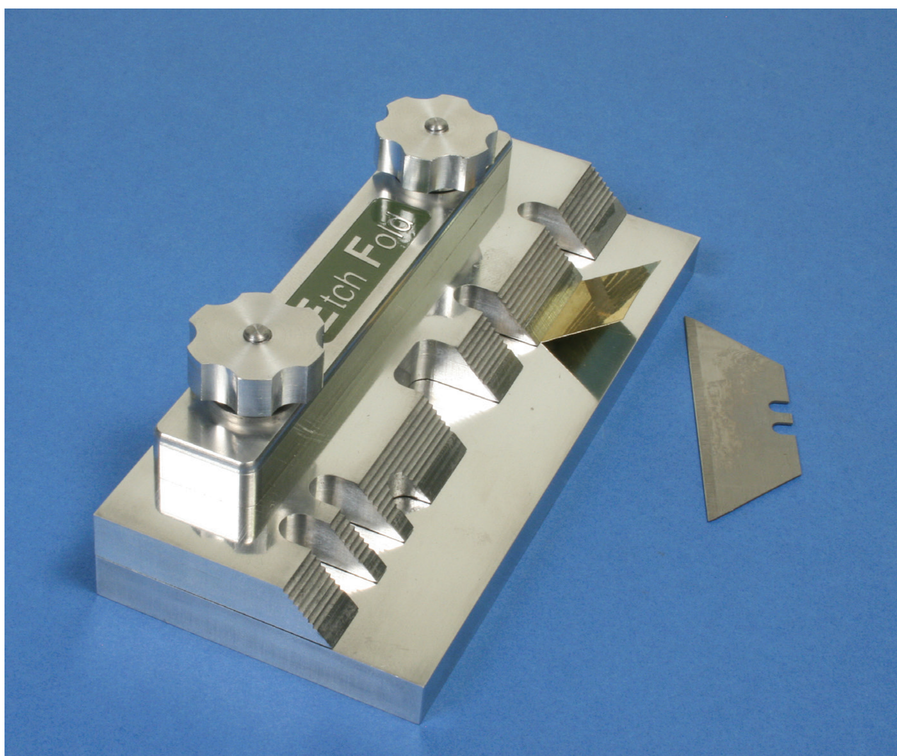
CNC in the Workshop



CNC Etch Folder

This document contains all the material on the Etch Folder project which was published in MEW issues 214 to 218 (Parts 12 to 16 of the CNC in the Workshop series).

Etch Folder

This document contains all the material on the Etch Folder project published in MEW 214 to 218.

THE ETCH FOLDER

Our last project was the vice stop, designed to illustrate surfacing and program creation. This time our project is about paths, straight lines and circles. There's a bit of arithmetic as well, to make life a little easier.

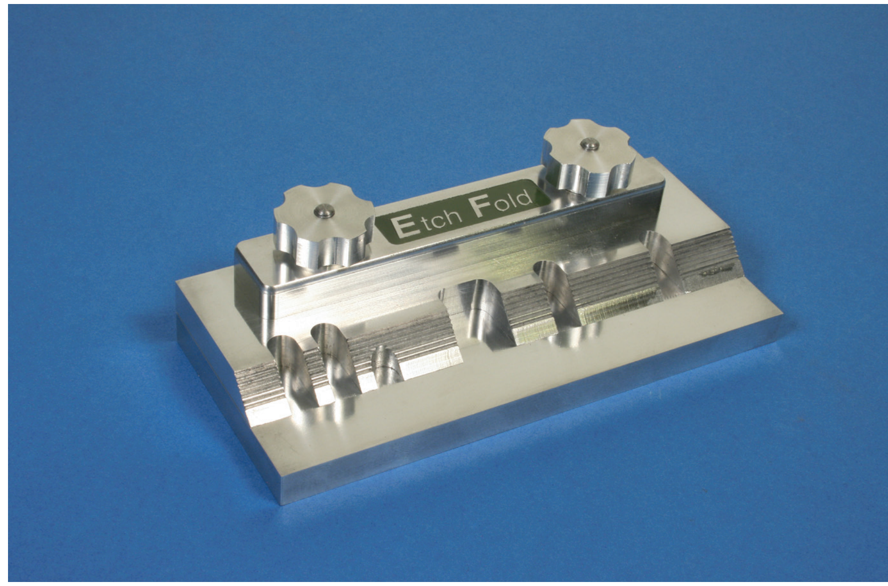


Photo 85: CNC Etch Folder.

Photo 85 shows a device which goes by many names, all jealously guarded trade names, so we'll call it a CNC Etch Folder. If you have to fold thin brass or tinplate, or thin etches for models (photo 86), you may find this little folder useful. The base is a simple rectangle which has two protruding studs. The clamp has several fingers to hold the brass as you fold it. Folding takes place using a razor blade or thin ruler or a sturdy piece of thin sheet, as shown in photo 87. The fingers allow the sides of an open box to be folded into a more or less complete empty box, with previously folded sides swinging clear in the gaps between the fingers.

This particular folder is made of aluminium, so it has a loose reinforcing bar clamped on top of the main clamp bar, for extra strength. If you have some heat treated aluminium, use that for extra strength in the fingers. You could even use steel, if you like, by adjusting the machining feeds and speeds. I made mine of aluminium because I had some. Besides; etches are normally relatively small, and the material is weakened along the intended fold lines by etching part of the way through the material, as the etch is made. That means the forces involved in bending are low.

None of the sizes are at all critical, although it is useful if the fingers are of different widths.

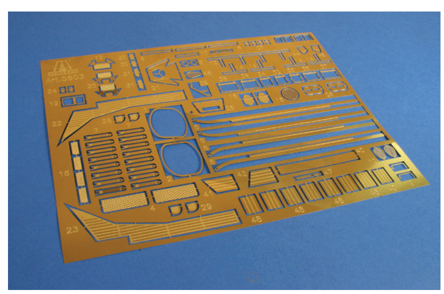


Photo 86: A sheet of etched parts for a model boat. Indiviudal parts are cut out, then folded into shape.

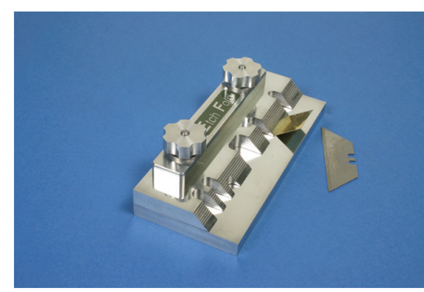


Photo 87: Thin brass folded in the folder, using a knife blade as a lever.

Fig 62 (next page) shows the principal dimensions of the main parts.

Of the three main parts, the base rectangle has four 90º corners; the reinforcing bar is similar but has rounded corners; and the clamp plate has a combination of 90º corners and rounded cut outs, so there's a mixture of shapes here; none of them complex. The knobs have an interesting shape, but they can be produced with relative ease.

In an ideal world, it would be nice to produce everything at one go, from a single sheet of 10mm aluminium, but the overall sizes make the front-to-back dimensions of a single sheet just a little too great for the average small mill. Instead, we will mill them individually, from 3 pieces of sheet (photo 88). That means we need to think carefully about how we will hold the workpieces so that we can machine them efficiently.

This is a perpetual challenge, and although a vice provides a general purpose solution, it's not applicable here because the most

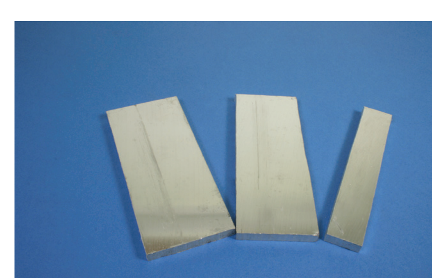


Photo 88: The main parts are machined from separate sections of sheet.

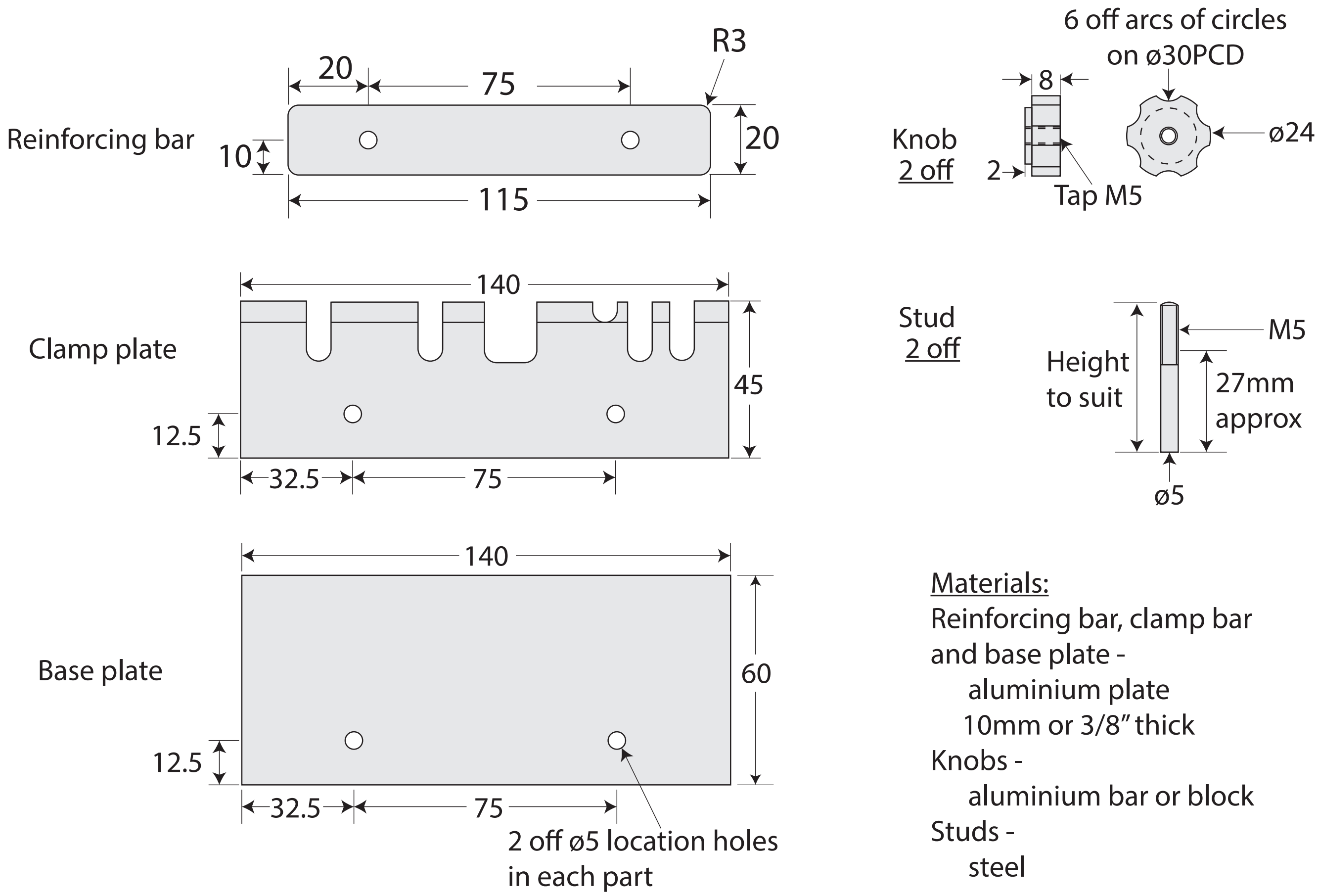


Fig 62: Principal dimensions of the main parts of the folder.

efficient way of machining these shapes is to machine all the way around the periphery at one go, treating it as one continuous path. One solution is to fix the parts to a plate designed to hold all three parts, one at a time. There are two identically spaced holes in each part, and they could be used to locate and clamp the parts clear of the machining paths. This kind of plate is called a fixture, which I define as a device used to locate and hold work in a known position, for machining. The fixture can be considered as a sacrificial surface, allowing the end of the cutter to pass a little below the lower face of the workpiece being machined; or the work can be clamped onto some spacers or a thin sacrificial sheet so that the cutter does not touch the plate itself.

The easiest way to secure the workpieces might be to use set screws or studs in tapped holes in the block. But while set screws will clamp the plate in position, they do not provide a sufficiently accurate way of locating work. A good general rule is: pins to locate, and screws to clamp. Pins in reamed holes provide repeatable location and prevent sideways movement of the work. That means if the work is taken off the fixture, it can be put back in exactly the same position. Screws resist movement of the work under the forces generated by machining, keeping it clamped firmly to the surface of the fixture.

