SHERLINE
PRODUCTS
INCORPORATED 1974

v1.1a

Changes and additions

v1.0c—Added 10/9/03—Page 66, Supplemental information for loading Sherline version of EMC when using with 8760 driver box. Page 13, Using these instructions for a lathe.

V1.0c—Added 11/13/03—Minor change to line one of coding on page 66. "/" symbol added to end of line.

V1.1a—Correction 8/30/04—A line of code and tool setting instruction has been corrected in the example for the modified circle program on pages 36 and 37.

Operating Instructions for the Sherline Vertical Milling Machine CNC System

P/N 8540

The most up-to-date version of these instructions can always be found on the Sherline web site at www.sherline.com/CNCinstructions.htm.

CAUTION: Do not connect or disconnect stepper motors when the driver box is powered up. Always turn off power to the driver box before plugging in or unplugging a stepper motor. Improperly connecting or disconnecting a motor can damage it.

An introduction to Programming and Operating Your Sherline CNC Vertical Mill

By Joe Martin

A quick history of CNC

I've been personally involved with CNC since the early seventies and soon realized that this was the way to make things. Back then all sorts of ideas were being tried to simplify the mass production of machined parts. Many manufacturers were using hydraulic power and came up with some interesting systems. The method of choice before NC (back then they were called Numerical Control (NC) because the storage device was a one inch wide paper tape) was a tracing system that duplicated parts by tracing them with a hydraulic system controlled by a really neat valve controlled by a stylus that the operator would move like a probe and the machine would duplicate their movements. Gigantic machines were built around this idea for the aircraft industry.

Hydraulic CNC systems?

It was only natural that this was the group of manufacturers would be the first to try this new field, and did they ever come up with some weird systems. You also have to realize that electronics were also pretty crude during that same time period. My first NC machine was a Cinematic manufactured by Cincinnati machine tool. It had ball lead screws driven by hydraulic motors and was fairly reliable and quite popular at the time. Tool changers were still in their infancy. I ran it until it was so out-dated that I gave it away when it still ran. Another interesting point is this machine had a "wire wrapped" control with few circuit boards. If a recent graduate of electronic engineering had ever looked into the control enclosure, I'm sure his first comment would be "impossible."

I thought I bought a telephone company

My first machine that was controlled by electronics turned out to be a disaster. I bought it used and it had a stack of manuals 18 inches high. I never dreamed I was going to look at every page but I did. When I opened the control and looked in there, I thought I bought a telephone company. I believe there were 120 individual circuit boards and thousands of individual transistors. It was manufactured by Edlund. We used to call it a "Deadlund." I wasted more time and money on that machine than I care to admit, and I felt relived as it was loaded on a truck for its final journey to the junkyard.

Problems with no memory storage

The way this machine stored memory was interesting. At this time there weren't any memory storage devices invented for commercial use. What this control did was feed a single block of code (a block of code contained the instructions the machine needed to make a single move) into a very long piece of Nichrome wire. The high resistance of the wire would delay the signal long enough so the signal could be amplified at the other end and sent back on the same wire without interference. The signal would then bounce back and fourth until the machine had moved to a location that corresponded to that block of code.

Sometimes I can be persistently stupid

New machines were thousands of dollars more than I could afford and not that reliable either. I remember a friend who had a three-year-old machine that cost over $100,000 that needed service. The first thing the technician said was, "I didn't think they had any of these old bastards still running." You can imagine how my friend felt when he made that $2500 payment each month.

You think I would have learned by then, but I can be persistently stupid at times. My next disaster was a MOOG milling machine. It used a Bridgeport base and didn't have leadscrews. It was entirely controlled by hydraulics. Movement was controlled by thin plates that moved with the machine slides with accurate holes located at every inch. Pins about 0.187" (5mm) would engage the proper hole and then another plate with holes spaced at 0.200" would come into play, and the last 0.200" relied on a single turn of a lead screw. The machine finally arrived at a position within a 0.001" of accuracy. I didn’t realize it when I bought it, but I was horrified to find that this mother worked like a player piano. The one-inch wide paper tape wasn’t read with switching devices like other machines. It actually was more like a valve that allowed or prevented air to get to cylinders that controlled hydraulic valves. When it read a block of tape (around 10 lines of holes of 8 holes each) the SOB sounded like a steam engine. I chalked that one off to having more balls than brains; however, we actually made more parts with it than we did with the Deadlund.

It should also be noted that I’m sure that these early NC machines were the best that could be designed with what was available, and the solutions that they came up with were quite ingenious at the time. I’m looking back at it from a slightly humorous position and in no way infer that the designers of that era weren’t up to the task. They just didn’t have the tools to work with that we do today.

The course is set

At this time, NC machines could only cut straight paths, and as soon as the electronics were available to store just a small amount of memory the new rage became “look-ahead” control systems. This meant the cutting tools wouldn’t create machining problems when they hesitated as the next block of information was read. Stepper motors were used for a short period to drive the lead screws; however, within a few short years the entire industry switched to DC motors. Encoders or resolvers were used to keep track of position. The DC motors controlled by fast computers and working in unison with accurate ball lead screws created a system that was very close to where we are today; however, they were slower and very expensive. Today, servo drives use AC motors controlled by varying the frequency to the windings, eliminating the brushes needed with DC motors. The latest innovation is linear motors that can move machine slides at incredible speeds.

A new way of thinking

The marvelous part of the CNC revolution wasn’t just the fact that it was eliminating workers from sometimes very strenuous and boring jobs; it was that there was finally a method of cranking these handles in unison to do things that the best machinist in the world couldn’t accomplish. By moving screws on the X- and Y-axes in unison, you could machine tapers, circles and, in fact, any shape you wanted. This allowed engineers to design parts with the shapes that they wanted, not just shapes that were possible to machine using the old methods. Machines that cost thousands of dollars suddenly became scrap iron. These new CNC machines didn’t care if they were cutting a complex shape or a straight line. Whether the tolerance was tight or not the machine was always “right on,” and the tools determined the tolerances.

Ball lead screws—the missing ingredient

The lead screws used on these machines should be mentioned. They are the interface between the computers and the mechanics. The problem of backlash was solved with “ball lead screws.” These screws have re-circulating balls that roll in a groove ground into a shaft at a pitch of two tenths of an inch. The pitch on these screws has increased over the years to achieve speeds over 1000 inches (25 meters) per minute. At a pitch of .200″ (5mm), a lead screw would have to turn at 5000 RPM to accomplish this speed, which is why ball lead screws have as high as one inch in pitch. Even more amazing was the fact that they improved the accuracy as they increased the pitch. You can make a .0001″ (.0025mm) correction on a good CNC lathe. Think about that. A slide will accelerate to a speed of over 1000 inches a minute in less than a second, move a short distance and decelerate, stop and still be accurate to one ten thousandth of an inch. The ball screws must be very precise, because a lead screw would be useless if it had any backlash (the amount you have to rotate a lead screw before the slide moves).

Ball lead screws are very difficult to make, which makes them quite expensive—several thousand dollars for each axis. The people who solved the lead screw problem should be commended as much as the electronic geniuses who came up with the computer controls. At Sherline, we have CNC machines that have been running over ten years and still don't have any noticeable backlash.

Carbide insert tooling changed the entire machine tool industry

At the same time, carbide insert tooling became available and took the market over like a storm. I don't think I could have ever convinced a machinist in the fifties that some day he would be taking 0.300" cuts on cold rolled at a cutting speed around 400 to 600 fpm (200 meters/min) using a 40 hp lathe at 0.020" (.5mm) feed rate for each revolution with little carbide tools made using powdered metal technology and held in place with little screws and get marvelous finishes on gummy old cold rolled at the same time. They do, and we at Sherline do all these things that give our customers a lot of bang for the buck. The consumers of products that manufacture using this technology benefit as much as the manufactures that use them.

Because lathes could also produce these types of moves, the large and expensive form tools were on there way out. This may not seem significant, but by generating shapes rather than forming shapes, shapes could be far more precise and at the same time machines didn't have to be so massive to prevent tools from chattering. Hand scraped ways were replaced by frictionless ingenious slides that lasted for years with little maintenance. Because there is always the possibility of a crash, machines are no longer built where the headstock is an integral part of the base casting, and in most cases they can be realigned if one of these disastrous events should happen.

Why I love CNC robots

In closing this section, I firmly believe that these machines that we call CNC are the robots of the future and I truly love them. They have allowed me, Joe Martin the designer, to design the parts that I've always wanted to design without dumping my problems on Joe Martin the machinist, who must produce parts for Joe Martin the businessman, who can supply you, the customer, a fairly priced quality product and at the same time allow Joe Martin the owner a reasonable profit to buy more of these marvelous CNC machines, of course. And so it goes.

Even though this doesn't do the subject justice I thought you might find that interesting. It's always wise to see at least a brief history of a subject before becoming part of it, even if it's only statistical.

Here is something I wrote in my book that's available online that you may find interesting. See http://sherline.com/business.htm.

> "THE NEW MANAGERS
>
> CNC machines allow a smart worker to produce more work than ever. Rather than eliminating the need for intelligent workers, these machines require organizational skills that few managers have. ***The machines they control are very complex, and the employees who can control these machines are equally as valuable to a company as that of standard managers.*** In the past managers were considered people managers. The managers who control these machines are robot managers. When you consider how important these machines have become to the quality and efficiency of manufacturing a product, the group of workers that controls these machines will soon be equal to the managing staff of any company in prestige and wages. Today's management still hasn't grasped the concept of this point, but soon they will be forced to accept this condition when they find their million dollar machines operating at half speed."

This CNC class is open to those willing to work hard

In order to make the following instructions more interesting for me to write and you to read I'm approaching the teaching of CNC programming the same as I would as if I were teaching a small class of students. The students who attend these specialized classes later in life have various reasons and expectations for being there, and I'm trying to address the whole class as I write. When I take off on a tangent I'm addressing individual students with a different background who I visualize sitting in the class and answering a question that I believe may be asked in a similar class, so please bear with me. I don't want any of my students left behind.

Enjoy the flight

In many of the coming examples I also describe the movement of the spindle as an aircraft that you're riding in. This may seem "far out there," but CNC milling is a 3D world, and if you try to describe these movements in the movement of the table slides you'll soon not know whether you are coming or going. You have to visualize the work staying still while the cutter moves. Again, bear with me and enjoy the flight.

An Introduction to the Sherline CNC System

A different way of learning

Long ago I became aware of the success rate of people who, when left alone with their video recorder and an instruction manual, failed to pass the test and had to stay up late to turn on their recorder to record a show they wanted to save. From their own statistics you would think manufacturers would have attempted to teach in a different manner, but they don’t. The products get more complex, making the instructions more complex, and the poor consumers of these products are at a loss. I don’t teach using these methods because they obviously don’t work. With my instruction you won’t be sure if you’re reading a novel or an instruction manual, but I’m sure you’ll be able to program and make a part when you get to the end.

The first instructions I completely wrote for my tools (I had written instructions about radio control equipment and rules for RC events in the past) were how to cut a screw thread using a rather crude device I designed (but which works well) for a Sherline lathe. I was surprised how few calls on how to use it came in after I sold my first 200, because it is a bit of a complicated thing to do. I then started to wonder if anyone was using it, and you’ll never know how relieved I was when a customer came in with a bunch of parts that he made with threads on them.

I knew I was on to something and have always written in this style ever since because it works. It had to work because at that time I was the only one at my company could answer questions of this type, and I didn’t have time. The fact that I’m a self taught person probably has something to do with it. I even went on to write a couple of books in this style that were well received.

A new challenge

I remember getting a call from Sears about my folksy style of writing that didn’t follow Sears’ instruction guidelines about the screw cutting attachment. I asked him how many calls they had received about how to cut threads using it. He answered, “none.” I asked him who there could answer a technical question of this type. He answered, “nobody.” I said, “With this being the case, don’t you think you should leave well enough alone?” I never heard from him again. Right now, I’m looking at these instructions as one of the biggest challenges I’ve ever taken on as an author, and I don’t plan to fail. The average instruction manual written on this subject is two to three hundred pages of very technical reading. You’ll never crack a smile reading one of these manuals. I’ll do it in less than fifty.

The only students I hired who learned CNC had good teachers, not good training manuals. In fact, I can’t think of any employee in the last 30 years that I’ve ever hired who learned CNC programming and how to run these complex machines on his own. You will! I had to learn on my own because I was so broke after I bought my first NC I was out of choices. I learned by staying up late, working my ass off and making costly mistakes. I would have given my left you-know-what if I had something like this to read rather than reading the junk written by professors trying to prove how smart they were to other professors rather than attempting to teach their own students.

Sherline customers want to build components

I saw the customer who was going to read these instructions as a person who wasn't particularly interested in computers or gadgets and was only interested in making parts. I believed they were curious about CNC and wanted to learn about it without spending too much money. They probably haven't tried to learn how to do something this complicated in the last 20 years. Remember that CNC is a new way of doing and thinking about the machining process, and I believe I'll help get your brain in tune with what you are about to learn with my ramblings.

Learning while you work

A student of this class could be already working at the trade. Words of advice for this group: You can't afford to have idle thoughts as you load parts on machines that other people set up and wrote the programs for unless you are satisfied to load parts for the rest of your life.

