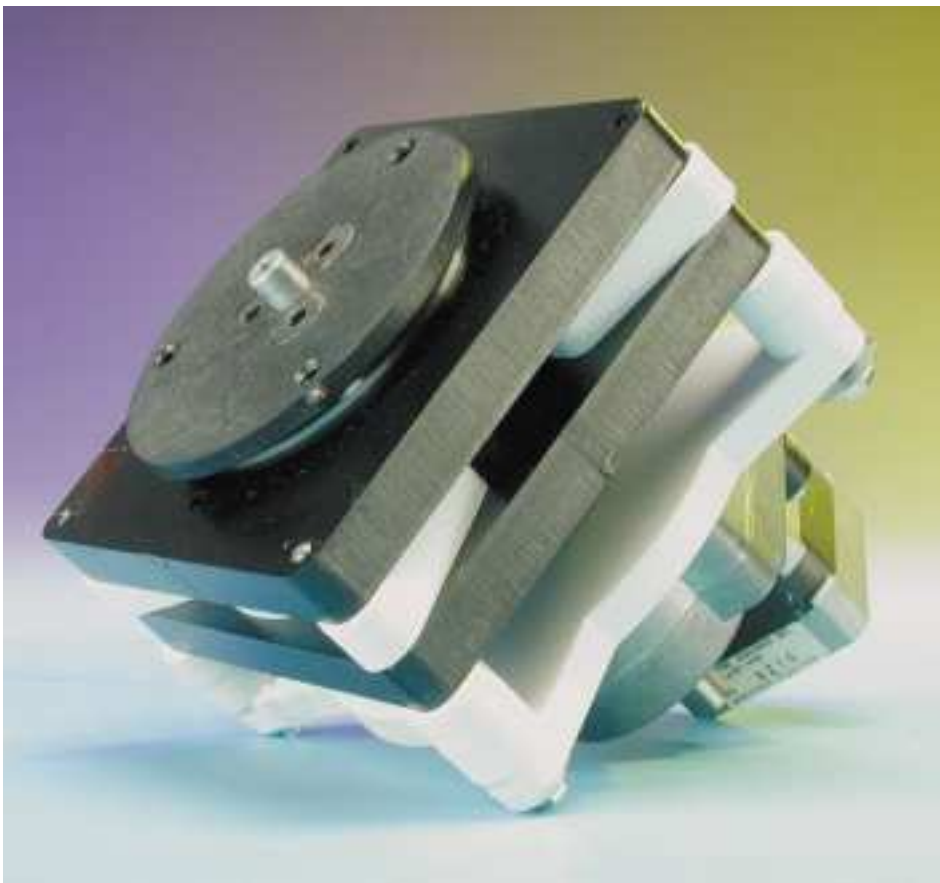


Figure 7. Completed drive unit viewed from the opposite side.

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.

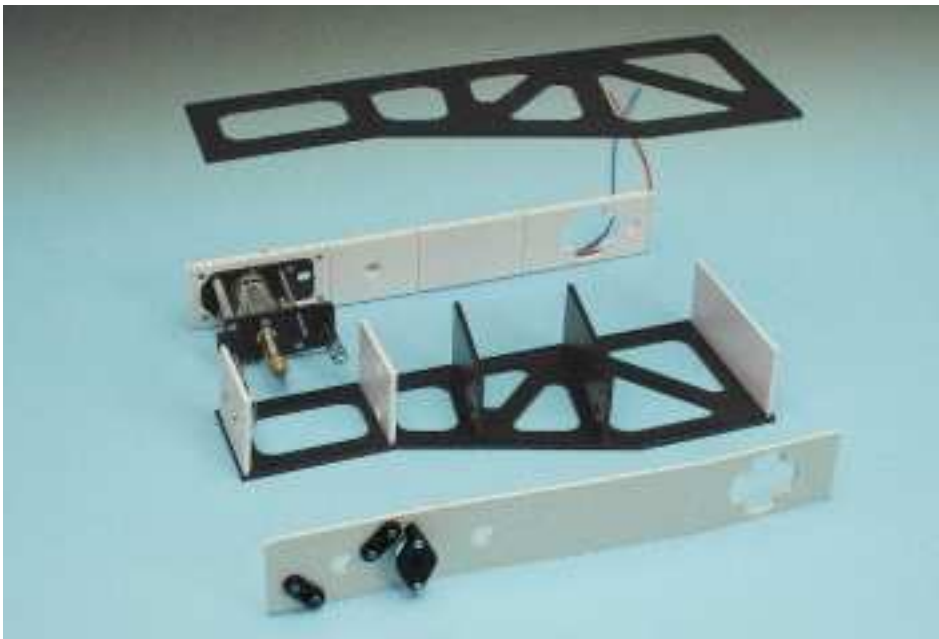


Figure 10. The parts of the arm laid out as they are to be glued. The lifting solenoid is not shown.