I have some steel plate 16mm thick, so that's what I have used for this job, and fig 63 shows the layout of the holes for suitable pins. Because the whole periphery of each workpiece will be machined in one operation, we can't use conventional clamps to secure the work, but we can use two tapped pins (fig 64 and photo 89) to provide clamping as well as fixing the location of the work. You could use thinner material for the fixture, but less than 10mm risks the pins tilt-

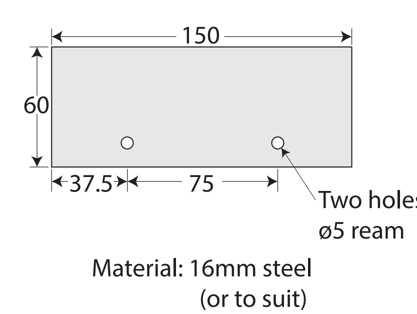


Fig 63: Layout of the fixture and the reamed holes for locating pins.

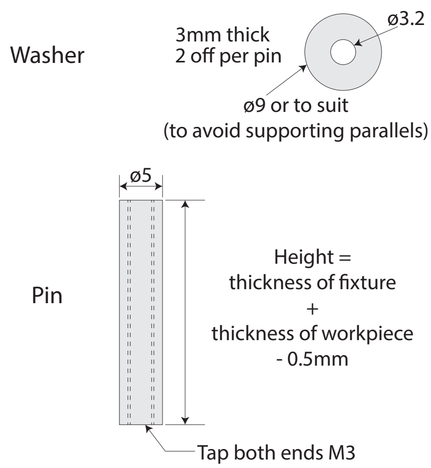


Fig 64: Suggested dimensions of the tapped pins and accompanying washers.

ing, with loss of position and repeatability. Although wood or particle board is useful as a backing material for drilling or cutting into, it is not suitable for fixtures because of its relative softness. Location pins are more secure in a metal plate, and threading wood is a non-starter. Even using woodscrews to clamp material to a wood backing is not a good idea because those screws will move under a sideways force; and repeated insertion and removal will quickly result in a very loose grip. The same would be true of plastic.

PREPARING THE MATERIAL

Cut the material for the base, clamp plate and reinforcing bar, leaving at least 3mm extra on the length and the width but note that the reinforcing bar should be cut to the same length as the other plates at this stage, for convenience (photo 88). We need two securing holes in each piece of material, for the fixture, at 75mm centres, 12.5mm in from what will be the finished edge of the workpiece. Make a small plate 140 wide (left to right) and 60 broad (back to front) using aluminium, steel or wood. I used MDF (photo 90). Hold it horizontally in the vice by sitting it on parallels, and set the Work Origin X0 Y0 at the front left corner. The sequence of operations is to clamp each piece of

Photo 89: Tapped pins to locate and clamp the work.

material on the plate, in turn, then drill and ream the holes. Begin by drilling two clearance holes of at least 5mm diameter, at X38.5 Y12.5, right through the material – but make sure you don't drill into the parallels. I normally use "standard" thickness parallels 8mm thick, but jobs like this suggest thin parallels might be a good idea.

Place the material for the base plate onto the plate in the vice, arranging it so that there is approximately 1mm overhand over the left edge and over the front. Check the other sides overlap, so that when finally machined a cut will take place on all sides, leaving the finished base plate 140 wide and 60mm broad. Clamp the work securely, making sure the clamps are clear of the hole positions, then centre drill, drill 4.9mm diameter, and ream the two holes. Repeat the process for the material for the clamp plate and the reinforcing bar. There are lots of other ways of doing this job, so feel free to use a method that suits you best.

PREPARING THE FIXTURE

To prepare the fixture plate, cut it a little oversize, select the best long side, and sit the plate down on that side, in the vice (photo 91). Use a facing cutter or large diameter end mill to skim the upper face (the top edge) level. Then turn the plate over so that the faced edge lies on the base of the vice, and skim the opposite edge. You may wish to check the resulting width (i.e. the distance between the two faced edges) and mill off just enough to bring the plate to a

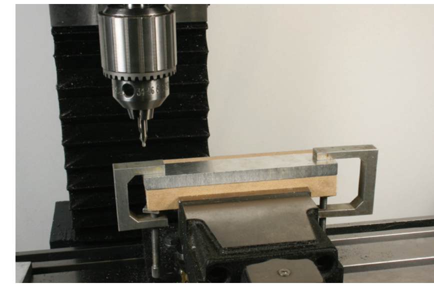


Photo 90: MDF plate in position to support the main parts as the locating holes are drilled and reamed.

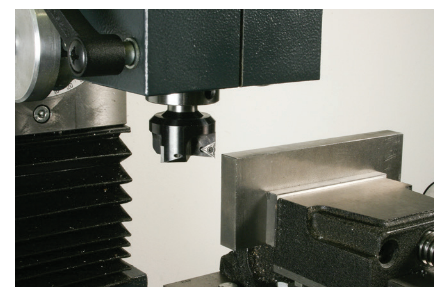


Photo 91: Skim the upper face to form the side of the fixture.

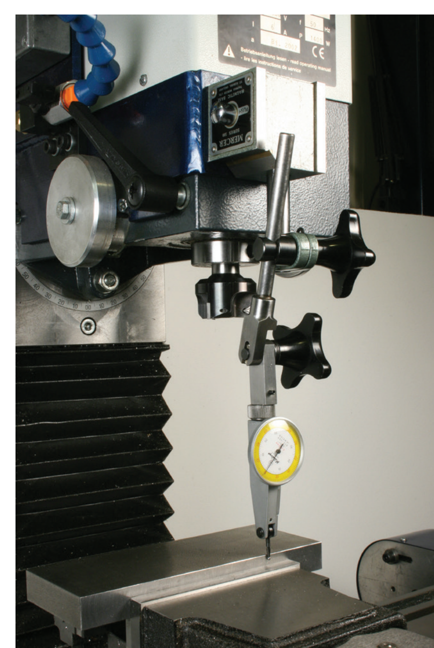


Photo 92: A lever-action DTI is used to clock the front edge of the material parallel to the X axis.

convenient size for your vice. I chose 60mm breadth so that the plate could be gripped easily in my Vertex K4 vice. That may mean the etch folder base plate will overhang the

fixture a little, but that is not important.
Sit the fixture plate on parallels, with the top surface just a little clear of the top of the vice jaws and with the ends clear of the sides of the jaws. The next operation is to trim the ends, but it is important that the left end is at right angles to the front face of the plate, so use a lever operated dial test indicator (DTI) to clock the front face parallel to the X axis (photo 92). Get it as parallel as you can, by rotating the vice slightly until there is no more than a tiny change in the DTI reading from one end of the front face to the other. This can be done relatively quickly, but it can also turn out to be a test of your patience, so take your time. Start with the DTI lever at the centre of the face, and move to one end. Note the reading, then swivel the vice to reduce that reading. It works best if you try to pivot the vice around the point where the DTI foot started. Re-test, and repeat; until there is no change in the reading from centre to end. Then check the reading from one end to the other.
Once you are happy, nip everything up tight, remove the DTI and trim the ends using the side of a large diameter end mill. Mark the front left corner of the fixture as the reference corner, so that we can locate accurately off the front and left edges of the plate.
Lever-action DTIs tend to have a small range of movement, so if you are not practised in the art of "clocking", you may find frustration levels can be quite high, at first. My plunge action DTI has a much greater range of movement, but I found it impossible to site it so that it would sit horizontally with its ball foot against the edge of the work. Another way to approach the task is to use a laser centre/edge finder; set the spot to the edge of the work, at the centre of its length, then traverse towards the outer end, watching the red spot as you go. I find it relatively easy to tap the work into alignment with the spot. Any small misalignment which remains will be well within the range of the lever action DTI.
If you made a setting plate for your vice (MEW 207) you can use that to sit the vice in an initial position which locates the jaws very close to parallel to the X axis, but it is worth using the DTI to check the front face of the work to get it dead on. That will, after all, be a reference face for the fixture, and the more accurate its orientation at this stage, the more accurately you can position the machined features of the final workpiece.
Set X0 Y0 at the front left corner, then move to X5 Y0 and set X0 Y0 there. The front left corner will do fine as the Work Origin X0 Y0, but my plate is 150mm end-to-end, while the workpieces are 140 broad, so it is more convenient to set the Work origin 5mm from the left end, to coincide with what will become the front left corner of two of the finished workpieces. Drill two 4.9mm holes at X32.5 Y12.5 and X107.5 Y12.5, then ream those holes 5mm diameter. Being lazy, I simply used MDI mode to position the Controlled Point (CP) then manually centre drilled, drilled, then reamed each hole. Watch out for the parallels under the plate! Take care, too, that the reamer doesn't bump into the base of the vice. Use a machine reamer if possible, although there is just about enough room to use a hand reamer (mounted in the chuck, of course). Run the reamer at no more than 1/3 of the speed of the equivalent drill, and use copious lubrication.
Make up the two pins (fig 64) and their washers, in the lathe. Make the pins up to 0.5mm shorter than the combined thickness of the fixture and the work. When the

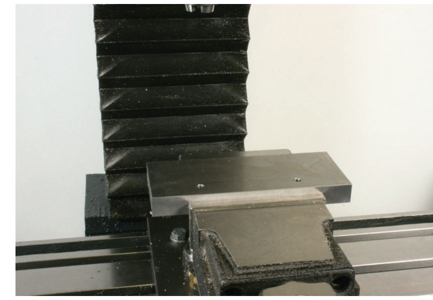


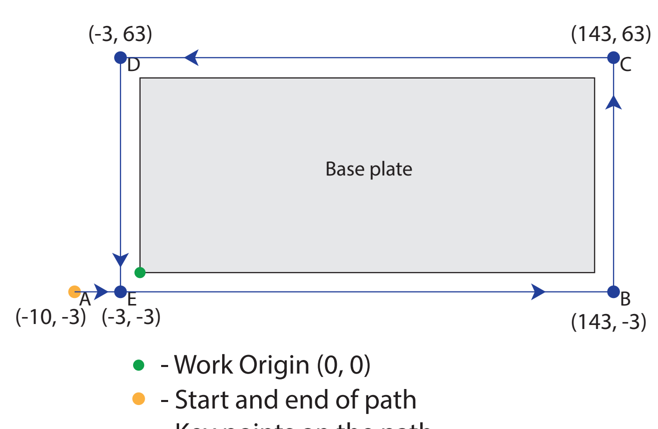
Photo 93: The completed fixture, mounted in the vice.

sacrificial material is added, that should make the pin 0.5 – 1.5mm shorter than the combined sandwich. Arrange the diameter of the washers so that they just clear the parallels.
Photo 93 shows the completed fixture.

MACHINING THE BASE

Assuming the material for the base has been cut to within a few millimetres of the finished size, the basic shape could be machined using four G1 commands to take the cutter along each side to the corner, turn, then travel along the next side. Assuming the Work Origin is at the front left corner, fig 65 shows the path around the shape, taking account of the radius of the cutter.
I will assume a 6mm diameter cutter, although a larger diameter would be more rigid and would make light work of aluminium. But my 6mm cutter is specifically for machining aluminium and does a terrific

Fig 65: The path of the CP around the base.



job. The flutes are very highly polished to prevent the swarf binding onto the cutter, and the cutting angle of the spiral faces has just the right amount of aggression. Lovely tool; I like it a lot.

Ignoring Z feeds for the moment, the basic path for the CP might be:

```
G0 X-10 Y-3
G1 X143 Y-3
G1 X143 Y63
G1 X-3 Y63
G1 X-3 Y-3
G0 X-10 Y-3
```

But let's be a bit more sophisticated, and improve the performance at the corners by eliminating some of the decelerate-accelerate cycle as the CP changes direction abruptly. Instead of simple G1 commands to move in a straight line, add a G3 command to steer the CP around each corner. That way, when the software looks ahead to see where to go next, and whether to stop and change direction, it will see a smooth change, and will not attempt to decelerate to a stop each time. The centre of the arc for the G3 command will be the point of the corner, and the arc will span 90º, so the X and Y co-ordinates of the start and end points will be relatively easy to work out.