First study the mechanical part of the setup; things like how the part is held and the type of cutting tools used and how long they last. You have to show some initiative and start analyzing the program you are running and ask intelligent questions. Notice, I said *intelligent* questions. Bosses are usually smarter than you think they are, and they easily know when some when clown is pulling their chain. Don't ever make suggestions unless you are sure you are right. A better way to make a suggestion of this type is to ask them just why the job is set up or run the way it is rather than your way. That way you're not threatening them with your question, and they'll appreciate a chance to teach you if you are wrong. If you find out you were wrong, it gives you a way out by saying that you were sure that they must have had a good reason. It wouldn't hurt to let that rule work its way into your personal life.

When you start studying their programs, you'll find them written like a story. Just like a book, the better the author, the easier the program is to read. Each tool used will have its own chapter. (In the real world of machining you'll run few machines without automatic tool changers.) Study these chapters one at a time and watch what the machine does as the program advances. In case you didn't realize it, I'm also instructing all you hobbyist out there on how to organize your programs and setups. Plan them like someone else is going to run the job and pick apart your own work. If you don't, you might find yourself trying to machine your vise off the table.

Getting you into the learning mode

I also felt that people of this type would want to read something about what they are in store for before they started. I wanted my customers to understand just how difficult CNC programming can be to learn at the start and just how interesting it can be at the same time. In other words, I felt "Joe home machinist" doesn't stand a chance unless he's inspired to do so, and in my own way I'm trying right now to inspire you to learn how to program and use these marvelous new CNC machines. I know how easy it is for you to go back to the way you already know or forget the whole idea. I don't want customers who think it'll be easy to learn because they'll be unsatisfied from the beginning and take out their anger on Sherline.

You're about to learn how to control your robot

These aren't instruction about operating a simple single system like a VCR, but instructions about controlling a very complicated device. You are not operators of these marvelous machines; you are masters telling your robot what to do, and you, the controller, have unlimited choices to make. I want to get this concept through to that average customer I mentioned, yet I know that only one out of five professional machinists I know seems to really understand this point.

I’m going to be a student in my own class

I was shocked at just how much I had forgotten about the process of CNC programming. I probably know as much as anybody of what you can do with CNC machines, because I own at least twenty modern machining centers, yet I haven’t really written any code in fifteen years. After thinking about it for a couple of days while unsuccessfully attempting to write a simple program I decided this is a good thing. I also decided that I’d learn this from the Linux site on my own and not use the books that were available to me. I believed that if I wrote the instructions as I learned how to program EMC myself I’d write better instructions; after all, all you have to be is one page ahead of your students to teach. Just joking, teachers.

Something for nothing

Sherline Products Inc. is not charging you for this EMC program, and it is free because of work done by NIST (National Institute of Science and Technology), a government funded organization. The EMC (Enhanced Machine Controller) was written by the NIST and is a very sophisticated program. We all now have the benefit of millions of dollars of programming along with the source code, if needed, simply by asking.

We all owe thanks to the EMC group who donated their time

At the time this program was written it was thought that only very advanced programmers and scientists would be using a program such as this, and the average person stood little chance of making it functional. Fortunately for us, a lot of very smart people before us, donating their time, started developing a programming interface that allowed both businesses and hobbyists to take advantage of what they had done without being engineers. I’m very pleased with what they have done. This group still works together through the Internet, and help is available through Internet chat groups. Many people have donated much of their time to make this system “user friendly.”

The EMC program operates under the Linux operating system, which is also a free system, and it would be wise to operate it from a dedicated computer for this task. A dual boot system can lead to problems, because Microsoft code has been purposely written to be the dominant control of your computer whether you want it or not. I like the Linux concept of people working together where all have the benefit of any single person’s work. For example, these instructions will have a Sherline link placed on the EMC for all to read whether they are Sherline customers or not. It will be my donation towards a successful system available to all. I’m learning how to program EMC by using the internet site put together by dedicated EMC users, so you can also fall back to their site for help. http://www.linuxCNC.org/

You will be learning the CNC language of the robot world

The standard “G” code method of programming is to simply replace the manual inputs into a machine (cranking the handles) with the commands that have become industry standards. This is the method you should learn before learning any other method of programming. It is the basic way that CNC machines “talk” to other CNC machines. Even if a simpler system were available today I still wouldn’t teach it. It would be similar to teaching a language that few citizens of the world spoke. Programs written using g-codes can be read throughout the world. G-code standards came about very early in the game because this game was controlled by engineers rather that marketing people.

Contouring programs, another world

Programs that allow you to do contouring quickly become very complex. Today there are some affordable programs available for sophisticated hobbyists. These programs can create as many problems as they solve, because of the complexity of each program. Each program has to work with other programs, and each program has its own set of complex rules. If you think you can learn these programs easier than you can solve a couple of trig problems you're in store for a rude awakening; however, if you already know how to use 3D CAD programs you could quickly find these programs very useful. It should also be stated that it is the only way available to create contoured machined shapes.

What it costs in the industrial world

The basic method that these programs work together in the industrial world and the cost of these programs is: 1) The part is designed using a modeling program ($5000 to $10,000), 2) The modeling program is converted to a CNC program (another $5000 to $6,000) and 3) the resulting code is run on a CNC machine starting at ($25,000) that makes the part. The people who can operate these systems are carefully chosen, trained and well paid. The fact is that any business has no choice except to use the program they purchased, because the programs cost so much and are so very necessary to remain competitive in today's world. They have to spend a great deal of money to accomplish this, while the hobbyist has a choice.

Today's low-cost programs were once complex, expensive programs

The business world understands how complex it is to implement these procedures; the hobbyist doesn't. Because the hobbyist may be purchasing a program that cost thousands of dollars just a few short years ago for a few hundred dollars, the programs themselves haven't become less complicated. They only cost less! In your mind you may be having visions of complex shapes being machined with little effort on your part while your two thousand-dollar investment turns out parts that the old masters of the machine trades would envy. Well my friend it's possible, but it's time to come down to earth and again think about the complexities of doing so.

EMC—still the latest and greatest

I chose the EMC program to work with because: 1) EMC uses coding standards that are used in the modern machine tool world, 2) You will not be dealing with an outdated programming system, 3) EMC can deal with the cutter compensation (cutter comp) that is needed to make complex parts in an efficient manner, 4) The cost saved by not having to pay for this marvelous programming system will allow you to purchase a separate dedicated computer to drive your mill and 5) There's great bunch of smart people working on EMC to constantly improve the system and make it easier to use in the future and remain current.

This is going to be easy if you do it my way

The Linux operating system is similar to use to all others, and I'll try and mention things that are a little different in using the EMC program. To start with, you don't have to double click to enter a program, so a single click on the big red "S" (Sherline, not Superman) with the file name "*inch_freq*" underneath, and after a few seconds you'll get the graphical interface to the EMC. Well, that was easy. Now, after BS-ing you to death on how hard this was to do I'm going to show you how easy this can be, but only if you do it my way.

Computers have the patience of a saint

If you've had little computer experience the first thing you will learn is that computers have the patience of a saint. If computers were more human they would probably like to toss their operators out the window as much as frustrated operators want to do the same to them. The only way you can keep a computer from giving you an error message is to not give it errors. The tricky part is computers don't always tell you the error to fix or how to fix it. Computers just tell you that there is an error of a certain type. This can become maddening, and you have to know when to take a break and walk away. Things of this nature are the worst when, for the sake of your own sanity, they should be the best. In many ways, learning programming is similar to learning machining. You can't take short cuts. If you find yourself going in the wrong direction, back up immediately.

Learn machining first

If you know little about machining and little about computers I'd suggest you first concentrate on learning about machining. How can you expect a machine tool and a computer to carry out your commands if you don't know the commands to give? Above all, don't start with a project so complex you don't stand a chance in hell of succeeding. Let each simple problem solved have its own rewards in knowing you just traveled another block further down the CNC machining super highway, route EMC.

> NOTE: Joe Martin has written a book called ***Tabletop Machining*** that is an introduction to metalworking at the small end of the size scale. Details can be found at www.sherline.com/bookplug.htm. If you are new to machining, you will find the information in this book quite helpful.

Cutter compensation—what makes CNC work in a machining environment

I could have sneaked out the back door and only wrote instructions for a couple of simple moves without cutter compensation and let it go at that; but deep down inside, I'd have known I was cheating you out of the real benefits of CNC. I believed that if I started teaching you to use this marvelous programming tool right at the very start it will soon be second nature, and our class will leap-frog ahead of our competition with those thick boring books being taught by instructors I wouldn't let near the least expensive CNC machine in my shop.

If you take the effort to learn this method of programming at the start along with a little simple trig, you'll find it well worth the effort. Cutter offsets are the way you control size with modern CNC machines. When you start making more complex parts and you don't use cutter comp you'll have to change the basic numbers of your programs to make a part on size. Sounds simple enough until you find that every time you correct a number here you end up screwing up another number there. Take my thirty-five years of experience in dealing with CNC and accept this cutter offset stuff as a fact.

I can’t help anyone who isn’t willing to work hard

Back in ancient times the great mathematician Euclid told a Pharaoh who wanted to learn geometry the easy way that Pharaohs had to walk down the same road as scholars to education. In a sense I’m also stating to you that you will have to learn programming the same way as I’m about to, and at this time I’m only a couple of pages ahead of you. I am certain that if I sat by the phone and answered questions twenty-four hours a day, and I don’t plan to, I couldn’t help those who aren’t willing to put forth the effort it takes on their own to learn to work with these marvelous tools.

Route EMC

This road that you are beginning to travel can become one of the most interesting roads you have every traveled; however, you must take the time to gather the facts that can make a trip so interesting. Instead of seeing great ancient cites on this road we’ll be gaining the knowledge that will make our next stop even more interesting and more complex. We’ll study a little and then apply these rules to your machine. We will first make simple moves that become more and more complex and at the same time more interesting as we travel on. Eventually we will reach Martinsville, a city on this road, where you will have so many choices as to the direction to travel that you’ll be leaving me and heading out on your own to design and build things that you never thought possible in the past. The rules and facts that you will learn on this road will allow you to have a machine at your control that home machinists a few short years ago couldn’t dream of. Take the time to enjoy this trip because you’re on a road with no end if you have set your goals correctly.

You may want to consider a CAD program in the future

I’m going to close this section off by inserting something I wrote about learning and using AutoCad®. The ease with which many of the programming problems can be solved has to be mentioned at this time; however, I wouldn’t recommend dropping what you are doing. From what I understand there are several excellent programs available at very reasonable cost, but I don’t want to recommend any particular program.

The following paragraph is from my book “Tabletop Machining.” It covers some of my comments about learning to use a CAD program.

> **“Learning AutoCAD®**
>
> The first task I took on to learn the program was to lay out and design a set of bevel gears that could be cut on a Sherline mill using a rotary table mounted on an angle plate. By drawing the cross sectional view accurately, the angles needed to cut this gear could be taken directly off the drawing without using a trig table or a calculator. I was beginning to see the light.
>
> I’m getting pretty good at using the program now and it has put a lot of the fun back in designing for me. Looking over my partner Carl’s shoulder at an assembly designed by computer was of no help for me, and I would need a standard full size or larger layout to design with. I’ve since found that doing the design totally myself, it has become the perfect way for me to work. I love the program. It has become the perfect program for a designer and manufacturer like myself and has made me more productive than ever.

I'm fascinated with the program because it does a drawing without errors when used properly. The program eliminates much of the boredom of adding and subtracting numbers as you go. Accuracy isn't attained by the precision of your lines but rather by the accuracy of the information the program is given. Angles are not derived from a protractor and using divider points but from calculations by a computer to as many decimal places as needed.

I learned drafting in high school back in the fifty's. My problem was I was a "slob" when it came to drafting a pretty drawing. My numbers and views would be correct, but my lettering and lines weren't neat enough. Things have changed now with the aid of this program and my lettering is just as good as the best. I can no longer spill gobs of ink on an almost completed drawing and have to start over. I can change my mind as much as I want without irritating anyone. I'm a happy man."

Operating Instructions

Part 1

CNC System Warranty

If within one year from the date of purchase of a new Sherline CNC system any tool or component fails due to a defect in material or workmanship, Sherline will repair or replace it free of charge. In addition, it has always been our policy to replace at no cost all parts, regardless of age, which are determined to have been incorrectly manufactured or assembled and have failed due to this cause rather than because of improper use or excessive wear caused by continuous use in a production environment. In cases such as this, Sherline will inspect the machine or part and will be the sole judge of the merit of the claim.

Freight charges for returning a machine are not covered. Save the original packaging material to use if the machine or computer must be returned to the factory for service. Sherline will not participate in insurance claims where the sender didn't make an honest effort to package the items to prevent damage caused by normal shipping stresses or repair said damage without charging the sender.

Merchandise that has been abused, misused, improperly maintained or insufficiently lubricated is not subject to warranty protection. Proof of date of purchase and dealer name may be required for service under this warranty.

Specific exclusion for wear due to production use

Although many Sherline tools have found there way into production environments, they have been designed for intermittent use in home shops. It would be unreasonable to expect a Sherline machine to stand up to the rigors of continuous production use. Continuous use of stepper motor driven leadscrews in a production environment can introduce wear that would be impossible to produce manually. Therefore, slides, gibs, leadscrews, leadscrew nuts and bearings that are subject to high wear due to continuous usage are not covered under this one-year warranty.

Non-warranty service and machine tune-ups

Sherline machines are easy to service, and all replacement parts are available from the factory. However, if in the future you wish to return your machine for any reason to have Sherline repair or adjust it, we will be glad to do so. Our shop rates are very reasonable and our production people have all the proper tools and fixtures to repair or service your machine quickly and return it to factory new condition.