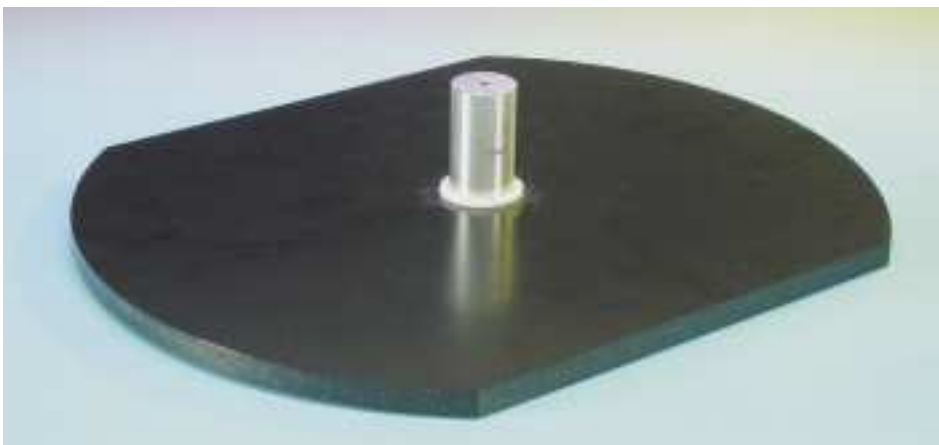


Figure 11. Turntable and axle.



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.

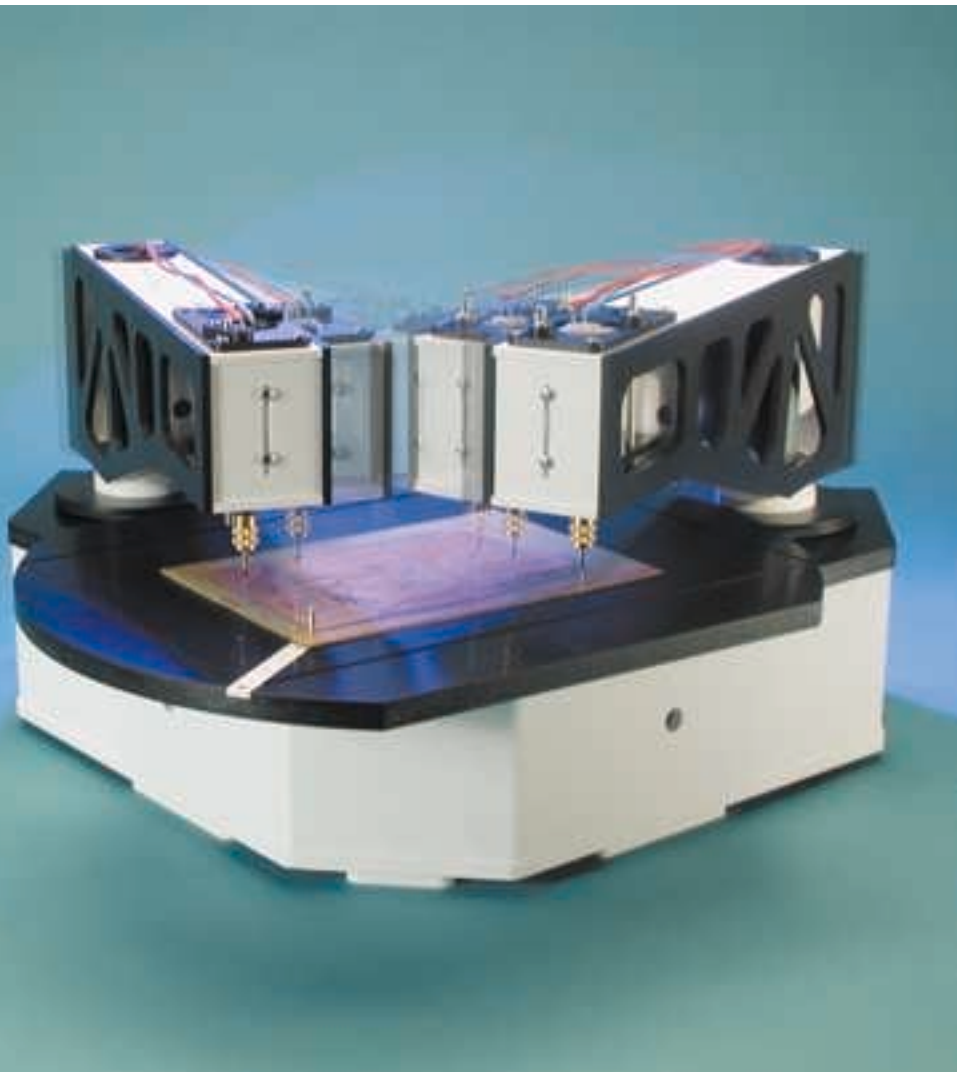
PCB Drilling Machine (5)

part 5 (final): ready to go!

Design by T. Müller (Radix GmbH)

www.radixgmbh.de

In this final part on the construction of the PCB drilling machine, all its functions are tested and the machine is calibrated. The machine is then ready for its first job.