Fig 66 shows the new path, and the code might look like this:

```
G0 X-10 Y-3
G0 X-3 Y-3
G1 X140 Y-3
G3 X143 Y0 I0 J3
G1 X143 Y60
G3 X140 Y63 I-3 J0
G1 X0 Y63
G3 X-3 Y60 I0 J-3
G1 X-3 Y0
G3 X0 Y-3 I3 J0
G0 X-10 Y-3
```

Now that we have our path, it's just a matter of cutting, lowering the CP (Z), cutting that same path, lowering the CP, and so on. That's what we have been doing up to now, and it is a perfectly workable method. But the smaller the vertical cuts, and the thicker the material, the more we have to repeat the same path over and over again, with only the Z values changing each time. There is a better way. There are three parts to the solution, and each is a very powerful addition to our programming repertoire, so take a deep breath and let's dive straight in. Splash.

Make a subroutine

Let's take each cut around the path by lowering the CP (Z) then using the code shown above to go around the path. If that action of lowering and cutting is repeated enough times, that sequence will do the job. That's the technique we used in our last project, so there's nothing new in the concept. For the moment, just to make life easy, we will forget about ramping into the work. One step at a time.

Instead of having to type the code for the path each time, it can go into a subroutine. A subroutine is simply a section of code we can use as many times as we like. We only need to type the code once, but can use it at any time, and use it repeatedly if need be. To identify the section of code, it has an identifying number at the beginning, and a specific command at the end. Here's what it might look like:

O100 (The start of the subroutine with reference number 100)
---- put the code to machine the periphery here ----
M99 (signals the end of the subroutine)

In O100, the O is a capital letter and the 100 is a number.

M99 is always the command used at the end of a subroutine; the number doesn't change. Although we gaily talk about the G-code language, and although the majority of commands do begin with G, the O code denotes a flow control command related to the logic of the sequence of commands in a program. There are M codes, F, S and T codes too, so why we favour G would be a mystery unless we knew that it is entirely a co-incidence and has nothing to do with the fact that some commands begin with the letter G (see the notes on the support website).

Reference numbers can be anything in the

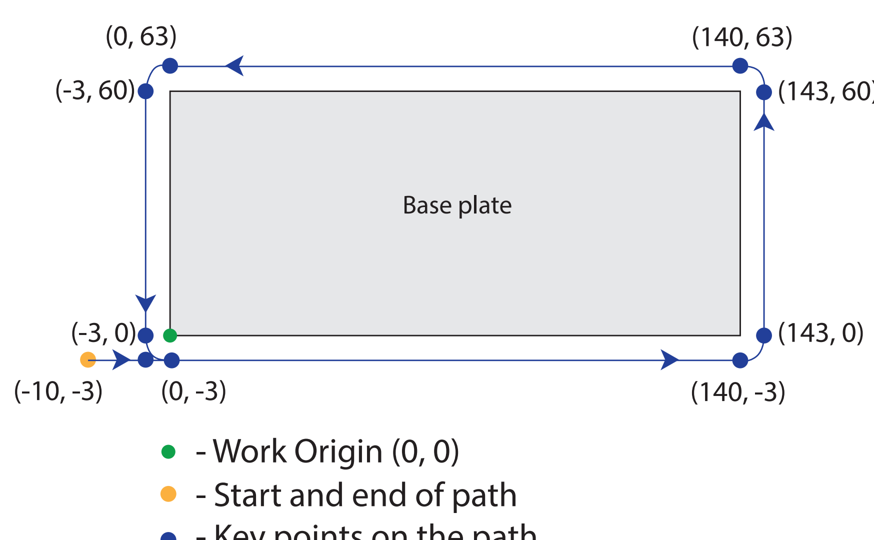


Fig 66: A different path around the base, turning corners using arcs to improve cutting performance and look-ahead trajectory planning.

range 1 to 9999.
To avoid confusion, subroutines go after the M30 statement at the end of a program so they do not appear in the normal flow of commands in the main program.

To use a subroutine, use the command M98 P~ substituting the reference number of the subroutine in place of the ~ character. So M98 P100 would call up the code from subroutine 100 and carry it out. When it has finished (i.e. when the subroutine reaches the M99 command) the program interpreter returns to where it was in the main program and carries on from there.
This program will repeatedly lower Z and cut the periphery:

```
G0 Z20 (Safe Z)
G0 X-10 Y-3
G1 Z-1
M98 P100
G1 Z-2
M98 P100
G1 Z-3
M98 P100
… and so on

M30

O100
---- put the code to machine the periphery
here ----
M99
```

That has the same sequence as the programs we have used before, but the structure is neater and, as we will see, it allows us to use other powerful techniques.

Clean code

Using the subroutine is termed "calling" the subroutine; so M98 P100 calls subroutine 100. It's a computer programmer's term, really, but useful nevertheless, because it is generally understood, at least in the word of code.

One of the important things about subroutines is: clean entry / clean exit
That means the subroutine should begin its movements from a predictable start point which must be the same each time. It should finish at a predictable end point, which must be the same each time.
The only exception to this rule is when we have deliberately created just that: an exception. Even then, there must be a reason for that, and the start and end points each time the subroutine is called must be predictable. The subroutine should leave no loose ends, and should cause no unexpected effects for the rest of the program.
In practice, this means we need to consider where the CP is before we call a subroutine, and think about how to get to a suitable position before the subroutine does its work. Similarly, at the end we need to think about where the CP will be and whether we need to do some additional moves before continuing with the other parts of the program. The more of this that can be done inside the subroutine, the better.

Improving the final finish

The program should do its job quite nicely, but we can easily improve the surface finish by taking a fine finishing cut in the opposite direction, using climb milling.
To do that, move the CP outwards a little for the initial roughing cuts (perhaps 0.1 or 0.2mm), use the subroutine technique as before, then take a final cut at full depth moving the CP clockwise and cutting to final size.

```
(Machine the work 0.1mm oversize)
G0 Z20 (Safe Z)
G0 X-10.1 Y-3.1
G1 Z-1
M98 P100
G1 Z-2
M98 P100
G1 Z-3
M98 P100
… and so on, finishing with a cut at Z-10
(Machine the work to final size by climb milling)
G0 X-3 Y-3
G0 Z-10
G1 X-3 Y60
G2 X0 Y63 I3 J0
G1 X140 Y63
G2 X143 Y60 I0 J-3
G1 X143 Y0
G2 X140 Y-3 I-3 J0
G1 X-3 Y-3
G0 X-10 Y-3
G0 Z20 (Safe Z)
M30

O100(Machine the work oversize, cutting anti- clockwise)
(Entry assumes X-10.1 Y-3.1)
(Exits with X-10.1 Y-3.1)
G1 X140 Y-3.1
G3 X143.1 Y0 I0 J3.1
G1 X143.1 Y60
G3 X140 Y63.1 I-3.1 J0
G1 X0 Y63.1
G3 X-3.1 Y60 I0 J-3.1
G1 X-3.1 Y0
G3 X0 Y-3.1 I3.1 J0
G0 X-10.1 Y-3.1
M99
```

Complete the program using the basic structure outlined in MEW 209, with Comments, an Initialisation Sequence, a Main Program and an End command, following that with any subroutine definitions.
Note that if your workpiece is sitting on a sacrificial sheet at least 0.6mm thick it won't matter whether you are using 10mm or imperial thickness 3/8" (9.525mm) material. The additional descent will simply be into the sacrificial sheet.

Mount the workpiece in the fixture, set the Work Origin with Z0 at the top surface of the material, jog the cutter above the work, and run the program to machine the periphery of the base of the folder.

MACHINING THE REINFORCING BAR

Like the base plate, the best way to deal with the reinforcing bar (fig 67) is to bolt the material to the fixture as shown in photo 94 (not forgetting the sacrificial material) then machine around the periphery. The bar is a little shorter and less broad than the base plate or the clamp plate, and it is symmetrical about its centre point which lies on the line between the two locating pins, so it is convenient to move the Work Origin to the centre of the bar. This is not essential, but it makes life easier when we have to identify the co-ordinates of the corners and the path of the cutter. There are some sophisticated ways of moving the WO when we are working with several workpieces, but, for now, it is enough to use MDI mode to move the CP to X70 Y12.5 and set X0 Y0 at that point. We can easily return to our original WO position for the clamp plate, later.

We can take a similar approach to machining the periphery as for the base plate but improve the way we handle the Z values. With the Work Origin set to the centre of the bar a suitable path around the periphery for a roughing cut 0.1mm oversize all around might be as shown in fig 68, created using the following commands:

G0 X-54.5 Y-19.1 (Set initial position clear of the work)
G2 X-54.5 Y-13.1 I0 J3 (Entry curve to reach the work)
G1 X54.5 Y-13.1
G3 X60.6 Y-7 I0 J6.1
G1 X60.6 Y7
G3 X54.5 Y13.1 I-6.1 J0
G1 X-54.5 Y13.1
G3 X-60.6 Y7 I0 J-6.1
G1 X-60.6 Y-7
G3 X-54.5 Y-13.1 I6.1 J0 (Turn to complete the periphery)
G2 X-54.5 Y-19.1 I0 J-3 (Exit curve to take cutter clear)

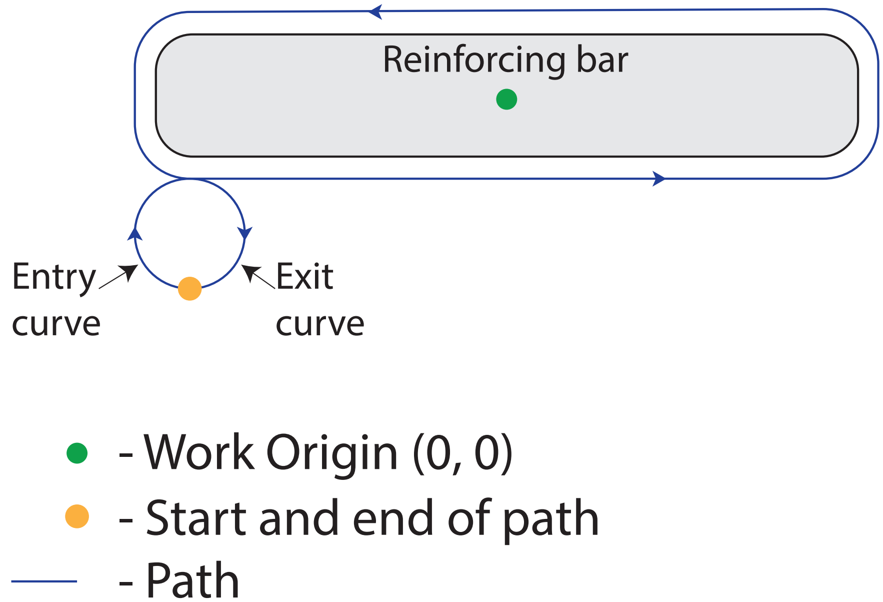


Fig 67: A machining path to form the Reinforcing Bar.

The first G3 command is used to provide a smooth entry curve into the cut, avoiding any noticeable marks caused by an abrupt change of direction and the consequent decelerate – stop – accelerate sequence. The final G2 command provides an exit curve, taking the cutter smoothly clear of the work.
Fig 68 shows how these entry and exit curves work.

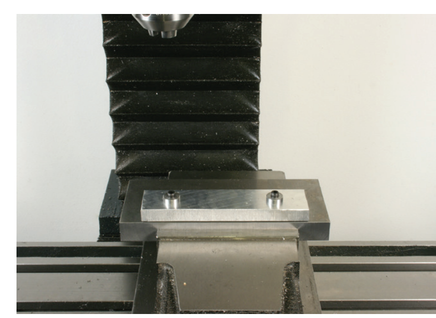
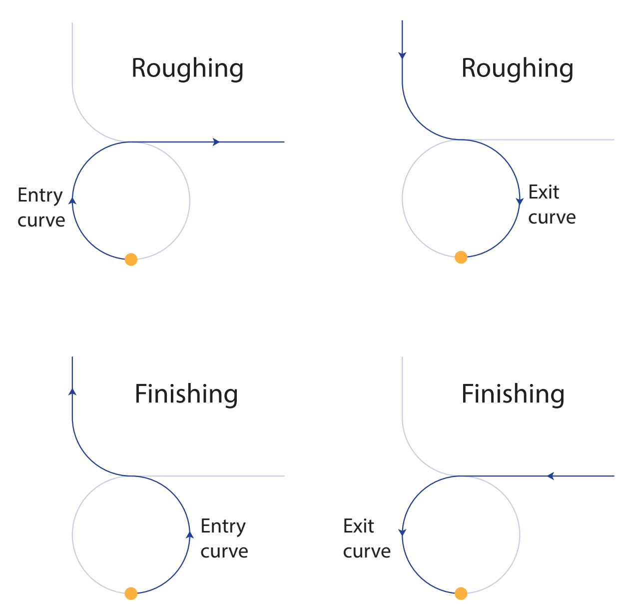


Photo 94: Workpiece screwed to a sacrifical base.