Returning machines or parts for warranty or non-warranty repair

Before returning any tools or parts for repair, please call first and obtain a Return Authorization (RA) number so that we can process your order immediately upon receipt. Remove all non-essential parts from the machine to save shipping weight and return only the portion to be repaired. Wrap and package all parts securely so they cannot move around in shipment. We recommend that heavy parts be double-boxed. Include a brief written description of the desired repair and your return address and phone number or e-mail address inside the box so we can contact you if questions arise. Repairs are normally return-shipped the next workday after they are received.

Instructions: Mill vs. Lathe

These instructions were written primarily for using the CNC system with a mill. A CNC-ready Sherline lathe can also be controlled using the same computer and driver system. You would simply plug the crosslide stepper motor to the X-axis cable and the leadscrew stepper motor to the Y- or Z-axis cable. Writing a program for the lathe is a little different in that you must remember that when you program a distance for the X-axis travel, the part diameter will be reduced by twice that amount. This is because you are taking that dimension off the radius of the part, which is one half of the diameter. Other than that, a little logic on your part should be able to accommodate the differences between the spinning cutting tool/moving part on a mill and a spinning part/moving cutting tool on a lathe. Because of the extra mass of a spinning chuck and part, extra caution should be taken to make sure your program is not going to cause any interference between crosslide or toolpost and the spinning chuck/part once it is running.

Safety Rules for Power Tools

1. Know your power tool—Read the owner’s manual carefully. Learn its application and limitations as well as the specific potential hazards peculiar to this tool.

2. Ground all tools—If a tool is equipped with a three-prong plug, it should be plugged into a three-hole receptacle. If an adapter is used to accommodate a two-prong receptacle, the adapter wire must be attached to a KNOWN GROUND. Never remove the third prong.

3. Keep guards in place—and in working order.

4. Remove adjusting keys and wrenches—Form a habit of checking to see that keys and adjusting wrenches are removed from the tool before turning on any machine.

5. Keep work area clear—Cluttered areas and benches invite accidents.

6. Avoid a dangerous work environment—Do not use power tools in damp or wet locations. Keep your work area well illuminated.

7. Keep children away—All visitors should be kept a safe distance from the work area.

8. Make your workshop "kid-proof" —with padlocks, master switches or by removing starter keys.

9. Do not force a tool—Do not force a tool or attachment to do a job for which it was not designed. Use the proper tool for the job.

10. Wear proper apparel—Avoid loose clothing, neckties, gloves or jewelry that could become caught in moving parts. Wear protective headgear to keep long hairstyles away from moving parts.

11. Use safety glasses—Also use a face or dust mask if a cutting operation is dusty.

12. Secure the work—Use clamps or a vise to hold work when practicable. It is safer than using your hand and frees both hands to operate the tool.

13. Do not overreach—Keep your proper footing and balance at all times.

14. Maintain tools in top condition—Keep tools sharp and clean for best and safest performance. Follow instructions for lubrication and changing accessories.

15. Disconnect tools—Unplug tools before servicing and when changing accessories such as blades, bits or cutters.

16. Avoid accidental starting—Make sure the switch is "OFF" before plugging in a power cord.

17. Use only recommended accessories—Consult the owner's manual. Use of improper accessories may be hazardous.

18. Turn the spindle by hand BEFORE switching on the motor—This ensures that the workpiece or chuck jaws will not hit the lathe bed, saddle or crosslide, and also ensures that they clear the cutting tool.

19. Check that all holding, locking and driving devices are tightened—At the same time, be careful not to over tighten these adjustments. They should be just tight enough to do the job. Overtightening may damage threads or warp parts, thereby reducing accuracy and effectiveness.

20. Don't use your lathe for grinding—The fine dust that results from the grinding operation is extremely hard on bearings and other moving parts of your tool. For the same reason, if the lathe or any other precision tool is kept near an operating grinder, it should be kept covered when not in use.

21. Don't let long, thin stock protrude from the back of the spindle—Long, thin stock that is unsupported and turned at high RPM can suddenly bend and whip around.

22. Wear your safety glasses—Foresight is better than NO SIGHT! The operation of any power tool can result in foreign objects being thrown into the eyes, which can result in severe eye damage. Always wear safety glasses or eye shields before commencing power tool operation. We recommend a Wide Vision Safety Mask for use over spectacles or standard safety glasses.

23. Disconnect the computer from all power sources before opening case—With the computer power switch off, some of the internal components of your computer are still electrical hazards and can provide a severe electrical shock. Disconnect the computer from the power source before opening the case to eliminate this possibility of an electrical shock.

24. Static electricity can damage your computer—Before touching any components inside your computer, be sure you are properly grounded. A spark of static electricity can damage delicate circuits in some components in your computer. Either wear an approved static electricity grounding strap around your wrist or touch the case of the computer with one hand before touching any components with your other hand.

25. Be sure you know what will happen BEFORE pushing the [START] button— Run your program in [BACKPLOT] mode before running the actual part to make sure it will run the path your are expecting. Make sure your machine is in the proper home position before starting the operation.

Electrical Connections

The power cord supplied is equipped with a 3-prong grounding plug that should be connected only to a properly grounded receptacle for your safety. Should an electrical failure occur in the motor, the grounded plug and receptacle will protect the user from electrical shock. If a properly grounded receptacle is not available, use a grounding adapter to adapt the 3-prong plug to a properly grounded receptacle by attaching the grounding lead from the adapter to the receptacle cover screw.

NOTE: The electrical circuit designed into the speed control of your lathe or mill reads incoming current from 100 to 240 volts AC and 50 or 60 Hz and automatically adapts to supply the correct 90 volts DC to the motor. As long as you have a properly wired, grounded connector cord for your source, the machine will operate anywhere in the world without a transformer. This has been true for all Sherline machines built since 1994.

Lubrication

CNC machines require more attention to keeping the leadscrews properly “oiled” than manual machines. When you crank the handles by hand you can quickly realize that a slide is going dry because you can sense the results of the extra drag. A CNC machine will not complain and will always work with the instructions that you gave it. You tell it to run, and it will die trying to please you. The Y- and Z-axis screws can be easily oiled; however, the X-axis screw needs special attention. You can’t see it because it is located under the mill table. The best way I have found to lubricate it is to move the mill table all the way to the right, put some oil on your fingertips and transfer it to the leadscrew. As the table is moved back to the left, oil will find its way to the internal brass nuts. Remember, that any type oil is far better than none. The slides also require constant oiling. I believe these oiling procedures should be done after every four hours of operation. For more on lubrication see page 5 of the *Sherline Miniature Machine Tools Assembly and Instruction Guide* that came with your mill.

Machining Tip—Use multiple end mill holders as a quick way to change tools

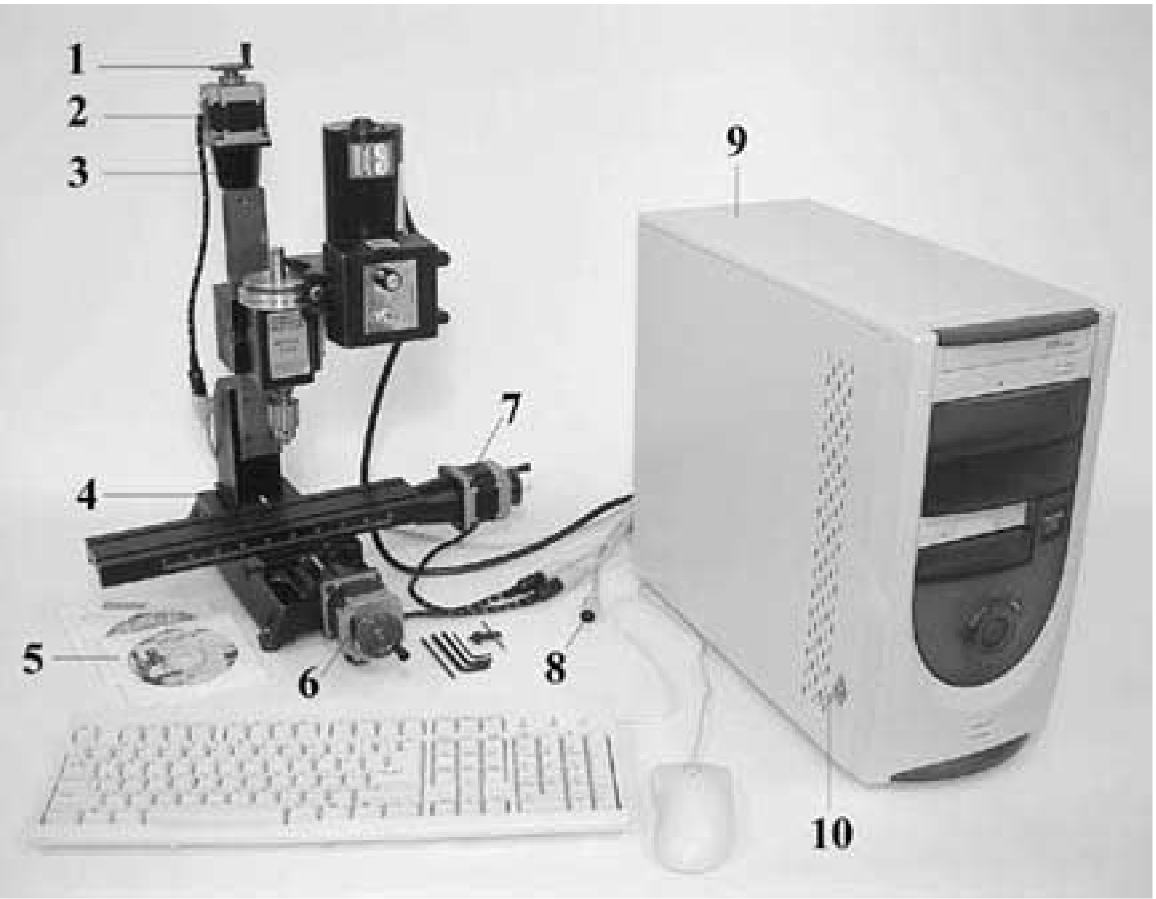
Sherline 3/8" end mill holders P/N 3079 offer a quick and accurate way to change tools. If just one tool holder is used, the cutter height must be adjusted each time it is changed. If multiple holders are used, each with its own tool pre-adjusted to the correct height, tools can be interchanged quickly with excellent repeatability on their height setting.

One hour of free consultation

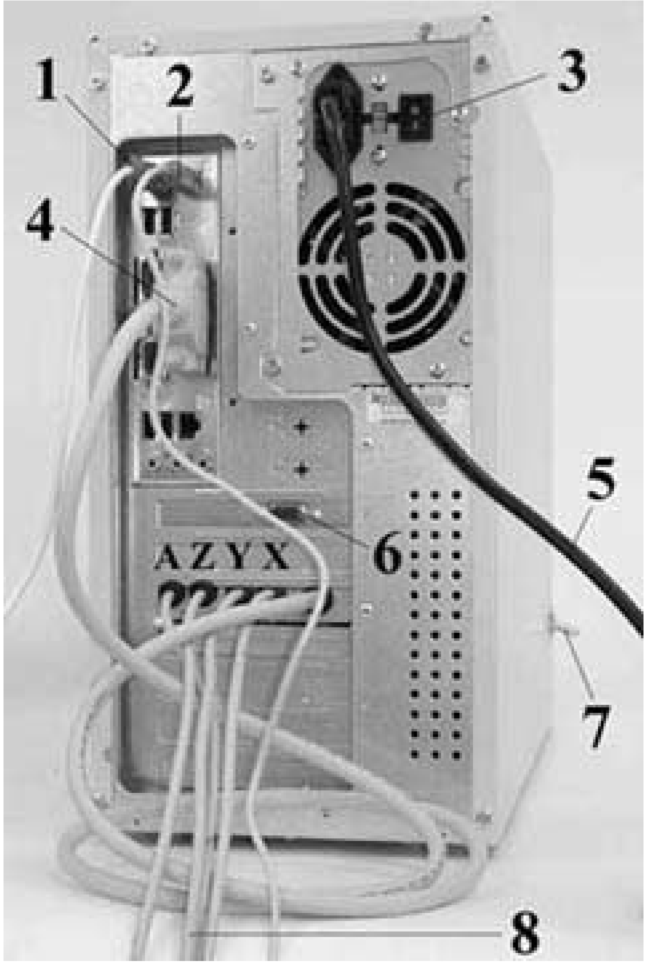
Included in the purchase price of your Sherline CNC mill was one hour of free telephone consultation with Ray Henry that we have prepaid. Ray is an expert in the many phases of the Linux EMC CNC program including installation, CNC programming, the Sherline CNC mill and the Linux operating system. Please do not call Sherline with questions of this type. You'll find Ray a very pleasant and knowledgabe person to deal with. Additional time may be purchased by the hour directly from Ray. Use your time wisely by making every attempt to solve your particular problem yourself by re-reading the instructions and working through the program before calling. Ray can be contacted at rehenry@up.net or (906) 875-6692, 9 AM-9 PM (Central Time Zone).

Remember that Sherline hasn't charged you a single penny for the Sherline version of the EMC program even though we have spent a considerable amount of money and time adapting the EMC program to the Sherline mill. We also don't have anyone available at Sherline with Ray's expertise to answer programming and installation questions. However, if you have a problem with a defective machine or accessory manufactured by Sherline, please feel free to contact us.