By the time you read this, the first PCB drilling machines have already been delivered — and perhaps also assembled. It is therefore high time that our description of the machine be completed. All that remains is the calibration of the machine. The machine must be calibrated before the driver software is able to determine the reference point, calculate drilling coordinates correctly, position the arm(s) and turntable and raise and lower the drilling head. A number of position sensing switches play a key role in initialising the machine. In this final part we describe a complete calibration procedure for the arms and turntable with the aid of the TanBoTest software, and the use of the TanBoDrive driver software for drilling PCBs. Please note that the German working title of the project, 'Tanbo' (for 'Tangential Bohrer') is retained in the names of the files and programs developed for the project.

Position sensing switches

In the machine enclosure there is an optical switch situated just between the two drive shafts for the tool arms, which operates on the reflective principle. **The position of the turntable** and of tool arms 1

and 2 can be determined using this switch. The sensor consists of a light-emitting diode (LED) which emits infrared light vertically upwards, and a phototransistor, also pointing upwards, which detects when a reflective object passes in front of the LED. For this reason there is a reflector made of metallised foil on the pointer attached to the tool arm drive shaft, as well as on the underside of the turntable.

This reflective light switch operates using unmodulated infrared light and is therefore highly prone to interference from other IR sources such as daylight, artificial light, and even the light from cigarette lighters! While under the circular part of the turntable the switch is completely covered, but the turntable's two flattened sides, which are opposite one another, can let stray light in. And why is the turntable not a perfect circle? It saves a little material, but more importantly this is the only way that access can be provided to the sensor to allow for inspection and cleaning.

If stray light should interfere with the sensor, the turntable need only turn until the sensor is again in the dark, and then continue to turn until the sensor sees light again **but only over a narrow angle**. In this way, the position of the reflective foil can be found. Upon subsequent rotation in the opposite direction the transition from light to dark must be found at the same point. Once the system has determined the position of the turntable, it then immediately knows where the flattened sides are that give rise to the risk of light interference: the software takes these into account.

The process for **initialising the tool arms** is simpler, since in each case the sensor is covered by the turntable and is therefore protected from stray light. The system looks for (and finds!) the reflecting position of the arm.

If two arms are in use simultaneously, there is a problem that must not be overlooked. If both arms attempt to travel to their end positions at the same time, there is a risk of collision. This difficulty is overcome elegantly as follows: the Windows software notes the position of



Figure 1. The TanBoTest utility tests the various microswitches and determines the operating parameters to be stored in the configuration file.

the two arms at the end of a run and stores the relevant information in a temporary file. When the program is next run, the file is loaded, and so a collision can be avoided. If the program terminates abnormally (which can be the rule rather than the exception under Windows!), this file will be found to be missing the next time the program is run. In this case the program enters manual control mode. A graphical representation of the drilling machine appears on the monitor: the tool arms can be dragged into position on the screen using the mouse so as to represent the actual state of the machine. This need not be absolutely accurate, although the machine will rely on your input.

Two microswitches are provided in the tool arms to determine which of three possible states the drill head lift mechanism is in.

State 1: Drill head lift mechanism at the top of its travel, in rest position. If no current flows in the solenoid, the force of the spring built into the lower part of the arm must push the drill head to the top of its travel and thus open switch BO (BO is a nor-

mally-closed contact). If the arm is not fitted, there is no BO switch, and the corresponding switch inputs (BO1-BO4) on the circuit board are open. The software can check each arm individually to determine whether the corresponding drill lift mechanism is in the up position.

State 2: Drill head lift mechanism is lower position, determined via BU. All BU contacts of all arms are normally open and are connected in parallel. This can work because only one head is allowed to be in the down position at a time. Safety logic in the GAL prevents multiple drill head lift mechanisms from being actuated at once. And if (because of a mechanical failure) more than one drill head should be down at the same time, this can be detected via the BO contacts.

State 3: This state is between the other two: neither up nor down. Assuming nothing has gone wrong, this can only happen during motion of the drill head. If this state occurs while the head is not in motion, the drill lift guide is probably jammed.

During drilling the Windows software can measure the time between the closing of the upper switch and the closing of the lower switch, which gives the time taken to drill the hole. A broken drill can be detected by this

time being too short, and an overly worn drill can be detected by this time being too long. If it takes longer than five seconds to drill a hole, the controller automatically switches off the drill lift mechanism and the drill motor.

The first test

The *TanBoDrive* driver software requires that the drilling machine be fully functional, both electrically and mechanically. The input data — in this case drilling data in Excellon format — are specially processed and transformed into the required commands, in the form of motion commands and drilling commands, to be sent to the microcontroller. More than this *TanBoDrive* cannot do. Linear interpolation, as required for interpreting HPGL-format files, is included (otherwise the machine could not even move from hole to hole), but HPGL import is disabled and will only be enabled when the previously-mentioned milling arm has been thoroughly tested. *TanBoDrive* should be compared to a printer driver: if the controller board or the mechanics are not in order, the program simply gives up.

First, then, servicing and diagnostic software is required to test all the functions of the controller board and the connected units. The various switches are particularly important, since they must be correctly wired as well as correctly fixed. All components must first be made to work as intended and all tests must be passed before we can move on to drilling a circuit board.