Fig 68: Entry and exit curves.



Arithmetic

Mach3 can do arithmetic, and that's useful here because we can make it add a number to the existing value of Z each time we use the subroutine. Repeatedly using the subroutine will then make it take successively lower cuts, dealing effectively with the need to lower Z each time. All calculations must be enclosed in square brackets.
Mach3 can cope with statements like:
[4 + 5] or [6 – 2]
It can also deal with two signs together, so sums like:
[4 -2] or [6 - -2] are fine. If you recall the rules we used in a previous instalment for dealing with two signs together, Mach3 knows and uses those same rules. Do use square brackets, though. Extra pairs of brackets are no problem; but too few brackets will cause errors.

If the answer to a sum is to be used right away, that's fine, so statements like:
G1 X[3 + 4] will be carried out without difficulty.
But we often need to do a calculation which won't be used straightaway, so the other thing we need is somewhere to store the answer to a sum, and we do that by using the equivalent of a storage box. To be compliant with the language of computer programming, the storage box is called a parameter (although there is some variation across computer languages, and you may find other terms used in software other than Mach3).
So that we can refer to a parameter later (to retrieve the answer to the sum) each parameter is given a number, in much the same way that each subroutine is given a number. Use the hash sign # to denote a parameter, and an integer (a positive whole number) to identify which parameter. So #252 means "parameter 252". Avoid parameter numbers less than 40 or greater than 4999, for the moment, as those behave in a slightly different (but sometimes very useful) way.

If we use a parameter to store the current Z height, we can change the value stored there each time the subroutine is called.
Here's an example:

```
#41=0 (Parameter 41 stores the current Z value)

G0 Z20 (Safe Z)
G0 X-54.5 Y-19.1
M98 P115
M98 P115
M98 P115
… and so on, until the subroutine has been called 10 times
M30

O115 (Subroutine to machine periphery 0.1mm oversize)
#41 = [#41 – 1] (Make the contents of parameter 41 one less)
G1 Z#41 (Set Z to the number stored in parameter 41)
---- Put the commands to machine the periphery 0.1mm oversize here ----
M99
```

Every time the subroutine is called, the value stored in #41 is reduced by 1. The effect of successive calls to the subroutine is to machine repeatedly around the periphery, 1mm lower each time.

We can go further, though, by using an extended form of the M98 subroutine call command, using the letter L followed by the number of times the subroutine is to be used. Think of L as meaning "Loop" in the sense of "repeat".

M98 P115 L10 calls the subroutine 10 times in quick succession, so it is exactly the same as typing M98 P115 ten times. The main program code now becomes:

```
G0 Z20 (Safe Z)
G0 X-54.5 Y-19.1
M98 P115 L10
M30
```

Add additional program code to machine the periphery to finished size by climb milling (as we did for the base plate). You only need to travel around the periphery once, at full depth, for that final cut.
There is a finished program on the support website, if your brain cells are as tired as your typing fingers.
Secure the material to the fixture, then run the program. Photo 95 shows the quality of finish you might expect to achieve from the roughing cut (on the right face) and the finishing cut (on the left face).

Photo 95: The surface finish we are aiming for.

Adding a rounded edge

The reinforcing bar will work perfectly well with sharply defined edges, but in an attempt to visually soften the outline, and to reduce the opportunity for cut fingers, it might be an idea to round or bevel the edge around the top of the bar. That can be done now, or the bar can be returned to the fixture later. I suggest doing it now.

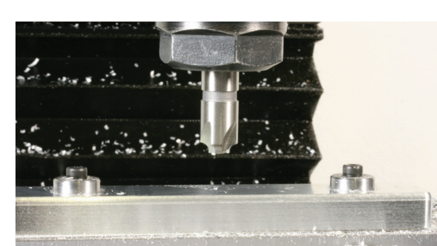


Photo 96: A rounded edge cut using a form cutter.

To round edges, you can use a corner-rounding end mill (photo 96) which forms a neat

radius around the edge as it cuts.
The geometry of this kind of cutter is interesting, as the angle between its "wings" is slightly greater than 90^0, making it a little easier to blend the radius into the vertical and horizontal faces of the work (fig 69).

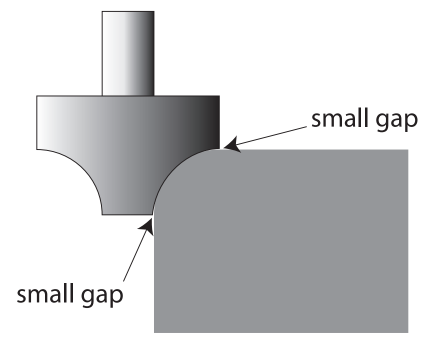


Fig 69: If the angle between the 'wings' is more than 90 degrees, it is easier to blend the sides forming the corner.

Four effects are possible:
a smoothly blended corner (fig 70a), a corner with one quirk (a step or ledge between the curve and an adjacent face) on the horizontal face (fig 70b) or the vertical face (fig 70c), or a corner with two quirks (fig 70d). The effect you use is purely an artistic choice. What's important is that we know how the radius is positioned in relation to the end and side of the cutter (fig 71). I used a 2mm

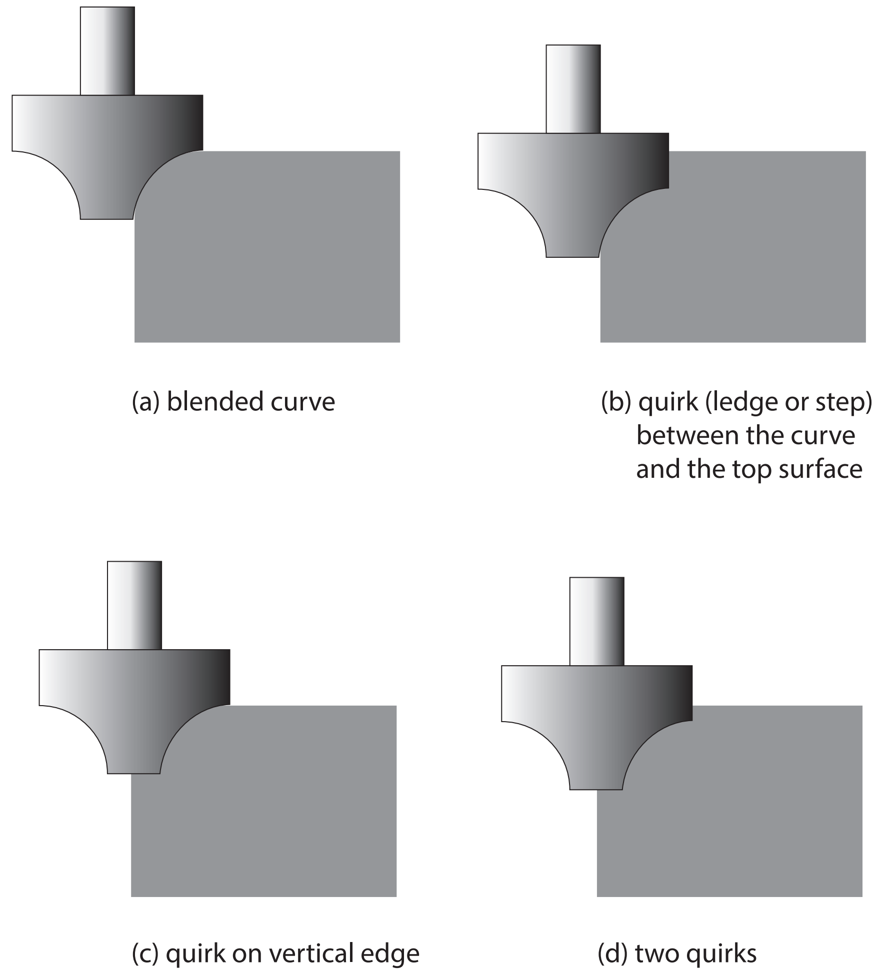


Fig 70: Four different corner effects are possible using the same cutter.

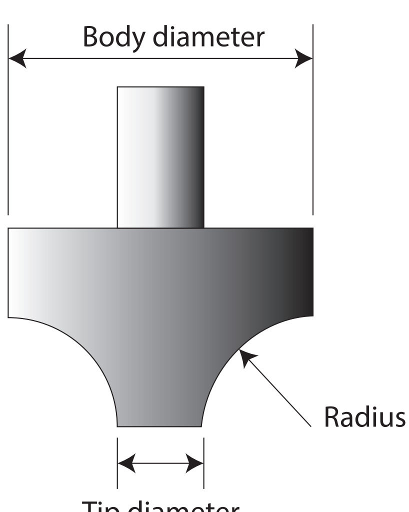


Fig 71: The essential dimensions of the corner cutter.

radius cutter. That means the radius of the internal curve on the cutting edge is 2mm. Assuming the radius of the cutting face is centred on a point co-incident with the bottom face and the outer diameter of the cutter, you can do a visual check of the curve by holding a 4mm diameter twist drill against the cutting edge, but you can also do a calculation as shown in fig 72. Some suppliers list not only the radius of the curve, but the tool diameter and the tip diameter, so if their tools have been ground accurately, your measurements should merely confirm the supplier's figures.

Now that we know the position of the offset of the curve in relation to the vertical axis

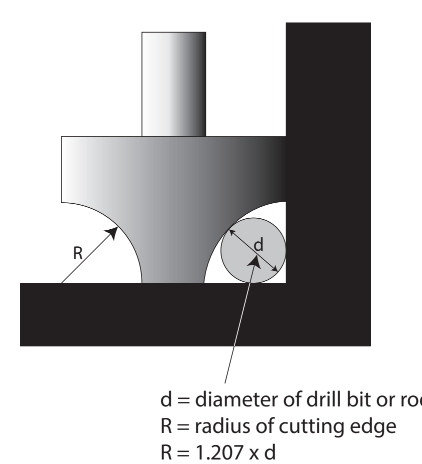


Fig 72: Calculating (or checking) the radius of the cutting edge using the shank of a drill or a rod of known size.

of the tool, we can use that information to work out the path of the CP. Fig 73 shows that for the cutter I used, the CP should run 6mm outside the periphery of the clamp bar, and the maximum Z distance should be 2mm.

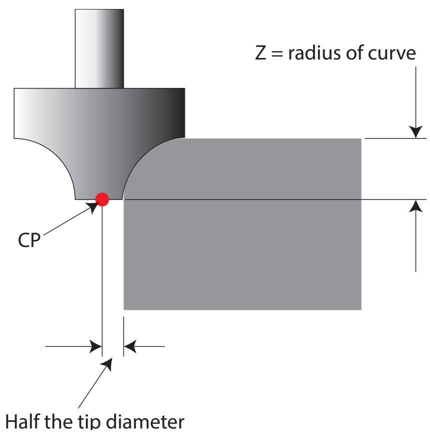


Fig 73: The CP is offset from the front face of the work, by half the tip diameter of the cutter.

Modify the program you used to machine the periphery, to take perhaps 4 cuts downwards to form the periphery 0.1mm oversize, then take a final climb cut to finish to size. The surface finish will not be as good as with the single flute straight cutter, but it should be pretty good. I found my cutter left an almost imperceptible witness around the vertical side of the bar, so I will take account of that next time I use that tool.

We will return to the important business of tool dimensions, shortly.

As with the other programs, there is a completed example on the support website.

Reset the Work Origin

Before moving on to machine the clamp plate, take the CP to X-70 Y-12.5 and set the Work Origin there, corresponding to the left front corner of the finished clamp plate. That undoes the move we carried out to shift the WO for the reinforcing bar.

MACHINING THE CLAMP BAR

The clamp bar presents an interesting design challenge. The nose is swept back at 45o, to allow over-bending of an etch, beyond 90º. The problem is the nose needs to retain some strength, so there is a tiny flat on the nose, of 2mm, and the rest of the face needs to support that nose while staying out of the way of an etch. The swept face could be a plain surface at 45º to the vertical but it will be over 14mm long (measured up the slope). There's a machining challenge.