System Components and Connections



System components:
1) 1-5/8" manual handwheel
2) Z-axis stepper motor
3) Stepper motor mount
4) Sherline vertical mill with drill chuck, headstock spacer block, hex adjustment keys and spindle bar
5) Backup Linux/EMC installation CD, Sherline instructions CD and Vector32 CAD program CD
6) Y-axis stepper motor
7) X-axis stepper motor
8) Cable for optional A-axis connection
9) Computer with CD-ROM and Floppy drives, keyboard and mouse
10) On/Off switch for stepper motor power supply



Connections on back of your computer:

1) Keyboard connector cable (purple)
2) Mouse connector cable (green)
3) Computer power switch
4) Connector from driver board to parallel printer port
5) Power cord
6) Connect your monitor here (blue)
7) On/Off switch for driver power supply
8) Output cables to stepper motors for A-, Z-, Y- and X-axes

Assembling your system

Getting your system up and running is a simple task. It consists of assembling your mill and connecting it to the computer as follows:

Mill

Unpack and assemble your machine according to the instructions included in the *Sherline Assembly and Instruction Guide* packed with the mill.

Computer

Unpack the computer and enclosed components. Connect the stepper motor cables leading from the back of the computer to each of the three stepper motors. A fourth cable is unused unless you have purchased an optional rotary table that can be connected here as a 4th (A) axis. Plug the purple keyboard cable and the green mouse cable into their receptacles. Plug your monitor to the blue monitor receptacle. Insert the female end of the power cord into its receptacle in the computer and plug the male end into a properly grounded wall socket or surge protected power strip. Turn on the computer and monitor but leave the power switch to the stepper motors in the "off" position for now.

Opening the instructions file on the computer

If you are not familiar with Linux, you will find the desktop that opens when your computer starts up to be comfortingly similar to the Windows® or Mac® desktop you are used to. On the desktop you will see images for several files. Using your mouse, click on the file labeled "Instructions" to open it. The instructions can also be found on the enclosed CD as well as at Sherline's web site at www.sherline.com/CNCinstructions.htm.

Please read all the instructions before attempting to use the EMC program to run your mill. CNC machining is a complicated process, and you will be directed at the proper time to run a program on the machine once you have gained the knowledge you need to do so.

Familiarizing yourself with the control panel

The Sherline Graphical Interface

By Ray Henry and Joe Martin

Start your engines

After having connected everything as shown in the reference photos above you will need to turn on the computer but not the motor drivers (the drivers are the individual circuitry that control the stepper motors) to see how this system runs. The power switch for the computer is located on the front of the computer; it is the large round button in the center. The driver power switch is located on the left side of the computer as indicated in the above photo. Up is on, and down is off.

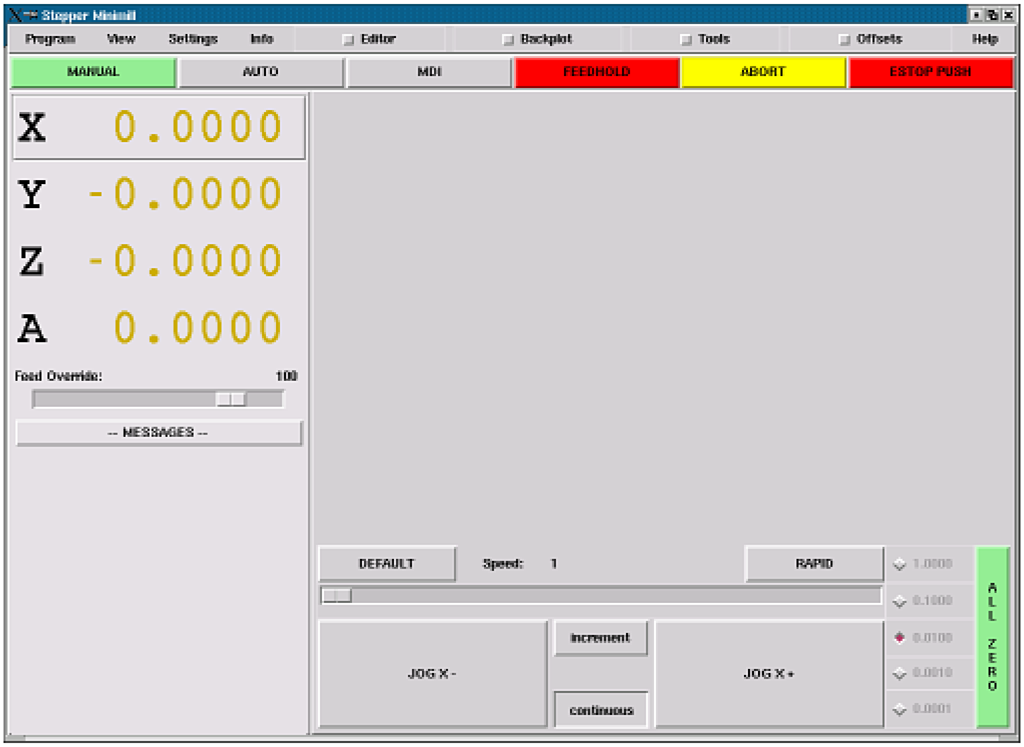
When the computer is turned on, text will start scrolling on your monitor. After a short period of time a box or window will appear that asks for a login and password. The login will be "root" and the password will be left blank. Left click the [Go] button with your mouse and the desktop (the basic opening screen of a program) will appear with many icons (small pictures representing a variety of different programs). Next, single click on the large red "**S**" icon that is labeled "**Inch_freq**.*" This will initiate the start of the Sherline EMC system. It will take several moments to start—sometimes up to 30 seconds to a minute.

*This assumes your mill has inch leadscrews with 20 threads per inch. Should you be using this control with a Sherline machine that is equipped with 1-mm pitch metric leadscrews, choose the "**MM_step**" icon instead.

Inch vs. metric dimensions

By default your Sherline CNC system works in inch dimensions. Should you wish to use metric dimensions to program your machine, even though it has an inch leadscrew, you can do so by entering the code **g21** in place of the **g20** code when writing the program for your part. **G20** indicates inch increments, while entering **g21** indicates you will be using metric increments. The program will automatically make the translation for you. You can also use inch dimensions on a machine with metric leadscrews by using the **g20** code instead of **g21** once the proper version of the EMC has been opened.

I had Ray Henry put together the interface specifically for the Sherline CNC. He also wrote much of "The Sherline Graphical Interface" section of the instructions. Many of the features of the standard EMC program were removed for the Sherline interface because they didn't apply for a mill of this size and type to keep things as simple as possible. Sherline machines do not have limit or home switches and do not have direct control of spindle speeds and such. It is a general motion interface to the EMC. With it you can jog any axis. You can set coordinate systems and tools. You can verify and view the spindle path and motion in a built in program called "Backplot" and dry-run a program before running the machine. The panel is laid out in a sensible manner and is easy to understand.



X 0.0000
Y -0.0000
Z -0.0000
A 0.0000
Feed Override: 100
Speed: 1

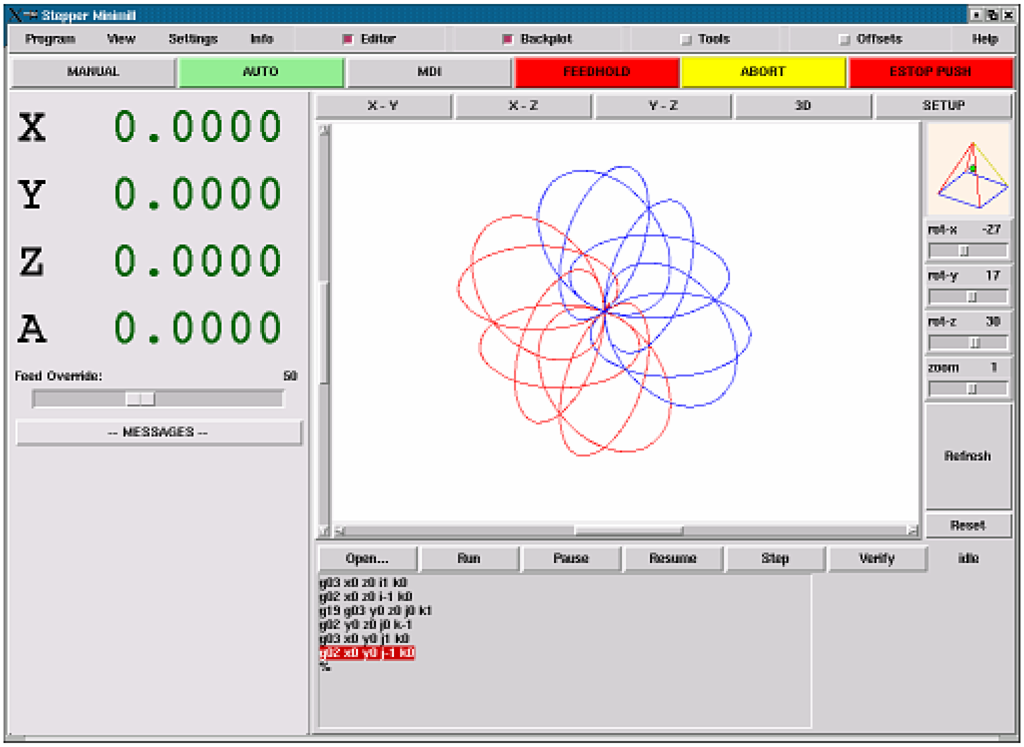
FIGURE 1—The control panel

The screen is laid out in sections. The first row is the menu bar across the top. Here you can configure the screen to display additional information. The small buttons that are located left center of the command indicate that they control the large blank section of the control panel that is sometimes called a "pop-in." This will allow you to view the editor, plotter, tool page and coordinate page. If more than one pop-in is active (button shown as red) you can "toggle" between these pop-ins by right clicking your mouse.

Below the menu line is a line of control buttons. These are the primary control buttons for the interface. Using these buttons you can change mode from [MANUAL] [AUTO] [MDI] (manual data input). You can also use [FEEDHOLD] or [ABORT] to stop or abort a programmed move. When you press the [FEEDHOLD] button it changes color and becomes a [CONTINUE] button. It toggles between running and pausing. Feedhold has the advantage of being able to restart the program from where you stopped it. When all else fails press a software E-Stop.

There are two columns below the control line. The left side of the screen shows axis position, feedrate override, and any messages that are sent by the EMC to the operator. You can add things like current tool number and length, type of position shown, and offsets in effect by looking under the <View> menu.

Directly below the pop-in area on the right side of the screen is the area that is controlled by the chosen mode. In [MANUAL] mode it's configured so that you manually control the slide movements. You'll have your choice of "jog" or "incremental" moves using the same big buttons. In [AUTO] mode it displays a set of buttons that allow you to load a part program, view the current location and [Run] it. In [MDI] mode it displays a line where you can enter your command in standard programming code.



X 0.0000
Y 0.0000
Z 0.0000
A 0.0000

Feed Override: 50

```
g03 x0 z0 i1 k0
g02 x0 z0 i-1 k0
g19 g03 y0 z0 j0 k1
g02 y0 z0 j0 k-1
g03 x0 y0 j1 k0
g02 x0 y0 j-1 k0
%
```

FIGURE2—The screen with BACKPLOT displayed

If you compare the second screen shot with the first you can see that we are in now auto mode, a program has been run, and the tool path is being displayed using Backplot.

Notice that the display position numbers have changed color. When you start first load the EMC program, the slide location position will be displayed in gold. This means that a homing routine has not been done yet. Once an axis has been homed, the numbers turn dark green.

Feedhold and Feedrate Override

You can operate feedrate override and feedhold in any mode of operation. Override will change the speed of jogs or feedrate in manual or MDI modes. A feedhold will be initiated if you click the [FEEDHOLD] button or if you press the pause button on the keyboard. Once in feedhold, the same actions will return you to the speed the machine was moving before the feedhold.

You can adjust feedrate override by grabbing the slider with your mouse and dragging it along the groove. You can also change feedrate a percent at a time by clicking in the slider's groove. In auto mode you can also set feed override in 10% increments by pressing the top row of numbers on your keyboard. The number 1 will set feed rate override to 10 percent of programmed or jog speed. The number 9 will set 90 percent.

Messages

The message display located under the axis positions is a sort of scratch pad for the EMC. If there are problems it will report them there. If you try to home or move an axis when the [ESTOP] button is pressed, you'll get a message that says something about commanding motion when the EMC is not ready. If an axis faults out for something like falling behind, the message pad will show what happened.

If you want to remind yourself to change a tool for example, you can add a line of code to your program that will display in the message box. Example: (msg, change to tool #3 and press resume) will display "change to tool #3 and press resume" in the message box. The "msg," comma included is the command to make this happen; without "msg," the message wouldn't be displayed in the message box.

To erase messages simply click the message button at the top of the pad or hold down [Alt] and press [m].

Manual Mode

Manual mode shows a set of buttons along the bottom of the right-hand column. A green button labeled "ALL ZERO" has been added to designate the present position as the home position. I felt that a machine of this type would be simpler to operate if it didn't use a machine home position. This button will zero out any offsets and will home all axes right where they are.

Axis Focus (Active)

Axis focus is important in manual mode. Notice that in Figure 1 above you can see a line or "groove" around the X-axis position display. This groove says that X is the active axis. It will be the target for jog moves made with the plus and minus jog buttons. You can change axis focus by clicking on any other axis display. You can also change axis focus in manual mode if you press its name key on your keyboard. Case is not important here. [Y] or [y] will shift the focus to the Y-axis. [A] or [a] will shift the focus to the A-axis. To help you remember which axis will jog when you press the jog buttons, the active axis name is displayed on them.