This requirement is met by the *TanBoTest* test software (**Figure 1**). When the software is started a dialogue box appears showing the states of all the switches, updated every 300 ms. The optical switches can be tested with a finger or any reflective object. The drill motors can also be turned on and off, and the lift solenoids can be supplied with a selectable current, in order to help check (and, if necessary, improve) the smooth running of the guides. The program, freely available from this url

www.radixgmbh.de/deutsch/menu_bestellung.html

is self-explanatory and shows the states of all the switches. It is essential to check the messages in the dialogue box and to test the three positions of the drill head lift mechanism. Test the optical switches with a reflective surface and with stray light. Set *Target* to 1 and push the drill head down: you should see activity on BO1 and BU (but not BO2); likewise for the other target. Using the box marked *Coil*, adjust the solenoid current to find the value required to drive the drill head down. The numbering of the targets is clearly defined: looking at the machine from the lon-

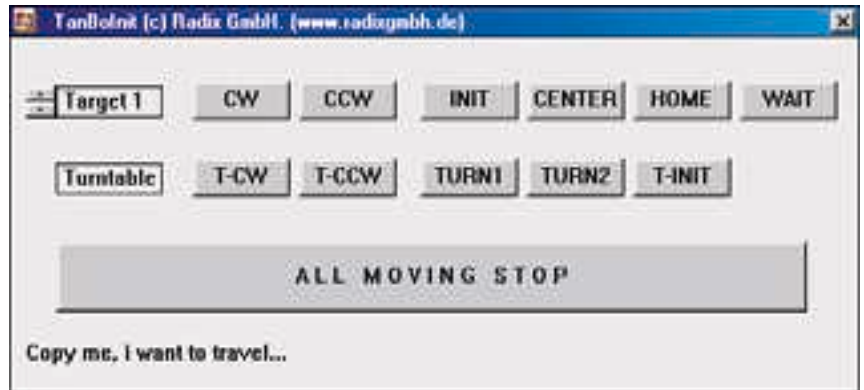


Figure 2. Initialisation program for calibrating the arms and turntable.

ger side and above, the left-hand arm is tool arm 1 and the right-hand arm is tool arm 2.

The switches are so vital to the operation of the machine that the switch inputs to the controller board are not fitted with a socket, but rather soldered directly. Although a little more tedious to assemble, the connection is more secure. A length of black PVC insulating tape should be used as a strain relief on the cable which comes from the drive shafts, so that the connecting pins are not under mechanical stress. Ensure also that the cable is free to move.

Here is a summary of all the functions of the test software:

The *Target* box selects between output drive stages 0-3 for drill and coil.

The *Coil* box can be set from 0-65 to control the solenoid current from 0 (off) to 65 (100 %) for the output stage selected under *Target*. Beware that the original solenoids are overloaded at the 100 % setting and cannot run continuously at this current. A duty cycle of about 30 % is possible. Continuous operation is possible at a setting of 26. The software switches the coil off if the setting is left unchanged for 5 s.

The *Drill On* button switches the output drive for the drill motor of the selected target on and off.

The next four boxes are to do with the drill cycle. Each target has its own press and brake values, while the shake and hold values apply to all four targets. Each click on the

Cycle button initiates a drill cycle with the selected parameters.

The *Press* box sets the force with which the drill is pushed down on the circuit board. This value — to a certain extent — affects the speed of drilling. If the value is too small, the solenoid will not exert enough force to drive the drill down. If it is too big, the drill can in extreme circumstances be damaged when it hits the circuit board. The value should be set to suit the drill: a value between 24 and 40 is normal. If satisfactory operation can be obtained with a value below 25, it is a credit to the conscientiousness and care with which the assembly has been built: congratulations!

The *Brake* box sets the braking value needed to cause the drilling head to come to a gentle stop at the end of its upwards travel. If the value is too high, the head guide will rebound down before finally coming to a halt; if on the other hand the value is too low, the upper switch will be (ab)used as a mechanical endstop, which is definitely something to be avoided. Find the setting where the head does not rebound and then add one or two to the value.

The *Hold* box sets the time (in units of 52 ms) for which the drill remains down after the circuit board is drilled through. Too short a time leads to swarf not being fully cleared from around the hole on the underside of the board, while if the time is set too long, it is simply time wasted. The default value of 10 (i.e. 0.52 s) is a

good typical value for clean holes of all sizes.

The *Shake* box is aimed at the automatic clearing of faults. If for example the guides are very dirty and not running smoothly, the force of the spring may not be enough to lift the head fully up and operate the upper switch. This fault can be simulated by pushing the guide down a couple of millimetres so that the switch BO indicates 'Head not up'. If the head stays in this position when it is carefully released, then the fault condition has been replicated. The shake value controls how the head is driven down a small amount and then released, without braking, so that it springs back. The force with which this is done must of course not be so great that the drill hits the circuit board when it is not running.

The *Cycle* button initiates a complete drilling cycle on the selected target. The feed time is measured and displayed (minus the Hold value). If the cycle completes without error, *OK* is displayed; otherwise, *Error* is displayed. If there is an error with a feed time of around 4.5 to 5 s, this indicates that the head never reached the lower position: this condition can be simulated by setting *Press*=0. With a feed time of around 0.5 to 1 s, the fault is that the upper end position was not reached: this condition can be simulated as described above with *Shake*=0.