It is relatively easy to machine a 45º face, but we would need to add a technique which can be better understood with a simpler example, so let's keep that for later.

The face could be formed with an angled cutter, and 45º is a convenient angle as that is the angle of many countersinks. The complication is in the length of the face. Assuming a large enough countersink, the cutter would end up machining the whole breadth of the face, at least on the final pass (fig 74),

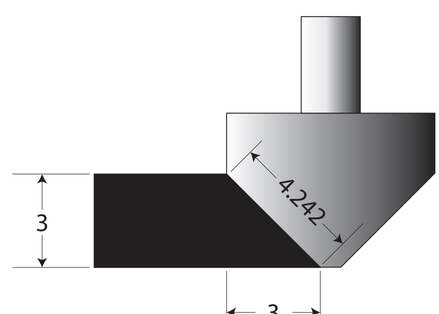


Fig 74: On a cutter with an included angle of 90 degrees, each cutting edge is inclined at 45 degrees, and the length of the face is 1.414 x the horizontal length of the cutting edge. Note that 1.414 = √2

and the tool would be liable to chatter, spoiling the finish. With a smaller cutter, adjacent passes along the face could be arranged so that the face can be finished in stages, but the problem with the witness mark from the corner rounding cutter suggests we may not get a perfect match between sections of the face. A test cut of a piece of scrap material (photo 97) confirms this, although the difference in this case is very slight. In theory overlapping cuts should match nicely, but it does require some adjustment of the distances involved, and depends on the precise geometry of the individual tool.

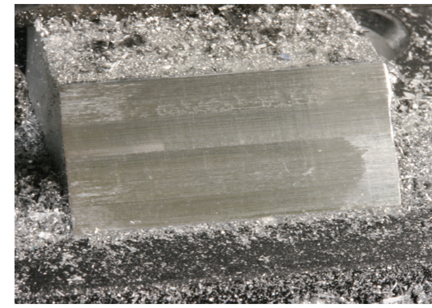


Photo 97: A test cut on an angled face.

The design could be altered to recognise the fact that while the plain angled face is important to provide support for the nose, only a short portion needs to be solid. The remainder can be a stepped face because the small reduction in the volume of material higher up the face will not significantly affect the strength of the nose, especially if the steps are small.

In the final design, that's the arrangement. Now, the 3mm tall plain section at the nose requires no more than a 4.3mm breadth of cut along the slope (i.e. the hypotenuse of a right angled isosceles triangle as shown in fig 75).

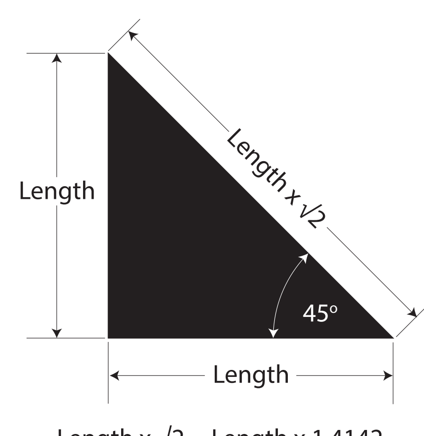


Fig 75: On a cutter with a cutting edge lying at 45 degrees, the horizontal breadth and vertical depth of the cutting edge are the same.

Machining the periphery of the clamp bar

The periphery of the clamp bar can be machined to final size, including the cut-outs on the far away face, using the same basic technique as used for the bed plate and the reinforcing bar. One sequence of operations would be:

- Trim the face which will form the front of the nose (without the steps or the angled portion) – optional.
- Machine the 7 steps, leaving a 3mm uncut portion at the nose.
- Machine the periphery.

Then change the cutter and finish the angled portion at the nose.

The first three steps can be incorporated into a single program; and the last step can be performed in MDI mode or by using a separate short program.

Fig 76 shows the overall dimensions of the clamp bar, omitting the detail of the steps and the cut outs, which are given in subsequent figures. The machining sequence is straightforward enough, although it is somewhat time consuming to define co-ordinates and create the appropriate commands. By the time you have finished this program you will deserve your Master of Arcs and Circles badge. Professorial status lies well beyond that, of course.

Trim the front edge by starting clear of the right hand end of the work, 0.1mm beyond the finished Y position of the edge:

```
#41=0 (Parameter 41 holds the Z depth of cut)
F100
G0 X150 Y48.1 (Y position for roughing cut)
M98 P120 L10
G0 Z-10
G1 X-10
G0 Y48 (Y position for finished size)
G1 X150
G0 Z20
```

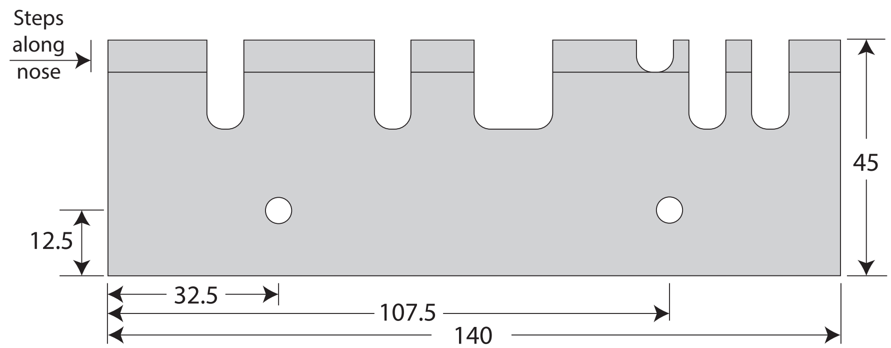


Fig 76: Overall dimensions of the clamp bar.

After the End command, define subroutine 120:

```
O120 (Trim the front edge 0.1mm oversize)
(Entry assumes CP at X150 Y48.1)
(Exits with CP at X150 Y48.1)
#41=[#41 - 1] (Z depth decremented)
G0 Z#41 (Down to depth of cut)
G1 X-10 (Cut along until clear of left edge)
G0 Z1 (Lift cutter clear of work)
G0 X150 (Back to the start, ready for a further cut)
M99
```

Fig 77 shows the arithmetic of the 7 steps. For material 10mm thick, the total height of the steps will be 7mm, leaving a 3mm thick x 2.8mm wide ledge at the bottom. That is 1mm per step.

If you are using imperially sized material (9.525mm thick) the total depth of the steps will be 9.525 - 3 = 6.525mm. The depth of each step will then be 6.525 / 7 = 0.9321mm In either case, the breadth of each step must be the same as the depth, to maintain the 45o angle. The tapered section at the bottom would have been 3mm wide, but it has been cut back by 0.2mm to provide the 0.2mm vertical face at the nose.

The rear edge of the CP for the top step will be at Y38.2 when using a 6mm diameter cutter in 10mm thick material, or at Y38.675 in material 9.525mm thick.

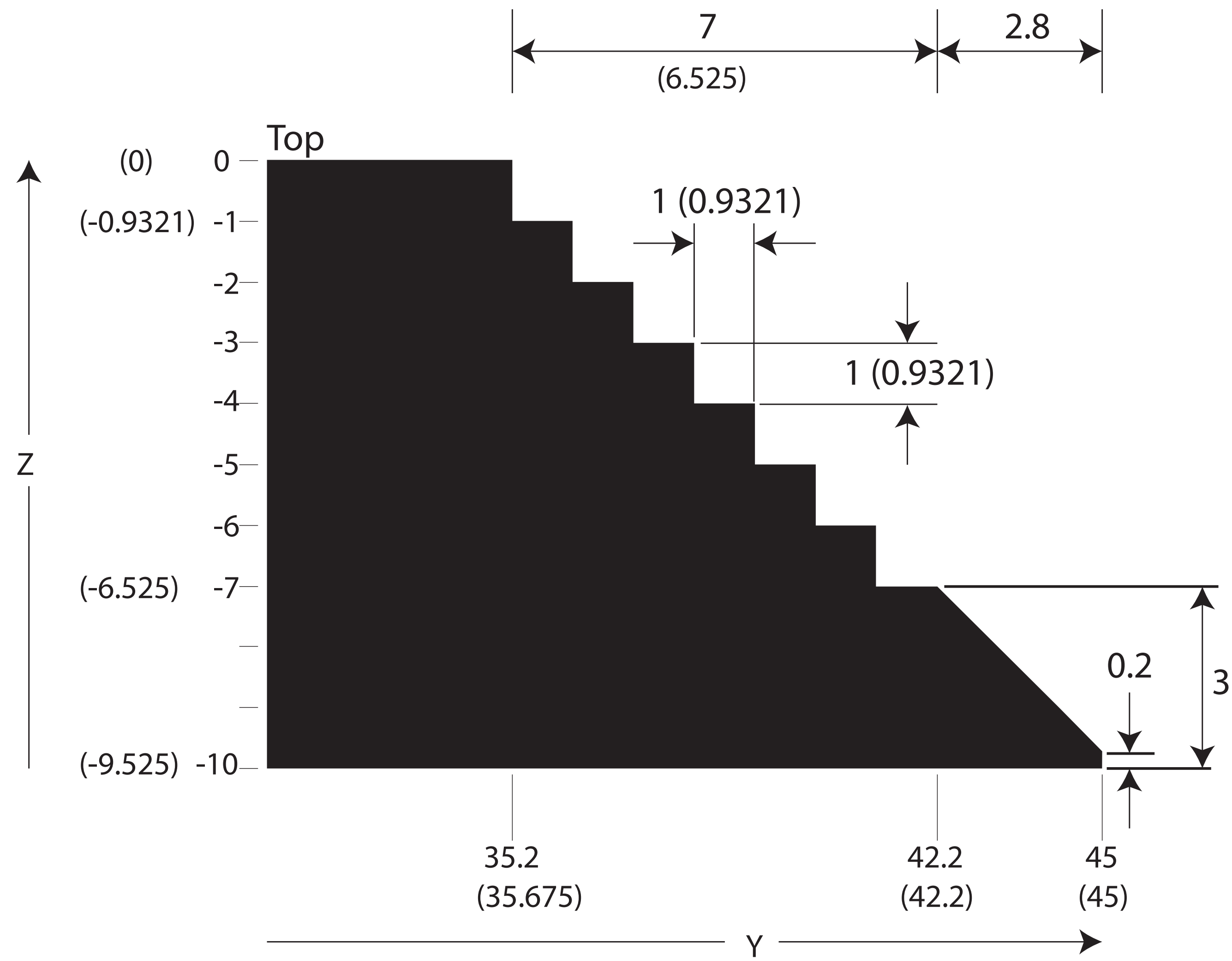
We can keep track of the Z heights and the Y distances as the CP moves from one step to another, by storing them in parameters, so #41 can hold the Z height, and #42 can hold the Y value for the CP, so that the end teeth of the cutter create the bottom of each step, and the periphery of the cutter creates the rear face of each step.

Now we need to think about how the subroutine will position the CP for each step, as there are several possibilities.

Imagine the program is part-way through machining the steps.The next time the program calls the subroutine, it should machine a step at that new position. Then it should subtract the Z feed from #41 and add the breadth of a step to #42, so that those parameters hold the Z and Y values which will be required to machine the next step. So, on entry, the Z and Y values stored in #41 and #42 should be those of the current step, and on exit #41 and #42 should contain the Z and Y values for the next step.

Before calling the subroutine for the first time, we need to put the relevant initial Z and Y values into #41 and #42 in preparation for cutting the first step. Fig 77 shows these are Z0 and Y38.2 (allowing for a 6mm diameter cutter).

There are other ways to think about this and other options for dealing with the parameters are listed on the support website.