Jog Mode

The EMC can jog (move a particular axis) as long as you hold the button down when it is set for “continuous,” or it can jog for a preset distance when it is set for “incremental.” You can also jog the active axis by pressing plus [+] or minus [-] keys on the keyboard. Again, case is not important for keyboard jogs.

The two small buttons between the large jog buttons let you set which kind of jog you want. When you are in incremental mode, the distance buttons come alive. You can set a distance by pressing it with the mouse. You can toggle between distances by pressing [i] or [I] on the keyboard.

Incremental jog has an interesting and often unexpected effect. If you press the jog button while a jog is in progress, it will add the distance to the position it was at when the second jog command was issued. Two one-inch jog presses in close succession will not get you two inches of movement. You have to wait until the first movement is complete before jogging again.

Jog speed is displayed above the slider. It can be set using the slider, by clicking in the slider’s groove on the side you want it to move toward, or by clicking on the [Default] or [Rapid] buttons. This setting only affects the jog move while in manual mode. Once a jog move is initiated, jog speed has no effect on the jog. As an example of this, say you set jog mode to incremental and the increment to 1 inch. Once you press the [Jog] button it will travel that inch at the rate it started.

Keyboard Jogging

When the control is in manual mode you can also control the slides using the keyboard cursor control arrows. The slides will move in the direction the arrows point, and the Z-axis can be raised or lowered using the [Page Up] or [Page Down] keys. The speed the slides move at will be determined by the manual feed rate setting.

Auto Mode

Yes, you can move the machine around in manual mode. You can even do some tolerable milling in manual mode, so long as you work a single axis at a time and set jog speed for the feedrate that your tool, machine, and material need. The real heart of CNC machine tool work is the auto mode.

Sherline’s auto mode displays the typical functions that people come to expect from the EMC. Along the top are a set of buttons which control what is happening in auto. Below them is the window that shows the part of the program currently being executed. As the program runs, the active line shows in white letters on a red background.

The first three buttons, [Open], [Run], and [Pause] do about what you’d expect. [Pause] will stop the run right where it is. The next button, [Resume], will restart motion. They are like “feedhold” if used this way. Once [Pause] is pressed and motion has stopped, [Step] will resume motion and continue it to the end of the current block. Press [Step] again to get the motion of the next block. Press [Resume] and the interpreter goes back to reading ahead and running the program. The combination of [Pause] and [Step] work a lot like single block mode on many controllers. The difference is that [Pause] does not let motion continue to the end of the current block.

Feedrate Override and Auto Mode

The number buttons along the top of the main keyboard will change feed rate whenever you are in auto mode. Pressing the gravé [`] gets you zero percent feed rate. One through zero gets you ten through 100 percent feed rate. These keys can be very handy as you approach a first cut. Move in quickly at 100 percent, throttle back to 10% and toggle between [Feedhold] and 10% using the keyboard's pause key. When you are satisfied that you've got it right, hit the zero to the right of nine and go.

Verify

The [Verify] button runs the interpreter through the code without initiating any motion. If verify finds a problem it will stop the read near the problem block and put up some sort of message. Most of the time you will be able to figure out the problem with your program by reading the message given and looking in the program window at the highlighted line. Some of the messages are not real helpful. Sometimes you will need to read a line or two ahead of the highlight to see the problem. Occasionally the message will refer to something well ahead of the highlight line. This often happens if you forget to end your program with a %, m2, m30 or m60.

Restart

To the right of the program window is a space that will show a set of restart functions. These pop in whenever a program is aborted or an EStop happens while the interpreter is running or paused. You can also display or hide these using the view menu.

I've got to warn you that [Restart] can be risky. [Restart] reads down through the program to the restart line. While it does this it sets up its "world model" so that it can resume right where it stopped, but only if the tool is right where it was. Most of the time, when we abort or EStop it's because something went wrong. Perhaps we broke a tool and want to change it. We switch to manual mode and raise the spindle, change tools, and assuming that we got the length the same, get ready to go on. If we return the tool to the same place where the abort was issued, the EMC will work perfectly.

It is possible to move the restart line back or ahead of where the abort happened. If you press the [Back] or [Ahead] buttons you will see a blue highlight that shows the relationship between the abort line and the one on which the EMC stopped. By thinking through what is happening at the time of the restart you can place the tool tip where it will resume work in an acceptable manner. You will need to think through things like tool offsets, barriers to motion along a diagonal line, and such before you press the [Restart] button. A restart is a good time to use feed rate override and the pause key on your keyboard.

MDI

MDI mode allows you to enter single blocks and have the interpreter execute them as if they were part of a program—kind of like a one line program. You can execute circles, arcs, lines and such. You can even test sets of program lines by entering one block, waiting for that motion to end, and then enter the next block.

Below the entry window, there is a listing of all of the current modal codes. You can also write this list to the message box in any mode if you look under the info menu and select active g-codes. This listing can be very handy. I often forget to enter a g00 before I command a motion. If nothing happens I look down there to see if g80 is in effect. G80 stops any motion. If it's there I remember to issue a block like g00 x0 y0 z0. In MDI you are entering text from the keyboard so none of the main keys work for commands to the running machine. [F1] will Estop the control.

Pop-in Editor

The editor is rather limited compared to many modern text editors. It does not have "undo" nor "paste" between windows with the clipboard. These were eliminated because of interaction with a running program. Future releases will replace these functions so that it will work the way you've come to expect from a text editor. Beware that you could lose the portion of your program that is highlighted by pressing any key. Use the save command often.

It is included because it has the rather nice feature of being able to number and renumber lines in the way that the interpreter expects of a file. It will also allow you to cut and paste from one part of a file to another. It will also allow you to save your changes and submit them to the EMC interpreter with the same menu click. You can work on a file in here for a while and then save and load if mini is in auto mode. If you have been running a file and find that you need to edit it, that file will be placed in the editor when you click on the editor button on the top menu.

Backplot

[Backplot] will show the tool path that can be viewed from a chosen direction. "3-D" is the default. Other choices and controls are displayed along the top and right side of the pop-in. If you are in the middle of a cut when you press one of these control buttons the machine will pause long enough to re-compute the view.

Along the right side of Backplot is the pop-in that you can display with the [SETUP] button. This will show a small graphic that tries to show the angle you are viewing the tool path from. Below it are a series of sliders that allow you to change the angle of view and the size of the plot. You can rotate the little position angle display with these. They take effect when you press the [Refresh] button. The [Reset] button removes all of the paths from the display and readies it for a new run of the program but retains your settings for that session.

Pop-in Tool

The tool page is pretty much like the others. You can set values here and they become effective when you press the [Enter] key*. You can't change tool offsets while the program is running or when the program is paused. The [Add Tools] and [Remove Tools] buttons work on the bottom of the tool list. Once a new tool has been added, you can use it in a program with the usual commands.

*Always use the [Enter/Return] key at the right side of the normal alpha keyboard, not the [Enter] key at the bottom right of the numeric keypad.

Limit switches

I felt that limit switches were unnecessary when using stepper motors with limited torque such as these. They have to be mounted in areas where they are exposed to chips, oil and coolant creating more problems than they eliminate. The Sherline mill is fitted with “Hard Stop” screws on the X and Y axes that limit travel. These stops are the heads of cap screws fitted to the table and base and positioned to stop a runaway axis without damage. If you make a programming error that would drive the table into a crash situation by over-traveling, the table will encounter these hard stops before damaging the leadscrews and nuts.

You could also locate a home position from these hard stops by stalling the stepper motors against them. Of course you should do this in manual mode and by bringing the table against the hard stop carefully in slow speed and "buzz" the motor against it. I personally would find a different way of locating a home position, using the dials on the handwheels for example, but it is an option for unusual situations

It should also be noted that the hard stop screw would have to be removed from the left underside of the table in order to have the full nine inches of advertised X-axis travel. With the stop screw in place travel is about 8.65".

Manual vs. CNC

Let’s think about what we have to do to write a program for CNC:

1) Create a spot on your storage system.

2) Write a program.

3) Make sure it doesn’t have any errors using the Backplot system

4) Dry run the program with the spindle well out of the way so it can’t possibly crash.

5) Now, accurately align the machine with the work so they work in unison when the program is run.

Push the button and have a cup of coffee while your little Sherline just works its butt off for you making good parts.

When you think about this, it has the same steps as to do it manually. Compare:

1) Make some special tooling to hold the special shape. With CNC you can hold raw stock in a vise and machine the shape you need.

2) We’d have to store the special tools in case we wanted to make another one.

3) You’ll make a lot of scrap with a manual machine if you don’t double check your work before making the first cut, and you will not have a computer to help you as you do with EMC.

4) No matter how you do it, you always have to line up the work to the machine.

5) Don’t put the coffee on yet because your real work has just begun while our boy with the CNC is pouring a second cup and making 7th part of ten pieces. (Oh, how I wish it could be this easy at the start!)

How to make a file for our programs and use “Backplot” to check them

1) To create a new file or work with any existing programs we have to be in the [AUTO] mode. Notice that a new line of commands appears along the bottom in this mode. The [Open] button is located in the along this row. Note: the method I'm describing in A) to E) below is to set up your computer for future use and isn't the necessary steps you'll use in the future. It's possible to create files in the wrong folders, and if you use this method you will always have your new programs in the proper folder.

> A) With power to the stepper motors turned off [Open] the existing program called "3dtest.ngc."*
>
> *This file comes with EMC. If you are using a computer supplied by Sherline, you should instead open the file called "0-new." The instructions in sections D) and E) below describe how to change "3dtest.ngc" to "0-new." Since Sherline owners will have this file already provided for them, they can skip sections D) and E) when they come to them and go on to section 2).
>
> B) Go to the Pop-in [Editor] command and open along the top row.
>
> C) The window will display the entire program.
>
> > 1) Before we go any farther I want you to open the [BACKPLOT] program that can be found on the upper line and a blank section will appear in the Pop-in section. The [Backplot] feature will allow you to visually check a program without actually running the stepper motors. Right clicking your mouse controls the program that is viewed in this section of the control panel. You can toggle between the editor and the backplot easily. We now have the perfect setup to learn CNC and it all fits on a single desktop. You can view and edit your program or view the moves the spindle will make while your program is running.
> >
> > 2) Now that we have a program loaded and the Backplot program running, let's take a break and test our setup. Make sure that [ESTOP] is active with the button highlighted and execute [Run].
> >
> > 3) Observe that the programmed moves can be viewed in the backplot box and the viewing area can be controlled with a variety of controls. This is very useful in checking out the programs you'll be writing in the upcoming lessons. The line of the program that is highlighted in red in the lower area of the control panel is the line of code that is being processed by the control as the computer runs the program. Note that the EMC program is processing more than a single line of code at the same time; therefore, this line should only be considered "approximate."
>
> D) For those of you who do not have a Sherline-supplied computer we are now going to create a file we will call "0-new." (If you have a Sherline computer with the "0-new" file already on it, open it and go to section 2) below.) First go back to the [Editor] window by right-clicking your mouse. By having a "0" as the first character of the file name it will always keep this file at the beginning of the program list. This makes it easy to find, because you are going to use it again and again. From the [Edit] window, use [File] then [Save as] (located along the top of the edit window) to save our existing program as "0-new."

E) Next, highlight (with the mouse button held down drag the cursor across the entire area you wish to highlight) and delete (press enter) the existing program with your mouse. Replace that program by typing “Save as another file name.” This will remind you to only use the “0-new” program to set yourself up for creating a new program. Use the [File>Save] command. Remember, to make any changes permanent you must save the changes before closing the program.

2) We’ll now use the [Save As] command to save the “0-new” file with the new name “test-g02”. By using the [Save As] command you also leave the “0-new” program as is to use again. You can create your own files and name these files as you wish from this point on. Note: “test-g02” is the first program you’ll write.

3) You have to get the new program loaded into the control (machine) in order to use it.

4) To accomplish this first use [File] [Open] to open your new “test-g02” program. The program will now be loaded in the control panel with the program name displayed.

All of this BS isn’t as bad as it seems and you will quickly create programs using this information along with “0-new”

Linux Tip—Saving a program to a floppy:

To save a program to a floppy disk, open the program you wish to save within the editor in EMC. Go to “File”, and “Save As.” Double click the “floppy” directory, then click on “Save.” Make sure that you have the correct name in the "File Name" field.

Maintaining your EMC programs directory

Many times you will create a number of programs that are similar or for testing use. After a while these programs can accumulate into a large repository that makes it difficult to find the program you are looking for. If you wish to delete a program the easiest way to do so is to double click the folder icon on the desktop that says "programs". Find the program(s) you wish to delete. Right click on them and then left click on the [Delete] option. This will permanently delete the program selected. Take great caution when deleting programs! You do not want to accidentally delete a valued program.

Linux Tip—If you're familiar with Linux you can navigate to your programs directory manually by using the following command:

```
# cd /usr/local/emc/programs
```

If you wish to delete a program, you can do so by using the following command:

```
# rm programname.ngc
```

Part Two

Programming Sherline CNC Machines with EMC

Although there is quite a bit of well-written instruction on how to program on the Linux CNC site, I believe you should only go there to learn beyond the limits of my instructions. Having too many instructors on a subject that you know nothing about isn't a good idea at this time. At the end of my instructions there are lists of the many words and commands available to you taken from the Linux EMC site; however, there isn't any reason to remember them now. I want you to learn them as you use them. The commands and words that I'll be using are so basic to writing code that they'll soon become a new language.