The values established for the drill cycle using the *TanBoTest* program are recorded in the configuration file.

The *Speed* button sends data at a steadily increasing rate to the controller until the PC's processor can no longer keep up with the data stream and the controller's FIFO empties.

The *Speed* indication in mm/s refers to the speed of interpolation on two axes, for example to simultaneous motion of the turntable and the tool arm at the given speed. The mechanical maximum speed limit is around 80 mm/s. If the speed test gives better results then there is a surplus of processing power, and further axes

can be made to move simultaneously. When trying the speed test, the stepper motors should not be connected!

The *EventsOn* check box, when checked, gives up some processor time during the speed test to the operating system, and hence to other processes. This is done to allow other tasks to make progress. The *TanBoDrive* control program uses a dynamic algorithm to determine how much processor time to allow the operating system, ensuring as necessary that the system does not come to a complete halt.

At start-up, the software must be informed of the address of the LPT port to which the controller is attached. This setting can be obtained from the Windows system information, and is generally 0378-037F_{HEX}. Open an ASCII text editor with an empty file and enter on the first line the first of these numbers (here 0378). Now press Return twice and save the file as *lptaddr.txt* in the same directory as *TanBoTest*. This file is processed by all the drilling machine programs and must be placed in the same directory as all those programs.

An important part of the test program is the measurement of the transfer rate for data streams over the Centronics interface. See the text box for further details.

Adjusting the arms

After you have checked the operation of the hardware — and, if necessary, corrected it — using the *TanBoTest* program, the machine must be adjusted and calibrated. Calibration — also known as setting the zero point — presents no difficulty with a linear machine, but how can we adjust something that is circular and does not have a start or an end? Further, several of the components of the drilling machine can be assembled in different ways: for example, the drilling axis can be moved to and fro by several millimetres before being firmly screwed down. In order to calculate the X/Y coordinates exactly, the length of the arm must be known absolutely precisely. It is all done, as we said in the first

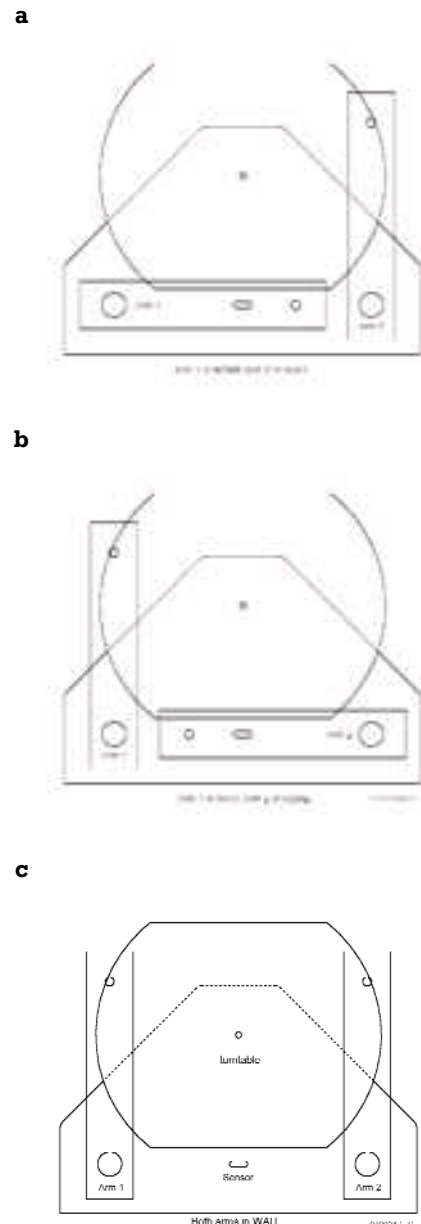


Figure 3. The arm positions for calibrating the turntable: home/wait (a), wait/home(b) and wait/wait (c).

article in this series, without precision measuring instruments of any kind.

All that is required for exact adjustment is the *TanBoInit* software (**Figure 2**), which also uses the *lptaddr.txt* file. Note: if at any time during the adjustment procedure anything untoward should happen, or if you think from the way the arms are moving that a collision is imminent, click on the 'emergency brake' ALL MOVING STOP: the machine will come to an immediate halt.

First we calibrate the arms, and then the turntable. The other way round is not possible, because the turntable can only be set up

with properly calibrated arms. Each arm can be moved at will using the buttons CW (clockwise) and CCW (counterclockwise). In *TanBoInit*, as before, which of the arms 1-4 is moved can be set via the *Target box*.

First check that all the tool arms turn appropriately in response to the CW and CCW buttons. Select also the targets for which an arm is not fitted, and press CW and CCW. In these cases nothing should move.

The ordering of the target numbers has already been checked in the test program. The four BO switches are shown in a column

above one another. If you push down the head on arm 1, you should see activity on BO1 and not on BO2. Check this very carefully, since otherwise the collision detection will not work. The drives are extremely powerful and are easily capable of destroying one another!