Figures in brackets ( ) refer to material 9.525mm thick (imperial 3/8”)

All dimensions are shown in millimetres

Fig 77: Dimensions of the seven steps.

```
(Cut the stepped slope to form the nose)
(Begin at the top)
#41=0 (Z depth of cut)
#42=38.2 (Y distance: - would be 38.675 for 9.525mm material)
G0 X150 Y48
M98 P130 L7 (7 repetitions, to leave the bottom 3mm uncut)
G0 Z20 (Safe Z)
```

After END, define subroutine 130:

```
O130 (Cut the “front” profile as a series of steps)
(Entry assumes CP is at X150 Y48)
(Exits with #41 holding Z height for next step)
(and #42 holding Y value to machine the rear of the next step)
(and CP at X150 Y#42 Z#41)
F100
G0 Z#41
G0 Y[#42 + 0.1] (Increase Y distance by 0.1mm for this roughing pass)
G1 X-10 (Cut beyond the left end)
G0 Y#42 (Position for a finishing cut by climb milling)
G1 X150 (Cut back to the start)
G0 Y48
#41 = [#41 - 1] (Lower Z height for next step.)
(Alter to #41 - 0.9321 for 9.525 material)
#42 = [#42 + 1] (Increase Y for next step.)
(Alter to Y + 0.9321 for 9.525 material)
M99
```

The remainder of the periphery can be machined in the same way as for the earlier workpieces, using #41 to control the depth of cut. The difference this time is that the back edge is not straight but contains cut outs. Fig 78 shows the points which define the cut outs in the periphery; fig 79 shows detail of a typical cut out, with the important points at transitions between portions of the outline. Travel to the corner of the first (right hand) cut out; arc around the corner; cut the straight portion. The bottom of that cut out consists of two quarter circles with a short straight line between the two, so cut around the curve (the right hand quarter circle); cut straight along to the beginning of the next curve; cut around the curve (the left hand quarter circle); cut the straight portion to the corner; then arc around ready to begin the sequence for the next cut out.

With quite a number of important points on the whole periphery an organised approach is required. Good organisation also simplifies the process.

You might use a simple table to keep track of the CP and build the sequence of commands, as shown below. I used a spreadsheet which you can download from the support website. The spreadsheet presents the points in much the same way as the table. Interestingly, a small extension to the spreadsheet will allow it to generate the commands for you, so it can be made to act as a simple G code generator. This is a topic to which we will return. It's the beginning of a road that leads to CAM and CAD/CAM programs.

So; I suggest you download and use the spreadsheet, or create a table of your own which looks like table 3 (shown on the next page).

Reading table 3:

the sequence begins with the CP clear of the WO, at X-3.1 Y-9.1 and travels in a semi-circle, clockwise, about a centre at X-3.1, Y-6.1 to X-3.1, Y-3.1

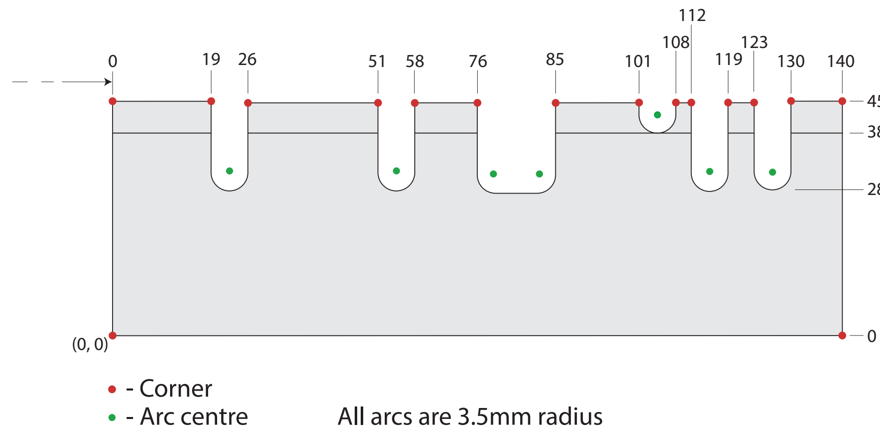


Fig 78: Points which define the cut-outs.

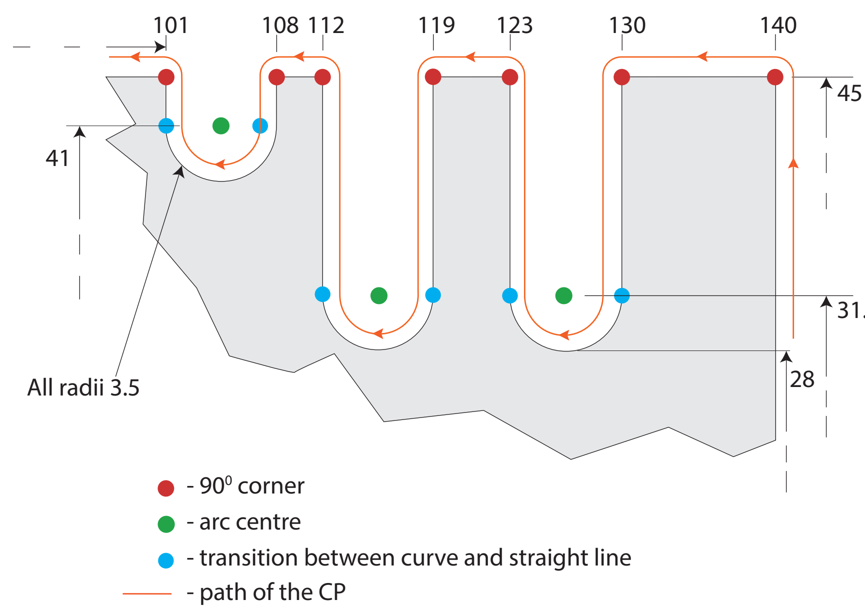


Fig 79: Defining points on typical cut-outs.

Assuming the CP has already been placed at that first point by a previous command, the second point is a centre point and requires no commands. Getting to that third point requires a G3 command type. Using the End point X and Y co-ordinates, then taking I=Xcentre – Xcurrent, and J=Ycentre – Ycurrent, completes the command as G3 X143.1 Y0 I0 J3.1

To get to point 4, which is the start of another arc, requires a G1 move in a straight line. Point 5 is simply a centre, so it has no associated moves. To get to the end of that arc requires another G3 command with X, Y, I and J found as with the previous G3 command. So the commands shown opposite many of the points are the commands required to get to that point from the last position.

Progressing down the table in that way, then extending the table to include the other points around the periphery, allows the command sequence to be worked out in a logical way. Just watch out for the broad 15mm cut out, which will require a straight line G1 cut between the end of one arc and

Point reference	Point type	X	Y	Command type	Command
1	Start of arc	-3.1	-9.1		
2	Centre of arc	-3.1	-6.1		
3	End of arc	-3.1	-3.1	G2	G2 X-3.1 Y-3.1 I0 J3
4	Start of arc	140	-3.1	G1	G1 X140 Y-3.1
5	Centre of arc	140	0		
6	End of arc	143.1	0	G3	G3 X143.1 Y0 I0 J3.1
7	Start of arc	143.1	45	G1	G1 X143.1 Y45
8	Centre of arc	140	45		
9	End of arc	140	48.1	G3	G3 X140 Y48.1 I-3.1 J0
10	Start of arc	130	48.1	G1	G1 X130 Y48.1
11	Centre of arc	130	45		
12	End of arc	126.9	45	G3	G3 X126.9 Y45 I0 -3.1
13	Start of arc	126.9	31.5	G1	G1 X126.9 Y31.5
14	Centre of arc	126.5	31.5		
15	End of arc	126.1	31.5	G2	G2 X126.1 Y31.5 I-0.4 J0
16	Start of arc	126.1	45	G1	G1 X126.1 Y45
17	Centre of arc	123	45		
18	End of arc	123	48.1	G3	G3 X123 Y48.1 I-3.1 J0
19	Start of arc	119	48.1	G1	G1 X119 Y48.1
20	Centre of arc	119	45		
21	End of arc	115.9	45	G3	G3 X115.9 Y45 I0 J-3.1
22	Start of arc	115.9	31.5	G1	G1 X115.9 Y31.5
23	Centre of arc	115.5	31.5		
24	End of arc	115.1	31.5	G2	G2 X115.1 Y31.5 I-0.4 J0
25	Start of arc	115.1	45	G1	G1 X115.1 Y45
26	Centre of arc	112	45		
27	…and so on				

Table 3: An organised list to help keep track of the CP and the commands required for the machining sequence.

the start of the next.

Put the commands in subroutine O140, along with some initial commands to set up the depth of cut using #41, and some others to ensure the CP finishes at the point at which it entered the subroutine.

With this number of points in a sequence, composing the commands in your head is a considerable strain, requiring the use of a couch and a darkened room, or at the very least an Aspirin.

Taking a finishing climb milling cut requires the calculation of a clockwise path around the periphery. Use the same tabular process.

Whereas with the roughing pass G3 commands predominated, this time G2 commands are more frequent. Put the finishing cut commands into subroutine O150 with appropriate depth control and final positioning.

Remember to save your work as you go along.

In the main program, cutting the periphery involves making some initial positioning moves, calling up the subroutines in the appropriate order, then making some final positioning moves.

```
(Rough out the profile)
G0 X-3.1 Y-9.1
G0 Z0
#41=0
M98 P140 L10 (10 repetitions to cut to full depth)
()
()
(Cut the finished profile)
G1 X-3 Y-9
#41=-10 (Set Z to full depth)
M98 P150 L1 (One pass at full depth. L1 could be omitted.)
G0 Z20 (Safe Z)
G0 X0 Y0
()
M5
M30
```

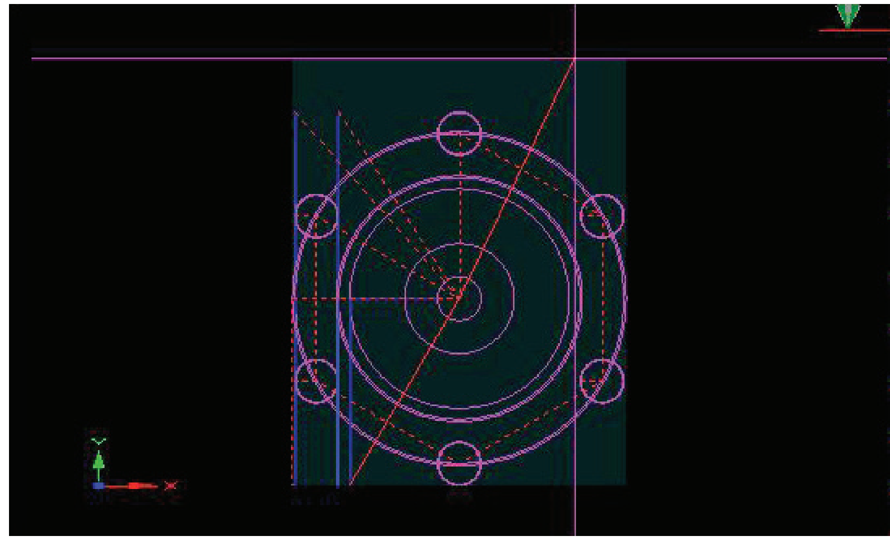


Fig 80: The Toolpath window showing a plan view of the path of the CP.

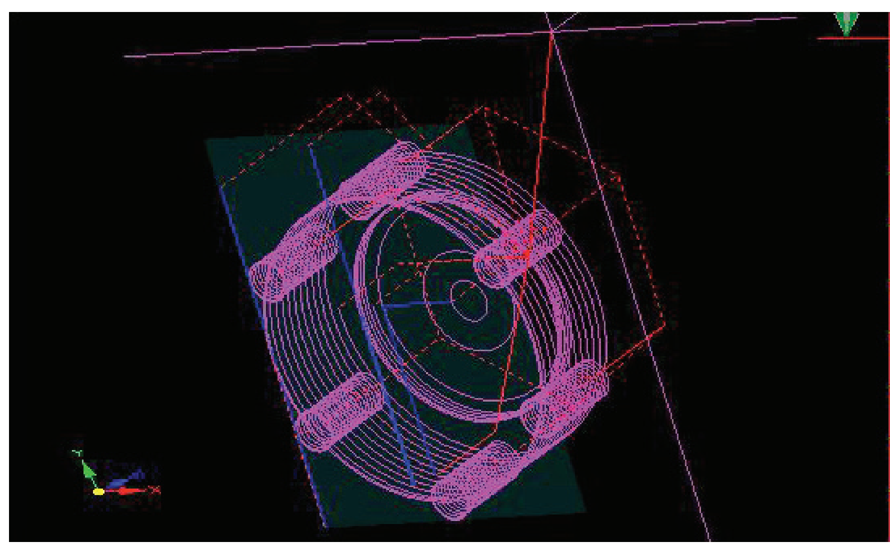


Fig 81: The Toolpath window showing a tilted view of the path of the CP which makes Z moves apparent.