Print a copy of these instructions

Although you could jump between the control screen on your monitor and the instructions, it would be a cumbersome way to work. You can print a copy of these instructions from the CD using any computer with a printer hooked up to it, or you can print a copy from your Linux computer by unplugging the servo driver connector from the parallel port and plugging in a printer instead.

To make it easier and to save paper, we have saved just the portion of the instructions that refers to the programming as a separate file you can print out. You have your choice of three formats: They are called **CNCprint.doc** (MS Word), **CNCprint.pdf** (Adobe Acrobat*) and **CNCprint.html** (opens in any web browser).

*Your Linux computer already has a program installed that will allow you to read .pdf files by clicking on them. In case you need to install Adobe Acrobat Reader on your Windows® computer, a copy is included on the Sherline Instructions CD.

Let the BS end and the programming begin

From this point on, you have to learn programming at your CNC system. You can't learn how to control these systems in your favorite easy chair; therefore, you should take a printed copy of "Part Two" to your system and enter, test and understand each sample program before going on to the next. Write a few similar programs, with each being different enough to make it challenging and constantly test yourself as you go. No shortcuts!

Are you ready?

About now I'd be "chomping at the bit" to see something happen so lets do something the best manual machinist couldn't do to start. Hell, anyone can cut a straight a straight line so let's make a circle. With the servo controller power off, manually crank the handles and approximately center the X-, Y- and Z-axes. Leave the power off to the servos until we get the program all checked out with Ray Henry's backplot command. Ray Henry has been a sparkplug for the EMC group for some time now and we all owe him a debt of gratitude.

G-code programming

These are the basic g-codes that you'll be working with. There is a complete listing of them at the end of this section copied from the www.linuxCNC.org site. There isn't any reason for you to remember them all at this time, because I want you to thoroughly learn them as you use them; however, you can get a general idea of how the system works by reading them now. They are quite simple. Remember that modal g-codes remain in effect until the control receives a new g-code that over-rides the present code. Modal g-codes control function, direction or speed and, in general, cannot be used in the same line of code with other modal g-codes for obvious reasons.

g00 rapid positioning
g01 linear interpolation
g02 circular/helical interpolation (clockwise)
g03 circular/helical interpolation (c-clockwise)
g04 dwell
g10 coordinate system origin setting
g17 xy plane selection
g18 xz plane selection
g19 yz plane selection
g20 inch system selection
g21 millimeter system selection
g40 cancel cutter diameter compensation
g41 start cutter diameter compensation left
g42 start cutter diameter compensation right
g90 absolute distance mode
g91 incremental distance mode

There are no erasers in the machine trades

One other quick thing to note, it should be obvious that I could have provided a disk with all the examples typed in or to be downloaded, but accurately typing in a program is as much a part of learning CNC as making chips. When you misspell a word in a letter, the reader of the letter usually knows what you were trying to say and will not even notice it. In the world of robots, our robots can self-destruct trying to carry out your foolish typing error. This kind of typing requires them same precision as machining. There are no erasers in the machine trades. Only people who express themselves with their mouths and writings have the use of this marvelous tool. Machinists express themselves with their work, and that's the way I like it.

In this exercise the home position is going to be where the tables were located when the EMC program was turned on, which is why I had you center the machine slides and is called x0,y0 and z0. The X-axis is the table that will move to your left when it moves in a positive direction. The x0, y0 and z0 is also designated in the backplot box with arrows pointing towards the positive direction away from this position. No more excuses, we're going to program that circle I promised you. Note: Although the EMC program doesn't need the "0" after the "g" in g00 or g01 I'll use them because they are the designations all popular CNC systems use. Also, I prefer lower case letters because you are less likely to make errors and mix them with numbers. The capital letter O and the number 0 can be difficult with some fonts and are located close together on any keyboard. You are not writing a letter when you program and it has to be accurate.

Circle program

Note: In these instructions, the code you should retype into your program will always shown in **bold face red** type.

The entire program:

```
 %
(g02 circle program)
g01 g90 x0 y0 z0 f2
g02 x0 y0 i-.5 j0
g01 g90 x0 y0 z0 f2
%
```

1) **%** –Always start a new program with a % sign. This informs our robot that a program is coming in. Each time you press [Enter], the program will start a new line, so [ENTER] it is, and we're off and running.

2) **(g02 circle program)** –Words in parentheses are not part of program and are used to help the operators quickly understand the program. You can enter notes into a program at any point to remind yourself or another operator in the future why the upcoming code was entered.

3) **g01 g90 x0 y0 z0 f2** –What we just ordered our robot to do is go to the home position if it's not already there.

    A) **g01**–Enters a mode where the feed (speed) is controlled by "f" which can be entered. Also note that the slow feed of f2 was entered so you could fully appreciate just what the machine slides do to generate a circle. The feed could be set for a faster rate for the backplot test and slowed down for the machine test if desired.)

    B) **g90**–Orders the machine to use the absolute coordinate system where the position will always be referenced from a zero point. Again, you can zero the axis by going to [Pop In] [Offset].

    C) **X0, y0, z0** is self evident. If the slides weren't in the 0 position they would have moved to it.

This is the basic information that all programs should start with. If I had entered a g00 command to start with (orders the machine to move to the programmed position at fastest speed) it would be a quicker but one that can be more risky to start with. Modern CNC machines can perform g00 moves at speeds of 1400 inches a minute (35mpm), and you're making a mistake by not taking advantage of this fact, but we're not pros at this time and I want you to learn about feeds. Hit the [ENTER] key.

4) **g02 x0 y0 z0 i-.5 j0** – This line is the code that is going to actually produce the x y movements to create that circle we have been talking about. In this case, all the previous commands are still in effect except the g02 is replacing the g01.

    A) **g02** is ordering our robot to cut a circle or start an arc with a clockwise (CW) move. For our faithful robot to carry out this order it needs more information or it's going to start complaining and sending you nasty little notes and refuse to work. Remember that the g02 will remain in effect until a g01 is called for.

B) The g02 the program needs the **x, y** position entered as to where this arc will end, and since we are creating a circle it obviously will end where it started so the present position is entered.

C) Here are a couple of new commands; **i and j,** and they are used to describe the center location of our circle. These dimensions are given as the actual distance the center is located from the present position. It is very important to understand that our present position is not the center of this circle and only a starting point for this circle to be created. The i represents where the center of the circle falls on the x-axis and the j represents the y-axis.

I would also like to add at this time that the distances calculated and entered for computer generated shapes have to be extremely accurate, and a 0.0002″ error could stop your program from running. This doesn't have anything to do with the accuracy you need; it pertains to the accuracy the computer needs to calculate the many points it creates to generate the x y movements for this one circle. This shouldn't be any problem because of the low cost calculators we all have today that figure things to many decimal places. Hit [ENTER]. (This is the last time for the [ENTER] command to be instructed. From now on you will hit [Enter] at the end of each line of code.)

5) **g01 g90 x0 y0 z0 f2**–I always add a line of code with a g01 or g00 and a x,y,z to return to the home position. The "g" commands stay in effect until they are overwritten by another g command. In this case it would have left the control in a circle command if we didn't add this line of code which wouldn't be a good practice.

6) **%** -To inform our robot that this is the end of our simple program to do a very complex thing we enter a % just as we did to start the program.

7) Now we have to load this program for your machine to run. [File] [Save and Load] In this case we are not ready for prime time, and we are going to check our program with [Pop In] [Backplot].

8) One more check –the program should look like this:

```
%
(circle g02)
g01 g90 x0 y0 z0 f2
g02 x0 y0 i-.500 j0
g01 g90 x0 y0 z0 f2
%
```

9) **Go for it**, click on [Run]. The backplot program should start tracing the path the spindle will travel in relation to the work. If you got one of those little messages and it didn't run, check for typing errors and correct them. Right-click [Editor] [file] then [Save and Load] [Run] until it works.

10) The time has finally come to watch this little beauty in action, so turn on the machine control and "**let her rip**." Notice how the slides are constantly changing speeds as each axis reaches its apex and changes direction. The circle is also being generated in the CW direction as the g02 requires. Run this little program several

times and take the time to really understand what's going on. From the control panel you can single step the program through and run things until you're bored.

You had your fun and it's time to go to school again.

Now guys and girls, it's time to go back as many times as it is necessary until you are absolutely positive you know exactly what every letter of your program does. I'm not looking over your shoulder, and the only way you can act foolish is by being on page 10 when you should be on page 3.

What a difference a minus sign can make

The first thing I want you to learn is how important that minus sign can be. (Also, a misplaced decimal point can easily cause a catastrophic machine event.) First, turn off the machine because there isn't any sense in having it run while we're doing this exercise. It'd also be distracting. Run your existing program through once again and generate the circle you started with. Leave the circle there and don't [Reset] until we are done with this entire exercise. Right-click to Edit your program and the only thing I want you to do is remove the minus sign from the i value in the g02 command. [Save and Load] and [Run]. I'll bet you didn't think it was going to do that! Just think what could have happened to that expensive fixture if you were controlling a 30HP mill. A quick way to the unemployment line. I'm now going to give you the golden rule #1 of any CNC shop.

Don't ever press the green **[Start]** button on a CNC machine unless you know exactly what the machine will do **BEFORE** you

PRESS IT!

Now let's take a closer look at that minus change we made and we'll see what happened is really obvious. By changing the "i" to a positive number the center of the radius is now moved to the positive side of our zero starting point, which in this case is a total distance of 1". The g02 command still went in a CW direction and the only difference is that the circle was generated on the positive side of the X plane. Are you up for your first test? You better be because I didn't write all this stuff for my own enjoyment.

From what I explained to this point, you should be able generate a 4-leaf clover on the backplot screen. Don't forget to use save and reload as you make these changes. You're on your own.

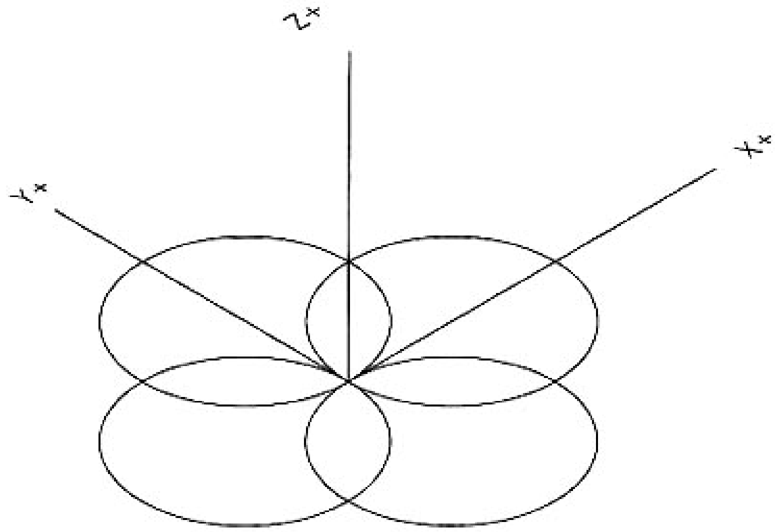


Figure 1

If you’re reading this, I’ll assume you passed the test, so now take your second test by erasing your program with the backspace key and doing the entire exercise over from memory**.**

Meet Millie

Think about what you accomplished at this point. You’re beginning to understand how we humans can order our robot around. I think we should give our little robot a name. Seeing it’s a mill we’re working with and I’m a male, I think I’ll call mine Millie. It will be the only thing with a female name in my lifetime that I stand a chance of controlling. I really want you to know the stuff I taught you blindfolded because we are heading for the hard, oops, I mean *interesting* part.

I’m buying

I’m a little bored with this exercise myself so let’s drop over to route g41 and see what’s going on. The programmers that reside on g41 are a little on the snobby side because they think they have some special talent that us regular guys can never learn. That was probably true when they learned it, but they didn’t have Joe Martin teaching them. In Joe’s class no one is left behind, and let me promise you this guys, we are going move into their neighborhood so fast they will think we always lived there. But for the hell of it, let’s stop by their fancy cocktail lounge and have a beer so they can get used to us. I’m buying.

Tool height offsets

Before we get too involved in the more complex cutter diameter programming, I want you to learn how to deal with the difference in tool lengths. It is obviously simpler to have a single zero point for all tools, and that is what the EMC program allows you to do. The length and diameter of each individual tool is stored in a file called “Tools,” located along the top row of the control panel. The tool length is usually called up in the first z-axis move using each individual tool by entering a g43 h1,2,3,--. The g43 tells the computer to use the tool length (h) stored in the line (1,2,3,---) of the tool data file. A g49 is entered when the z-axis is brought back to its home position to cancel out the tool length compensation that was called up when you first started using the tool.

From a practical standpoint, I like to make the shoulder located behind the ¾-16 spindle nose thread the zero point on the Sherline mill and use screw-on Sherline tool holders. This allows you to measure the tool length directly with calipers by measuring the overall length of the holder with the tool mounted. Also remember that the home stopping position for the z-axis shouldn't be zero. It should be a positive number that allows you to change the tool if necessary. The x0, y0 and z0 position should be chosen carefully and be related to the part that is being programmed, not to a convenient place to change tools.

Write a couple of lines of code with and without the tool length compensation to test what you have just learned. Run it using the backplot to fully understand it before going on, and, above all, always remember to cancel the tool length compensation after using it.