The turntable has its own CW and CCW buttons, while the *Target box* remains always assigned to a tool arm. Before an arm can be calibrated, the turntable must be rota-

ted so that the reflective light switch is off.

Begin with the calibration of arm 1 (target 1). Move arm 2 (if fitted) to the WAIT position, as shown in **Figure 3a**. In the kit of parts which contained the brass chuck, you will find a short polished metal pin, 20 mm long, left over. This is the only calibration tool required. Push this pin as far as possible, without using excessive force, into the hole in the aluminium shaft at the centre of the

Buffer Underrun?

The drilling machine controller software forces Windows to operate in real time. Although Windows can output data very quickly, it cannot coordinate the timing of the output exactly, and for this reason the controller includes a FIFO (first in, first out) memory which allows the transmitted data items to be brought back into step with one another.

This works perfectly as long as we can ensure that the data stream from Windows never stalls for so long that the FIFO becomes completely empty. The FIFO has 35 slots, from which we can make the following example calculation.

Let us suppose that a stepper motor is to be turned at the rate of 500 steps per second. The motors used in the drilling machine have an angular resolution of 1.8°, and so this corresponds to a rotation rate of 2.5 rotations per second. Since the following gearbox has a ratio of 200:1, the output shaft turns through an angle of 4.5° per second. The circumference of the circle swept out by the machine is 1510.6 mm, and so the head moves at a speed of 18.88 mm/s.

What data rate is required for this? The specification of the Centronics interface allows for at least 20000 bytes/s. On an older PC (150 MHz Pentium) the bandwidth was measured at around 46000 bytes/s.

Each motor step requires exactly one transfer, and so the required 500 transfers per second occupy a mere 2.5 % of the available bandwidth. Bandwidth is therefore not a problem.

At 500 steps per seconds each step lasts 2 ms. The FIFO, with its 35 slots, can hold 70 ms of data. In these 70 ms the processor we used runs for 10.5 million cycles. If these cycles are consumed by Windows and the FIFO cannot be refilled with data in time, then there will be a break in the data stream.

For this reason it is not possible to run programs in parallel with the drilling machine that load the system heavily or which consume practically all its processing power.

How can we check that the data stream is in fact continuous?

A control signal on the controller board indicates when the FIFO is empty.

This signal is taken to pin 12 of the Centronics interface (paper empty) and is asserted when all the bytes in the FIFO have been processed.

So why is it so important to have a continuous data stream? Stepper motors that are to be used at a high stepping rate must be

brought gradually up to the desired target speed using a specially-designed ramp function. The moment of inertia of the rotor is so great that it cannot follow a rapid speed change and drops steps. This effect is much more noticeable when starting than when stopping, and depends on the individual motor and its driver stage. So, if the pulses are suddenly stopped while the motor is still turning, the rotor will not come to an immediate halt, but rather jump on one or two positions because of its moment of inertia. That does not matter, or course, if the system has feedback, for example in the form of an encoder on the drive shaft.

And the solution? The best answer is not to allow interruptions in the data stream. Before issuing each motor pulse command, we can check whether the controller FIFO signal is asserted, indicating that the FIFO has emptied. Then — if we know the characteristics of the motor — we can decide on the basis of the previously-set command rate whether the motor might have got out of step (this can only happen above a certain, relatively high, rotation speed), and whether the drive in question should return to the limit switch position to recalibrate itself. If this behaviour recurs—that is, if you notice that the machine is returning to its reference points too frequently—then either you will need to use a faster computer, reduce the load on your system, or reduce the pulse rate for the motors.

None of this is really a problem, but the unnecessary recalibration movements do affect the average speed of the machine and hamper the system somewhat. A reduced motor step rate can increase the overall speed of the system, if it reduces the number of recalibration movements.

And there is an even better solution: the FIFO memory can be expanded, as used to be done with print spoolers. Exactly the same principle can be applied to the drilling machine. With a small static RAM of for example 8 kbyte the capacity of the FIFO can be increased to 8.25 seconds of data at the frequency of 500/s mentioned above.

This extra FIFO would take the form of an additional unit; it is not possible simply to change the microcontroller. Microcontrollers with a large amount of on-chip RAM are significantly more expensive. The additional module could be supplied with power via the OptSpare pin (5 V DC). Another possibility would be a small module on a printed circuit board to fit in the microcontroller socket which includes the extra buffering.

If such an idea is implemented, you will be able to read about it either in Elektor Electronics or on the drilling machine website.

turntable. This hole is machined extremely accurately: on pulling out the pin you should hear a satisfying 'plop'.

Fit the largest insert into the chuck on arm 1 and tighten the nut loosely. Loosen the eight screws in the head guide bearing block. There are eight oval places where the 4 mm shafts for the head guide assembly are fitted. Do not remove the screws completely, but only loosen them so that the head guide assembly can be moved laterally to and fro.

Using the CW and CCW buttons, bring arm 1 to the centre of the turntable until the chuck can be pushed down exactly over the pin. The head guide assembly must have enough play that no force is required to push the arm into the correct position. The precision obtained by this procedure is improved if the last command given is always CCW.