A completed program can be downloaded from the support website, and you should be able to identify the various sections within the main program as well as the subroutines.

Load it into Mach3, then go to the Toolpath window, refresh the Toolpath (necessary whenever you go into this section) and examine the toolpath carefully so that you can identify the paths the cutter follows.

Because there are several sections, and many paths, the display can become confusing. It is also the case that the roughing and finishing paths are very close to one another, and may be indistinguishable. To check the paths logically, use some simple debugging techniques.

Debugging

Debugging involves some simple techniques to fault find, or to split the program into sections so that part of a toolpath can be isolated and examined on its own. It's a science, but we can use some simple methods without becoming immersed in the theory of the whole thing.

The Message window gives you immediate feedback, identifying an error and the faulty line of code, to allow immediate editing. But not all errors trigger a message. Code may be interpreted without error, but the path may be in the wrong place, physically.

Visually, the best guide is the information in the Toolpath window.

Does the basic layout look right, in plan view (fig 80), and are all the features in the right

places in relation to one another?
Check you understand the G0 moves (usually shown in red). Do they begin and end in the expected places? This is often best checked by tilting the view (fig 81).
Check the moves or cuts. Do they head in the right direction? Are the depths of the cuts correct? What happens at the end of a sequence? Where does the CP retract to Safe Z?
Do sequences of parallel moves or cuts actually run in parallel or do they unexpectedly diverge? Cutting down a face using parallel cuts which have different Z depths should show as parallel cuts with uniform spacing.

Do the G2 and G3 arcs face the right way, or are the centres or radii wrong, making them turn to wrong way and appear to balloon?

Check the extents (shown above the Toolpath window on the Toolpath screen). Do cuts extend further than you intended?

To make a program run only part-way through, perhaps to see what is happening as it begins, insert an M30 statement within the main program.
You could, for example, insert M30 immediately after the section trimming the nose, making that read:

```
#41=0 (Parameter 41 holds the Z depth of cut)
F100
G0 X150 Y48.1
M98 P120 L10
G0 Z-10
G1 X-10
G0 Y48 (Y position for finished size)
G1 X150
G0 Z20
M30
```

Leave the rest of the program alone.
In the Toolpath window, refreshing the Toolpath and should show just the initial moves and the path which trims the nose face.

If that's as expected, you can carry on to the next stage. If not; fix it now.
You might need to move the M30 up a few lines, to isolate just the first section, stopping before the final trimming cut:

```
#41=0 (Parameter 41 holds the Z depth of cut)
F100
G0 X150 Y48.1
M98 P120 L10
G0 Z-10
M30
```

If the subroutine is not behaving, try calling it just once, by removing the L10:

```
#41=0 (Parameter 41 holds the Z depth of cut)
F100
G0 X150 Y48.1
M98 P120
```

This approach lets you isolate parts of the program so that you can see what's causing an errant path.

To check what's happening within a subroutine, try putting an extra M99 part way through the code:

```
O120 (Trim the front edge 0.1mm oversize)
(Entry assumes CP at X150 Y48.1)
(Exits with CP at X150 Y48.1)
#41=[#41 - 1] (Z depth decremented)
G0 Z#41 (Down to depth of cut)
G1 X-10 (Cut along until clear of left edge)
G0 Z1 (Lift cutter clear of work)
G0 X150 (Back to the start, ready for a further cut)
M99
```

To suppress a section of program, perhaps to allow you to see what a later part is doing, you can comment out earlier sections by putting brackets around the statements you do not want executed. Take care that this does not alter the starting position of the paths which follow; and you may need to insert an extra statement or two to set up the initial positions. Once you have the following section running correctly, remove the brackets and any additional statements you added. Be methodical about this, so that you do not introduce errors which were not there originally.
It is a good idea to work on a version of the program which you have saved just for that purpose. Keep track of the versions, too.

You can insert messages of your own which will appear in the Message window.
Normally, rounded brackets indicate comments, which are ignored as the program runs, but if the comment begins with MSG, it is interpreted as a command to print subsequent text of that comment in the Message window.
The message command takes the form:
(MSG, Text of the message)
So (MSG, Completed subroutine 270) will print "Completed subroutine 270" in the message window. Note that the comma after MSG is essential. Subsequent messages will overwrite the contents of the message window, though, so you need to think about how and when to stop or pause the program in order to see your message (or not, if the program didn't run as expected).

Insert pauses or temporary stops using the G4 "dwell" command or the M0 or M1 commands. The G4 command takes the form G4 P~ where ~ should be replaced by a number of seconds or milliseconds. The units to be used are seconds, unless the checkbox for "G4 Dwell in ms" is ticked on the Config > General Config menu screen (right hand column).
G4 P10 will make the program pause for 10 seconds, or 10 milliseconds.
G4 lets you run the program in stages, pausing between each stage to let your thinking catch up with the machining operations as they take place. Very useful, sometimes.
The M0 command stops the program, tem-

porarily. Subsequently clicking the Cycle Start button will make the program resume at the next line. This gives more flexibility than the G4 dwell command, because the length of the stop time is under your direct control.

The M1 optional stop command is useful because its effect can be turned on or off in the Config > General Config menu. If the "Stop on M1" command checkbox is ticked (in the M01 Control section in the left hand column of the menu screen) the program will stop, temporarily, when it encounters an M1 command. If the box is left unticked, M1 commands will be ignored. This option can be toggled On and OFF from the Program Run screen, which is very useful, so that you can include M1 commands within your program at strategic points, then enable them all or disable them all as you please.

When the option is enabled, M1 commands are treated as M0 commands, and clicking Cycle Start will make the program continue at the next line.

Both M0 and M1 work best, in my opinion, if they are placed on their own program lines.

MACHINING THE SLOPING NOSE

The nose is angled at 45o to the vertical and matches the angle on a typical countersink, so the easiest way to machine the slope is to take a couple of cuts with a countersink. The challenge is in knowing where to position the cutter.

You really need to know the dimensions of your countersink quite accurately, and to do that you will need to make some trial cuts.

First, check that your countersink has a 90º included angle, as some countersinks deliberately have an 82º angle. If the angle is stated as 90º, assume that it is, unless the countersink has been resharpened.

Depending on how the flutes are arranged, you may be able to measure the diameter of the plain section above the angled nose (fig 82). Or it may be stated on the cutter (as mine is). The missing dimensions are the height of the angled portion and the diameter of the tip on the bottom end (fig 82).

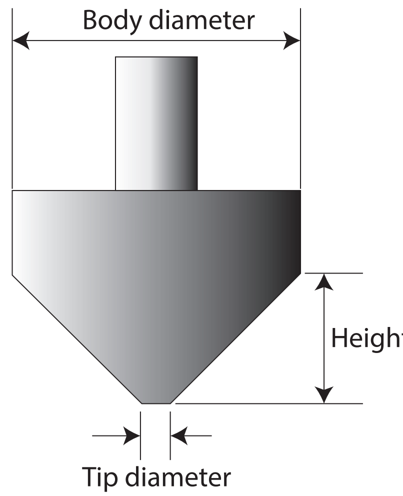


Fig 82: A typical countersink, showing the body tip diameters.

The height can be measured by taking a trial cut. Move to the side, away from the fixture, then use a twist drill to produce a sizeable hole in a scrap block of material. Make the diameter just a couple of millimetres smaller than the full diameter of the plain section above the countersink. Mount the countersink, set Z0 at the top face of the work, then

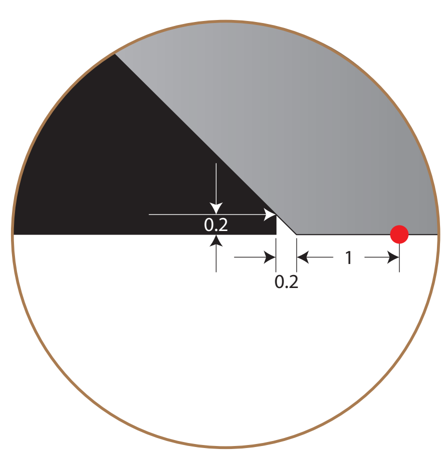


Fig 83: Take some trial cuts until the countersink just produces a plain section at the top of the hole (not shown in this diagram). Measure the depth of the plain section, and subtract that from the total depth of cut, to find the vertical height of the cutting edge.

take a series of trial cuts down into the hole, until the countersink just produces a plain section at the top of the cut (fig 83). The height of the tapered section of the countersink will be halfway between the current and the previous depth of cut. An alternative approach is to cut deeper, to produce a pronounced plain section at the top of the hole, then measure the depth of that, perhaps using the protruding tail of a digital calliper. The height of the tapered section of the countersink will be the depth of cut minus the depth of the plain section.

Once you have the height of the tapered section, you can calculate the width of the flat tip on the end of the countersink:

The height and the width of the cutting edge are equal, when the cutting edge is angled at 45º, so:

tip diameter = diameter of the full body – 2 x height of the tapered section

For example:

diameter of full body = 13.4mm
height of tapered section = 5.7
tip diameter = 13.4 – 2 x 5.7 = 13.4 – 11.4 = 2mm

The nose geometry and the placement of the cutter is shown in fig 83, and the CP should be placed 1.2mm away from the face of the nose, at Y46.2. Because the cutter will take a relatively broad cut, take it clear of the face, at 48.2 and take a series of smaller cuts along the face, say 9 cuts at 0.2mm, followed by a final climb cut of 0.2mm. The program will have a very simple structure, and might use just one subroutine. It will look a lot like the program to face the nose at the start of this section on machining the clamp plate, except that it is the Y cut which is varying and not Z. Start with Y at 48.2, and subtract 0.2 each time you call a subroutine to cut the face. There is a completed program on the support website.

EMBELLISHING THE REINFORCING BAR

The reinforcing bar is mechanically fine, but lacks a certain aesthetic appeal. Before removing the fixture, put the bar back in place. Then set the WO to the centre of the bar. If your WO is currently as set for the front left edge of the finished clamp bar, move to X75 Y12.5 and set X0 Y0 at that point. We can now add a logo to the bar.
There are two approaches to this. The logo creates relief lettering, meaning that the tops of the letters are at the same level as the top surface of the bar, but the surrounding section of the bar is machined away, so that the letters effectively stand proud of their surroundings (photo 98).

Photo 98: A professional touch for the folder.

Method 1 uses an engraving cutter to remove all of the surrounding material.
Method 2 recognises that there are some largish sections to be removed and uses a 2mm diameter end mill to remove the larger areas first, then uses an engraving cutter to remove the smaller areas to finish the detail which is too small for the 2mm end mill. Method 2 uses two programs; one for each cutter.
Both methods use an engraving cutter with 15º half-angle (30º total included angle) and a 0.2mm tip. This is a common and readily available shape of cutter.

The support website contains a single program to use for method 1, and two separate programs to use for method 2. Method 2 is quicker, but requires a tool change between programs. You choose. I used method 2, but the logo is small, so either method will do just fine. If you use a standard end mill, keep it lubricated.
If you are able to use a 2mm single flute cutter designed to machine aluminium you won't need to bother.

Once the logo has been machined, remove the bar. Choose an attractive colour and either paint the surround or fill it with cold enamel (which, despite its name, requires the work to be heated to 150ºC, which perhaps raises a trades description issue?). I used cold enamel because I had it to hand; and I put the work in the oven along with a small steak pie. Happiness all around.

MACHINING THE KNOBS

The knobs require a different kind of work holding arrangement. They could be machined at the end of a square or rectangular section bar, held vertically in the vice, but I chose to use 1inch round bar, held vertically in a chuck.
A bed mounted chuck is a very handy accessory, but there are some safety issues you should bear in mind. I began with a chuck adapter (photo 99) designed to be bolted to a piece of material. The adapter has the same thread as my lathe chucks, because it is handy to be able to transfer from lathe to mill and back again. However; despite the apparent benefits, I should tell you this will not work satisfactorily. First, the adapter as supplied did not have a flat back, so that required attention. Second, the adapter had only two small holes for securing bolts, and they are on peculiar pitch centres. That's not a major problem, but the arrangement is not sufficiently secure, as the two bolts form a single axis across which rocking can take place. Again, that's something which can be altered by adding two additional bolt holes. But the final straw is that the chuck will not screw onto the adapter sufficiently far to bottom against the end of the thread or against a flat flange. The result is that it is virtually impossible to guarantee that the chuck will not unscrew as cutting forces attempt to turn it and slacken it. The practical result is that work is regularly spoiled

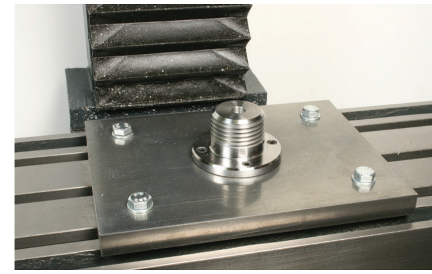


Photo 99: A plate-mounted chuck spigot.