Time to get serious

Let's think about just what g40, g41 and g42 does. To begin with g41 (g42 is the opposite of g41) will locate the spindle to the left of the programmed edge that you want to machine according to the d (tool dia.) amount entered into the tool data table. G40 simply cancels the g41 or g42 commands.

Computers can keep you from ruining your part

These commands allow you to put the actual dimensions of the part you want directly into the program and then it will take the diameter that you entered into a [Tools] file and automatically correct the path of the cutter so that the edge of the cutter is always located on the edge on the part. Your computer will be making thousands of calculations a second to do this, and old Millie will just bust her hump for you to make this happen if you follow their rules. If you don't, they'll start sending you those nasty little messages again, but look at it as a blessing. They'll keep you from wrecking your part.

Another thing to keep in mind is that one of the great benefits of the tool offset commands is being able to use end mills with slightly different diameters without having to calculate all new tangent points. Before CNC, when NC was the only system available, there was a surplus of used end mills on the market that had not been re-sharpened because it was cheaper to scrap out dull end mills without re-sharpening them than to write long complex programs to account for their smaller diameter.

Don't ask for impossible

The first thing you have to learn about using the g41 or g42 cutter comp commands is that you can't ask your machine to do the impossible. What is impossible is cutting an inside "V" with an end-mill.

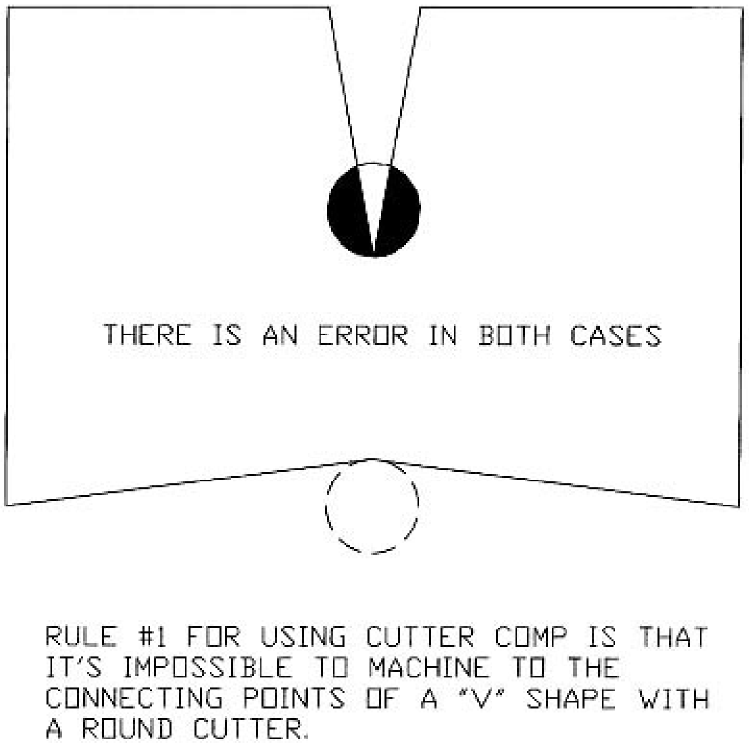

THERE IS AN ERROR IN BOTH CASES
RULE #1 FOR USING CUTTER COMP IS THAT IT'S IMPOSSIBLE TO MACHINE TO THE CONNECTING POINTS OF A "V" SHAPE WITH A ROUND CUTTER.

Figure 2

If you firmly imbed this into your mind, you'll save the many hours of wasted time that I spent learning it. Of course I knew that this is impossible, but I kept asking my computer to do it when I was entering the cutter offset mode. As soon as a g41 or g42 is entered into the program you can't ask your machine to cut "V" shapes without an arc being programmed that is at least the radius of the cutter you are using. What confused me was the cutter diameter that was retrieved with the d-command had an effect on whether you would or would not have an error. I believe I truly understand it now and had to rewrite several pages of garbage I wrote with confusing rules.

Safety first

I believe you shouldn't approach the work at this time in a rapid g00 mode for safety reasons; therefore, all my examples will include a short section of controlled feed rates before the point of contact is reached. My program examples will always be complete and will run on the backplot program that comes with your system. I have yet to find a program that checks out with this program that wouldn't run on the Sherline mill if the slides were located in the proper position before the program was run. The more I work with the EMC program the more I like it.

Flying Sherline Airlines requires you to learn how to land before flying

The whole trick in using g41 or g42 is getting to the starting point in a very specific manner. Also, remember that our computer is very smart and looks ahead at what you plan to do when you enter cutter comp. If you ask it to do the impossible, it will send you a message. What we want to accomplish is approach the starting point in two separate moves so that we are coming into the part like a plane on final approach to an airport. The plane flies to the first point, where it lowers it landing gear (g41—a move where the cutter comp is entered during the move) and feeds (g01) to the final landing point with its wheels (cutter) in position to land. Come in too steep and you'll create a deep V cut that is impossible to machine. (See Figure 2.)

You can overcome this error by programming a 90° radius cut to the part. Of course, the programmed radius must be larger than the radius of your cutter. To think of a programmed move in aircraft terms you are "flaring out" for touchdown. This has a second advantage from a machining standpoint of not leaving a slight undercut that's created when a cutter is brought up to edge and immediately makes a sharp turn. This undercut is caused by cutter deflection.

The coming programming examples don't use this method of bringing the cutter up to the part because it wasn't necessary in my examples. In the real world, this will be a problem that you'll regularly encounter, and you'll have to use every trick in the book to produce the parts that are needed.

For the time being, remember that the closer you align your cutter (aircraft) with the directional edge that you want to machine (runway) the better the landing. Remember, in the world of CNC there is no gravity, and you can touch down any way you please.

Let's go back and add g41 to the circle program. Modify your circle program to read like this. Refer to the drawings with examples in this section.

The entire program:

```
%
(circle using g40, g41, g42)
g00 g90 g40 x1 y1 z1
g42 d1 x.5 y1
x0
z0
g01 z-.3 f10
g03 x0 y1 i0 j-.5
g00 z1
g40 x1 y1 z1
%
```

Now I'll go through the program one line at a time.

Note: Before this program can be run the tool offset has to be entered. Go to [TOOLS] and enter 0.250" (6.4 mm) for the tool length, and 0.375" (8.4 mm) for the tool diameter. You must use the [Enter] key after entering each change in order for the change to be entered into the table. Close page.

1) **%** –Last time—Start program

2) **(circle using g40, g41, g42)** –Last time—Program Name

3) **g00 g90 g40 x1 y1 z1** –Home

4) **g42 d1 x.5 y1** –End Mill (aircraft) approaches part as tool compensation is entered (gear down). **d1** will define the offset as it moves to this new g42 position. We're still traveling too fast for landing.

5) **x0** – We're directly above the touchdown area and come to a halt.

6) **z0** – Fortunately we're flying the new Osprey and they can come straight down.

7) **g01 z-.3 f10** –Landing speed and touchdown from a vertical plane.

8) **g03 x0 y1 i0 j-.5** –We cut a quick circle just like last time and we're out of here.

9) **g00 z1** –Straight up

10) **g40 x1 y1 z1** –And home.

11) **%** –Park it and have a coffee. Good flight!

The diameter was generated as before, but this time the final size of this diameter will be controlled by the diameter entered in to the d1 location. By entering a diameter larger than the actual tool, which is 0.375", we are assured that the diameter will not come out undersize, because it keeps the spindle farther away from the finished edge. This can give you the opportunity to take a second pass and allow you to bring it accurately to size by changing the d1 value. You have to admit, I'm a clever old bastard.

I think it's a good time to take a break from circles and cut something straight before you get motion sickness. Create a new file for this flight. How about a nice simple rectangle with cutter comp of course? The material we'll be serving today is 1" x 1" x 2" (50 mm x 50 mm) 6061 T6 aluminum, and our cutting speed will be 6 IPM. In the event of a crash, prepare to pick up your paycheck on the way out the door. Have a nice flight.

On this flight we are going to take a couple of quick passes around our 2 inch square block and machine a 0.150" boss on it to 1.800" square with a 0.375" end mill.

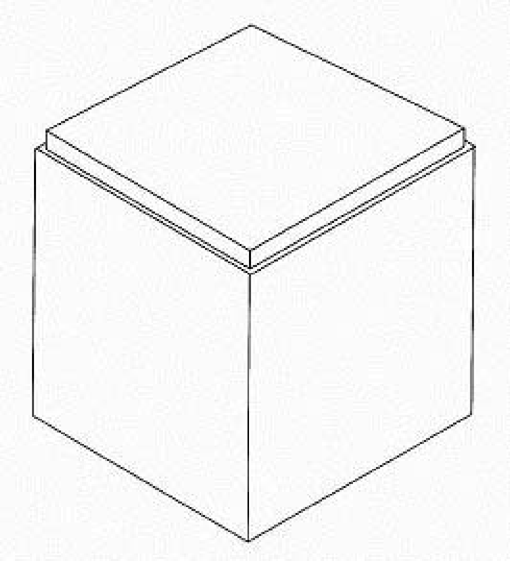


Figure 3

The entire program:

```
%
(cut rectangle with cutter comp)
g00 g90 g40 x1 y0 z1
z-.140
y.090
g41 d1 x.5
g01 x0 f10
x-1.910
y1.910
x-.090
y.100 z-.150
x-1.900
y1.900
x-.100
y.100
x-.200 y-.200
g00 g40 z1
x1 y0 z1
%
```

Nothing worth commenting about in that program except that my final move before canceling cutter comp was to eliminate a burr that would have formed if I had gone straight out, and you should be able to figure rest it out yourself. What you can do is set the value of d1 at a very small diameter and run the program through. Then set the d1 at the true diameter of the cutter used and you can see the shape of the part and then the cutter path followed using g41. Notice how the machine goes around sharp corners. Neat! This isn't a bad idea to use throughout these problems to help you better understand the process.

Planes that don't fly

In the CNC world there are three different planes. They come into play when we want our computer to do circular interpolation in a vertical plane. With this being the case, then either the x or y coordinate will have to be predominate with the Z-axis. We can control these with the g17 (x,y) g18 (x, z) and g19 (y, z) inputs. When we are working in these vertical planes generating arcs and circles, the z offset for the arc center point will be controlled by the k input, again designating the offset in the z direction given as an incremental value. Got that?

Here is an interesting little program I wrote:

The atomic circle program

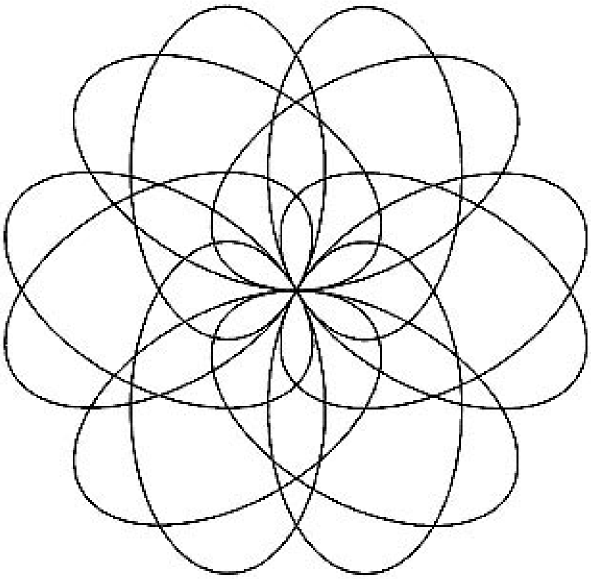


Figure 4

The entire program:

```
%
(atomic circle)
g90 g17 g40 g00 x0 y0 z0
g03 x0 y0 i0 j1 f100
g02 x0 y0 i0 j-1
g03 x0 y0 i1 j0
g03 x0 y0 i-1 j0
g18 g03 x0 z0 i0 k1
g02 x0 z0 i0 k-1
g03 x0 z0 i1 k0
g02 x0 z0 i-1 k0
g19 g03 y0 z0 j0 k1
g03 y0 z0 j0 k-1
g02 y0 z0 j1 k0
g03 y0 z0 j-1 k0
g90 g17 g40 g00 x0 y0 z0
%
```

Run it on the backplot program. Millie could also run it, but you'll have to change the f100 feed program to f12 so Millie can handle it. Be sure the slides are located in a position where the slides can be moved two inches (50mm) in every direction. Single step this program through and again be sure you are positive that you can understand every move made so well that you can explain to someone what the move is and why it is going to make it this or that way before going on. Our class is very understanding and we are all going to wait for you while you go back and catch up, but don't let it happen again.

The Great Race

Let’s go PC aircraft racing instead of R/C. First, set the tool diameter D1 to .600". Next, we’ll lay out a triangular course. Our micro delta racing aircraft have strange flying characteristics allowing them to fly forward and backward in a nose down attitude. (Sometimes it just takes a good imagination.) Racing rules will only allow the racer to go around the pylons by turning left. Cutter comp will make this easy because it will automatically calculate the tangent points needed to run this program, making this an easy program to write. By juggling the turn-in points we can shorten the course, thereby lowering the time. There’ll be more rules, but first lay out a triangular pylon racecourse. The vertical lines are the pylon marks. The start finish line is included, and the program will be designed to lay the course out at the beginning of each new race.