Now press the head assembly fully down and tighten the nut fully. The entire assembly is held vertically in the centre of the table by the steel bolt and so the bearings are forced into their position. Now, with particular care, tighten up first the lower, and then the upper bearing screws, initially gently, and then tightly. The nut can now be removed. The head guide now glides (we hope) smoothly up. The smoother this motion, the faster the machine will be able to drill.

Two important things have been done here: the head guides are now perfectly vertical and the arm length is set to exactly the desired value. One of the polar coordinates — the length — is now done. The coordinate that remains, the angle, is handled by the software. Press the *INIT* button for the arm that has just been calibrated. The arm automatically moves out until its pointer is located over the optical sensor. The software now knows exactly the number of angular steps between the end point and the mid-point. The mid-point corresponds to an angle of 45°.

Now, to test the middle position, remove the bolt from the central hole and fit it directly in the chuck. In the software, press the *CENTER* button, and the arm should move to the

middle position. When the axle is pushed down, the pin should sit perfectly in the hole. Release the chuck, allow the head to slide up, move the arm away and then remove the pin from the hole.

If the positioning is not perfect in this experiment, the arm angle can be adjusted using the CW and CCW buttons. The last command should, as before, always be CCW. After each adjustment the *INIT* button must be pressed again to store the new reference values. After the button is released, the arm travels immediately back to the sensor position. Once the mid-point positioning is to your satisfaction, press *HOME*. Target 1 now travels to its home position. The procedure with the second arm is the same: first, as shown in **Figure 3b**, bring arm 1 out of the danger area and click on *WAIT*. When calibration is finished, be sure to keep the bolt in case you wish to recalibrate the machine after modifying or expanding it.

Turntable calibration

If you have calibrated two arms, then arm 2 will be over the sensor and arm 1 will be in the wait position. Press *WAIT* also for arm 2 so that both arms are vertical and parallel to one another. If you have only calibrated one arm and this is in the home position, then you should also press *WAIT*. To calibrate the turntable put the approximately 8 mm long registration pin from the bag which contained the chuck into position 1 of the registration guide: this is the outermost hole on the rim of the turntable. Set target to 1 and press *TURN1*. Arm 1 makes a small movement from its waiting position towards the turntable.

With the *T-CCW* button (NOT *T-CW*!) turn the turntable so that the bolt in the registration guide lies under the chuck. Now with the *T-CCW* and *T-CW* buttons you can carry out a fine adjustment so that the chuck slides over the bolt when the head guide assembly is pushed down. The last command should again be a *T-CCW*. When you are happy with the positioning, press *T-INIT* and then the button *TURN2*. The turntable turns about 140° coun-

terclockwise. At the same time the tool arm also moves and the two come to the second possible intersection point. The mathematical background to the positioning scheme for the drilling machine is shown in a Flash animation at this url

www.radixgmbh.de/deutsch/menu_link.html

When the turntable and the tool arm come to a halt, the test pin should once again slide into the chuck smoothly and without undue force on either part when the head guide assembly is pushed down. Once the calibration of the two turntable points is complete, press *T-INIT* again: the turntable will turn to the point where the optical switch is activated, the *HOME* position, and then the tool arm moves to the *WAIT* position.

If a second arm is fitted, the calibration procedure should be repeated. Although this is unnecessary from a mathematical point of view, it helps to compensate for tolerances in the system.

When the program is exited by clicking on the cross at the right-hand end of the title bar, the file *TanBo.def* is written to the hard disk. This file contains all the measured parameters which are used by the control programs. Without *TanBo.def* the *TanBoDrive* program cannot be started.

Setting up the machine also requires checking the planarity of the turntable. The more level the turntable, the less packing material is required between circuit board and turntable. The turntable is fixed to the drive axle via a generously-dimensioned fixing flange in the form of a circular steel disc by three M6 screws, easily accessible from above. By turning the turntable you can test to see if the surface moves up and down. Mark the 'deepest' point and dismantle the turntable. On the marked point, on the flange, lay a scrap of paper wetted with salad oil. The reason for the oil is that it makes the paper more dimensionally stable so that the thickness of the paper is not affected by humidity. Screw the turntable down onto the flange, tightening the screws fully. After at most two or three trials you should be able to turn the turntable without measurable vertical wobble.

The packing material — a thin piece of card or a couple of sheets of paper — is needed because drill points are in general not cylindrical but rather conical in shape. The card must therefore be thick enough for the point of the drill. The maximum feed distance, and so the lowest position of the drill, can be set by an adjustment screw at the front of the tool arm.

Drill!

The machine is now calibrated and ready to drill your first printed circuit board. The basic requirement (apart, of course, from an etched but as yet undrilled circuit board) is a drilling file in Excellon format. Practically every printed circuit board layout program is capable of exporting a drilling file in this format. Among other things, the Excellon file contains the Cartesian (or X/Y) coordinates of each hole specified in the layout. The file also contains control information and (possibly in a second file) information specifying the diameter of the hole to be drilled at each given coordinate pair. This file can be opened, examined and even modified using a normal ASCII text editor.