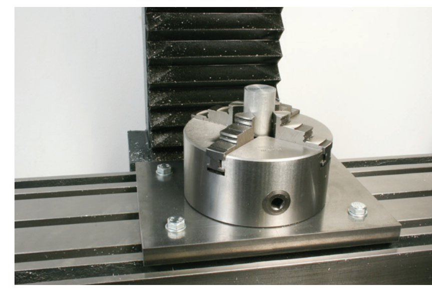


Photo 100: A chuck mounted on the spigot.

and cutters endangered as the chuck moves violently during a typical cut.
My advice is not to use an adapter like this with lathe chucks which simply screw onto the lathe nose. Instead, remove the backplate from a chuck and mount it directly onto a sturdy plate, inserting countersunk screws through the plate from the underside. You now have a really useful accessory which you will use time and time again. If you are feeling flush, dedicate a chuck to this task, leaving it on the plate.
Photo 100 shows a 4 jaw self-centring chuck mounted on a 16mm plate which has holes arranged to match the mill table tee slot centres, just like a vice plate. Do remember to orient the chuck so that you can reach the key hole and turn the key.

Fig 84 (*see next page*) shows the dimensions of the knobs which are machined upside down by creating a circular outer diameter; a smaller diameter short boss; and 6 holes whose centres are outside the periphery of the knob.

Begin by chucking a piece of bar sufficiently long for two knobs, two parting off allow-

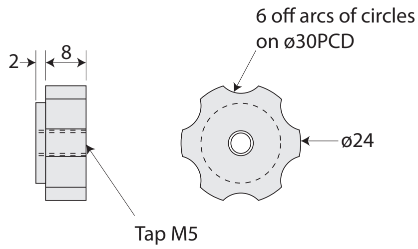


Fig 84: Dimensions of the knobs.

ances and an extra length for gripping while the second knob is being cut.
Drill a 4.2mm hole down the centre, deep enough for both knobs and a parting off allowance (roughly 3mm). Tap it M5 now, or wait until later. Use hand tapping, perhaps assisted by a manual chuck-held hand tapping guide.

The outer diameter is created using a series of circular cuts, descending as each cut progresses, so these are spirals, of the kind you have used before (MEW 212).
The boss uses similar cuts, but of a smaller diameter,
Each scallop is formed from a series of similar circular cuts which, because their centres are outside the periphery of the knob, only cut where they overlap the main body of the knob.

Begin by trimming the top of the bar level and flat. The bar should have been cut squarely within 0.5mm. Set the Z height by lowering the cutter to almost touch the end surface, and set Z0.5 at that point.
Surfacing will take place at Z0, to avoid the complication of having to reset the Z height within the program after surfacing. It is possible to do that, but we need to use some additional techniques which would complicate things at this stage, A simple approach works best, for now.
Set the X0 Y0 at the centre of the bar.
Surface by using parallel passes, or circular passes as shown below:

```
(Trim top edge)
M98 P160
```

and the subroutine

```
O160 (Trim top edge)
(Assumes CP is at X0 Y0 Z10)
(Exits with CP at X0 Y0 Z10)
G0 X-10 Y-17
G0 Z0
G1 X-10 Y0
G2 X-10 Y0 I10 J0
G1 X-5 Y0
G2 X-5 Y0 I5 J0
G1 X-2 Y0
G2 X-2 Y0 I2 J0
G0 Z20
G0 X0 Y0
M99
```

The periphery of the knob can be trimmed by roughing oversize then climb milling to finished size, using:

```
(Trim periphery of knob)
#41=0 (Initialise depth of cut)
G0 X-15.2 Y0 (Initial positioning move clear of the surface)
G0 Z0.1 (Positioning just above the top surface)
M98 P170 L11 (Spiral cuts down below depth of knob)
G3 X-15.2 Y0 Z#41 I15.2 J0 (Level cut around periphery)
G0 Y-17 (Move clear)
G0 X-15 (Go to new radius)
G1 X-15 Y0 (Start cut)
G2 X-15 Y0 I15 J0 (Climb mill periphery)
G1 X-15 Y17 (Move clear)
G0 Z20 (Safe Z)
G0 X0 Y0
```

and the subroutine

```
O170 (Trim periphery of knob: one pass anti-clockwise)
#41=[#41 - 1] (Make depth of cut 1 lower)
G3 X-15.2 Y0 I15.2 J0 Z#41 (Spiral down to depth)
M99
```

The boss is cut using the same technique:

```
(Trim periphery of boss)
#41=0 (Initialise depth of cut)
G0 X-11.2 Y0 (Initial positioning move clear of the surface)
G0 Z0.1 (Positioning just above the top surface)
M98 P180 L4 (Spiral cuts down to the depth of the boss)
G3 X-11.2 Y0 I11.2 J0 (Level cut)
G1 X-11.2 Y-17 (Move clear)
G0 X-11 (Go to new radius)
G1 X-11 Y0 (Start cut)
G2 X-11 Y0 I11 J0 (Climb mill periphery)
G1 X-11 Y17 (Move clear)
G0 Z20 (Safe Z)
G0 X0 Y0
```

and the subroutine

```
O180 (Trim periphery of boss: one pass anti-clockwise)
#41=[#41 - 0.5] (Smaller cut because end cutting as well as side cutting)
G3 X-11.2 Y0 I11.2 J0 Z#41
M99
```

The scallops are more of a challenge, because we need to find the centres before they can be machined. The knobs were designed in a CAD program, so I could have shown the co-ordinates of the centres on the drawing, but it is worth knowing how to find them when they are not given. That also allows you to experiment with the size and positions of the scallops, to improve the look and feel of the knobs.
There are at least three techniques we can use, but one simple method is to use machinists' tables which show how to work out the X and Y co-ordinates of sets of circles where the pitch circle diameter is known, Fig 85 shows values used to calculate the co-ordinates of the centres of each circle

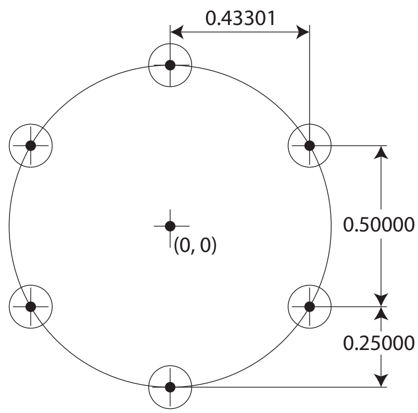


Fig 85: Values used to calculate the co-ordinates of each of the circles in a regular circular pattern of six.

in a regular pattern of 6 circles of diameter 1unit, and table 4 shows how this relates to each hole.

Circle no.	X	Y
1	0	0.5
2	0.43301	0.25
3	0.43301	-0.25
4	0	-1
5	-0.43301	-0.25
6	-0.43301	0.25

Table 4: Multiply the values in the table by the PCD to find the X and Y co-ordinates of the centres of the 6 equally spaced circles. The centre of the PCD is assumed to be at X0, Y0.

Multiply each factor by the PCD of the pattern (30mm in this case) to find X and Y. Instead of doing the calculation manually, let Mach3 do it for us. For convenience, put the PCD into a parameter (#43 in the following program segment) and multiply each of the factors from table 4 by the contents of #43 to find the X and Y co-ordinates of the centres of the holes. Then use those co-ordinates to position the CP ready to machine each hole in turn. This approach allows the PCD to be altered easily, and Mach3 will then automatically adjust the centres of the 6 circles.

The machining sequence is similar for each hole, so once the CP is in position for a hole, use G92 to temporarily make that position X0, Y0 so that we can use the same subroutine for each hole. Remove the temporary offset using G92.1 at the end of the subroutine, to restore to original co-ordinates before moving to the next hole position.

```
(Create scallops)
F100
G0 X0 Y[0.500*#43]
M98 P190
G0 X[0.43301*#43] Y[0.250*#43]
M98 P190
G0 X[0.43301*#43] Y[-0.250*#43]
M98 P190
G0 X0 Y[-1*#43/2]
M98 P190
G0 X[-0.43301*#43] Y[-0.250*#43]
M98 P190
G0 X[-0.43301*#43] Y[0.250*#43]
M98 P190
F100
G0 Z20
G0 X0 Y0
```

and the two subroutines (one of which calls the other)

```
O190 (Cut one scallop)
(On entry assumes CP is at centre of scallop, and Z#41)
(On exit X and Y are the same as at entry, and Z20)
(Uses G92 to temporarily shift the origin)
(to make the centre of the scallop 0,0)
(Offsets are removed before exit back to the main program)
G92 X0 Y0 (Apply an offset to make the current point 0,0)
#41=-2 (First 2mm removed when cutting boss)
G0 Z[#41 + 0.01]
G2 X-1.9 Y0 I-0.95 J0 (Entry curve within circle)
M98 P192 L18
G2 X-1.9 Y0 I1.9 J0 (Flat bottom)
F50
G3 X-2 Y0 I-0.05 J0 (Small semi-circular entry curve out to periphery of finish cut)
G3 X-2 Y0 I2 J0
G0 X0 Y0
G92.1 (Remove offset, restoring original co-ordinates)
F100
G0 Z20
M99
```

```
O192 (Scallop: spiral cut down)
(On entry assumes X-1.9 Y0)
(Exits with X-1.9 Y0)
#41 = [#41 - 0.5]
G2 X-1.9 Y0 I1.9 J0 Z#41
M99
```

Note the entry curve within subroutine O190:

G2 X-1.9 Y0 I-0.95 J0 (Entry curve within circle)

That is an entry curve inside the circular cut out, taking the CP from the centre of the circle to the periphery.

Then, later in O190, there is a tiny transition curve as the CP is put into position for the finishing cut. The cutter is already at X-1.9 Y0 and needs to go to X-2 Y0, so there is a small entry curve there:

G3 X-2 Y0 I-0.05 J0 (Small semi-circular entry curve out to periphery of finish cut)

In both cases, the direction of the entry curve is the same as the movements which follow, so one is a G2 and the other a G3 command. This is all to give a smooth transition as Mach3 interpolates positions by looking ahead. These small refinements to basic coding make all the difference to the finished job.

There is a completed program on the support website which you can run to machine

a knob. Once machined, remove the bar and part the knob from the bar using the lathe (which is a good example of how handy it would be to be able to take the chuck from the mill to the lathe).
Return the bar to the mill and machine another knob.

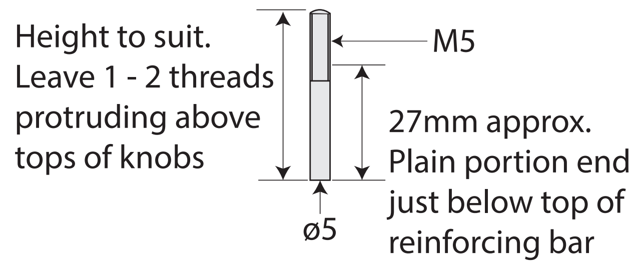


Fig 86: Dimensions of the threaded knobs.

COMPLETING THE FOLDER

Turn two studs with M5 threaded ends (fig 86) to fit the knobs, and trim them to length so that each stud is just long enough to leave a couple of threads protruding above the tops of the knobs.

Polish the plates and the bar until they gleam. Use 600 grit wet and dry, followed by 800, then 1000 and 1200 grit. Use a full sheet of each grade, spread on a flat surface, and move the plates back and forth across each sheet with even pressure. Keep them flat to the sheet, to avoid rounding the faces or edges.
As you change from one grade of sheet to another, carefully clean the work so that no grains of the previous grade remain on the surfaces. Don't touch the edges; they should already have a nice finish from the cutter.

Make the plates sparkle with a final polish using Brasso or Solvol Autosol.
Loctite the plain ends of the studs into the base. When the Loctite is dry, clean any excess then assemble the folder.

Put your sunglasses on, and admire another impressive job.
Pride of place on the front shelf, I think, angled to catch the sunbeams.