The *IMPORT* button in *TanBoDrive* (Figure 4) opens a standard Windows file selection box and waits while the Excellon file is selected. The file is searched for the two points used in the reference system; the diameters specified for these holes are not used. Once the two coordinate pairs are found by *TanBoDrive* they are used as reference points, but, of course, not drilled. It is necessary to specify in which of the two reference points the fixed pin is fitted: for this reason a thumbnail of the drilling pattern is shown in the dialog box.

Place the ready-etched printed circuit board on the turntable and select one of the twelve holes in the registration system. Fit the circuit board over the short metal pin and slide the free pin so that it fits in the second hole in the printed circuit board. Push the free pin outwards a little so that the circuit board is better centred. If the circuit board is badly warped, it is necessary to hold the corners down with adhesive tape.

In the program select in the *FixPoint* box the number of the registration hole to which the circuit board is fitted. The positions are numbered from outside (greatest radius) to inside from 1 to 12. There is no need to count: with each click the drilling pattern picture is updated so as to show the selected pin position. You can therefore check at a glance that the number is set correctly.

At this point there are two things that might go wrong. First, the circuit board might be rotated 180°: the hole shown over the fixed pin in the dialog box is in fact over the moving pin, and vice versa. Click in the check box marked *ROTATED* and the circuit board will be shown correctly on the screen.

Some programs produce drilling information as viewed from the reverse of the board (as the copper foil patterns are shown in the component mounting plans for *Elektor Electronics* printed circuit boards). Since the drill

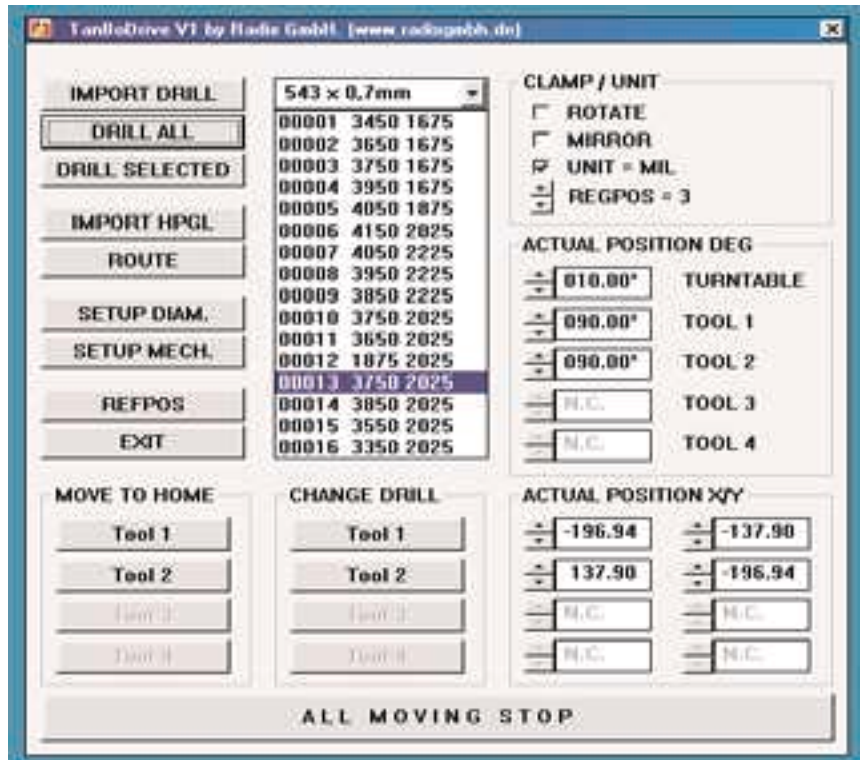


Figure 4. In the driver software *TanBoDrive* an Excellon file can be imported and the orientation of the circuit board can be set: ready to go!

tends to wander if the board is not drilled from the copper side, the *REFLECTED* check box must be clicked to correct the situation.

There is only one thing left to do: click on *START*. The drilling machine moves the first tool arm into the drill change position and displays an information box with the required diameter of drill. Fit the appropriate drill in the chuck and click on *OK*. The machine will now run, drilling at each point requiring the first drill size. If a second arm is fitted, it will meanwhile move to the drill change position.

Further requests follow, depending on how many different sizes of hole are to be made or if multiple identical circuit boards are to be drilled. It is unnecessary to describe all these here, since the software is very user friendly and practically self-explanatory. If something should go wrong, or if any difficulties are encountered, a platform where you can exchange experiences with other users and praise or berate the developers of the machine is provided at the Radix homepage.

Note:

Kits and individual parts for the PCB Drilling Machine are supplied by the author via the Radix GmbH company in Germany. At the time of writing, negotiations are under way with C-I Electronics to handle international distribution of the mechanical kits, possibly including the Elektor Electronics driver board. Further information from C-I Electronics, P.O. Box 5514, NL-3008-AM, Rotterdam, The Netherlands. Fax (+31) 10 4861592, email dil@euronet.nl. Website www.dil-dos.nl

(010024-5)