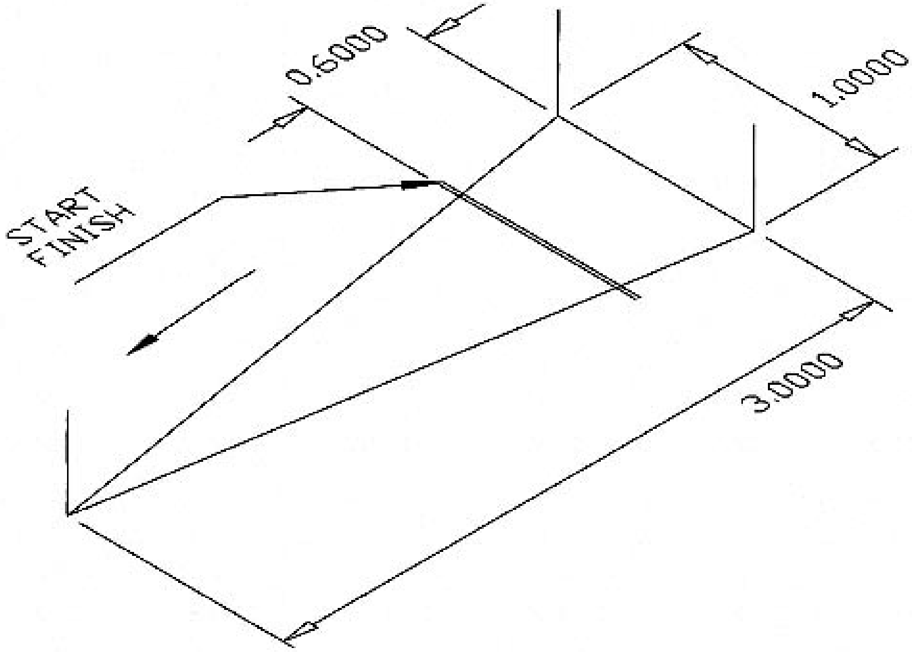


Figure 5

The entire program:

```
%
(EMC race course)
(Set tool diameter D1 to .600)
g00 g17 g40 g90 x-1 y1 z0
g01 y.5 f100
z.5
z0
x-4 y1
z.5
z0
x-1 y1.5
z.5
```

```
z0
y.5
g00 x-1.6 y.8
g01 y1.8 f100
(course complete)
(Taxi to starting line)
x-1.6 y1.3 f15
m00
```

This will pause the program while you get ready to set your stop watch. To start the race you'll have to use [Resume] button because the [Run] button will not restart the program when it is in a "paused" mode caused by the m00. Now for the race…

```
(Use resume button to start race)

x-2 f6
x-4 y1.4 z.5 f10
g03 x-4 y.6 i0 j-.4
g01 g42 d1 x-3 y.5 f20
x-2
x-1 y.5
y1.5
(lap 2)
x-4 y1 f25
x-1 y.5
y1.5
(lap 3)
x-4 y1 f30
(top speed)
x-1.123 y.578
y1.373
(lap 4)
x-3.862 y.961
x-1.123 y.578
y1.373
(lap 5)
x-3.862 y.961
x-1.123 y.578
y1.373
x-1.6
(race over)
g40 x-3.5 z1
```

```
x-4 y1.5
g03 x-4 y.5 i0 j-.5
g01 g90 x-2
g18 g02 x-2 z1 i0 k1.5
g01 g90 g17 x2
g02 x2 y-.5 i0 j-.5
g01 x-2 z0 f20
x-2.5 z.15 f15
x-3 z0 f10
x-3.2 z.1 f5
x-3.4 z0
x-3.6 z.05
x-3.7 z0
g03 x-3.7 y-.9 i0 j-.2
g01 g90 x-1 f25
g03 x-.5 y-.4 i0 j.5 f15
g01 y.5
g03 x-1 y1 i-.5 j0
g40 g90 g00 x-1 y1 z0
%
```

The same program with comments from the pilot:

```
x-2 f6 –Gaining airspeed and lift-off
x-4 y1.4 z.5 f10 –Climbing out and headed for turn 1
g03 x-4 y.6 i0 j-.4 –Large radius turn to prevent stall
g01 g42 d1 x-3 y.5 f20 –Turn on navigation aid to help in turns
x-2 –2nd point for entering g42
x-1 y.5 –Turn 2 point
y1.5 –Turn 3 point
(lap 2)
x-4 y.1 f25 –More speed and back to turn 1
x-1 y.5 –Turn 2 point
y1.5 –Turn 3 point
(lap 3)
x-4 y1 f30 –We're at top speed headed for 1
(top speed)
x-1.123 y.578 –We tighten up the course for the rest a race
```

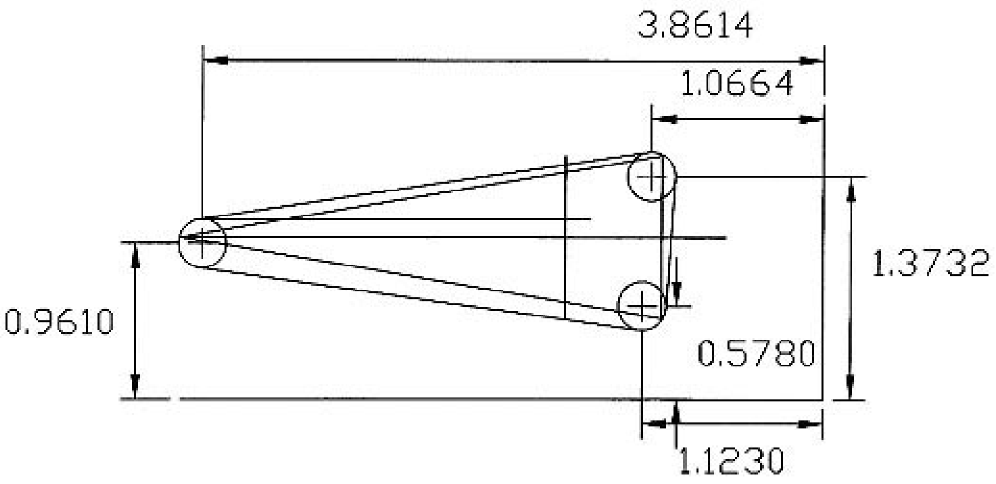


Figure 6

I cheated and used AutoCad®…

```
y1.373
(lap 4)
x-3.862 y.961
x-1.123 y.578
y1.373
(lap 5)
x-3.862 y.961
x-1.123 y.578
y1.373
x-1.6
(race over) –We won
g40 x-3.5 z1 –Turn off the navigational aid and climb out
x-4 y1.5
g03 x-4 y.5 i0 j-.5 – A nice lazy fast turn
g01 g90 x-2
g18 g02 x-2 z1 i0 k1.5 –We change planes and do a victory loop
g01 g90 g17 x2 –Enter the landing pattern back in x, y plane
g02 x2 y-.5 i0 j-.5 –Turn onto final landing approach
g01 x-2 z0 f20 –Throttle back, oops, dam it (bounce)
x-2.5 z.15 f15 –Flare it out, dummy
x-3 z0 f10 –Damn, right in front of everyone
x-3.2 z.1 f5 –I give up. I never did like tail-draggers
x3.4 z0
x-3.6 z.05 –At least I'm still alive
x-3.7 z-.01 –I think I dented their runway
z0
```

```
g03 x-3.7 y-.9 i0 j-.2 –Let’s get out of here before they remember me for that lousy landing
g01 g90 x-1 f25 –A nice fast taxi home
g03 x-.5 y-.4 i0 j.5 f15
g01 y.5
g03 x-1 y1 i-.5 j0
g40 g90 g00 x-1 y1 z0 –Home at last. I wonder if they’ll remember my lousy landing more than our win
%
```

Your homework assignment is to figure out this program that I just spent more time on than I care to admit. There isn’t anything new in it, and you have to understand all the code used. It may be a little “far out” but it is easy to follow if you know about aircraft.

Your next assignment is to see if you can lower my time from leaving the starting line until the race is over time is displayed. After lap 2 is completed, the turn-in points are the only thing you have to work with, and you can’t change the feed-speed or the feed override on the panel.

My time was an astounding 1 min 42.7 sec. Your job—lower it.

Test two

Up until now all you have done is copy my examples into your computer, and you’re probably thinking this CNC stuff really isn’t that hard after all. It’s time to bust your bubble before you get a big head and become a pain.

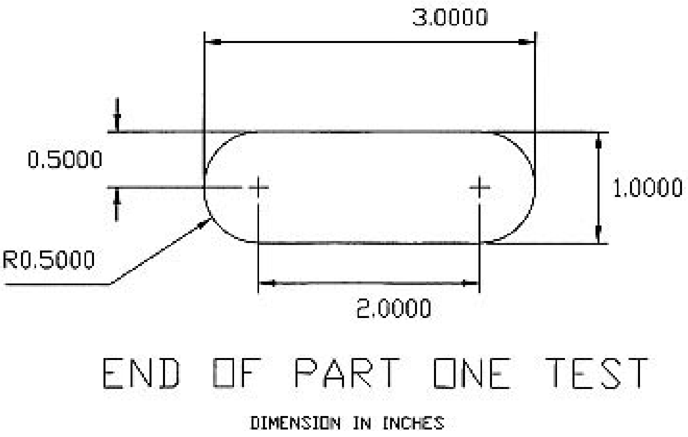


Figure 7

I want you to write a complete program on your own to profile this part. To pass the test, you must use cutter comp. I have taught you all the fundamentals to write this program, and the only suggestion I will make is to test the program a segment at a time as you write it. That’s the great advantage of having Ray Henry’s backplot program to work with.

Change a couple of dimensions and make one end square and take another test.

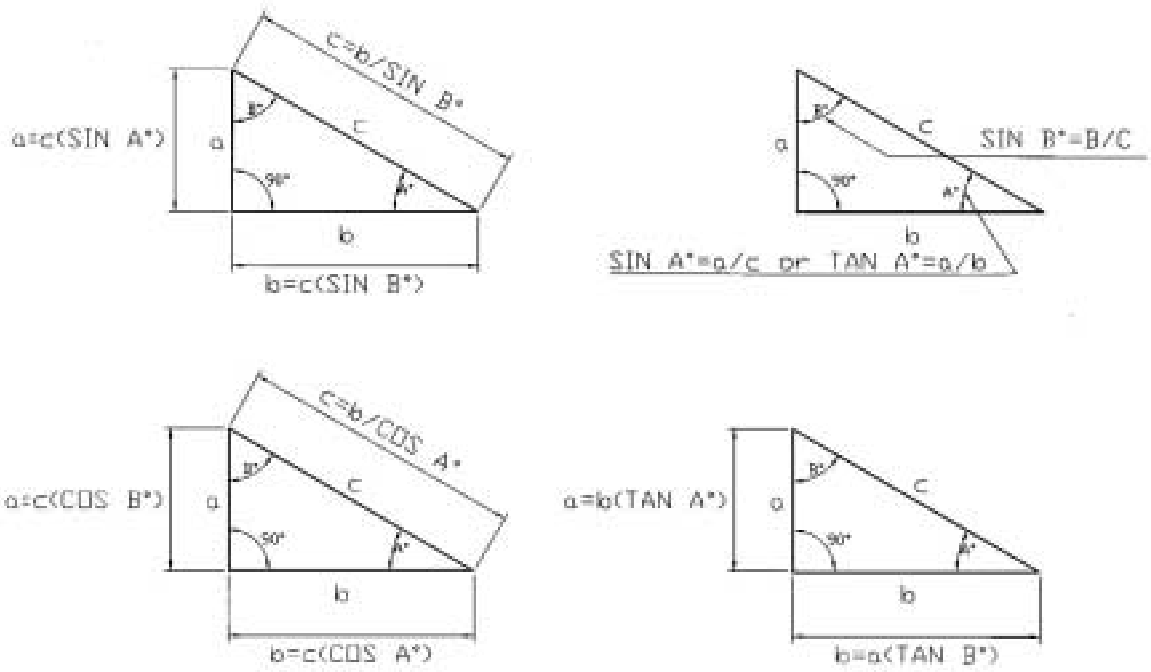
There isn't any sense in entering the next section unless you can easily write these programs. If not, go through that section again and again until it is second nature.

Back to school

We're on the move again and leaving route g41 for a while, but I'm sure we'll return. We are headed back to school for a refresher class on trigonometry, more commonly called trig. Today all you need is a $10 calculator that has trig functions or a hook-up to the Internet where they have a site that will solve your problems instantly by entering any two known sides or a side and an angle other than 90°.

http://www.pagetutor.com/trigcalc/trig.html

It just doesn't get any easier than this, or you can use the formulas below and a calculator. All you have to enter are any two known things about the right triangle you want to solve other than the 90-degree angle which is always present.



Basic Trigonometric Functions

Also: $c^2=a^2+b^2$ $c=\sqrt{(a^2+b^2)}$ $a=\sqrt{(c^2-b^2)}$ $b=\sqrt{(c^2-a^2)}$

Figure 8

When programmers solve for tangent points, they usually start off knowing the hypotenuse of the triangle they are working with, because it is the radius of the arc of which they are solving for the start and stop points.

The hardest part of this part of the course isn't solving a right triangle. It's finding the correct right triangle to solve to give you the answer to determine your tangent coordinates that will determine the start and stop points of an arc. You'll remember that I told you that these points must be given to an accuracy of four decimal places in order to work in unison with the computer. It has nothing to do with the accuracy you require, but only with what our loyal computer requires.

I dug out my old calculator to solve these problems and then checked my answers against what I had already calculated in AutoCad®. I'm going to assume you will be making the same mistakes I did, so I'll pass my thoughts on about the problems I had. When you're dealing with angles less than 10° changes are subtle and errors will not stand out. Be absolutely sure that you are using the correct equation to work with the information that you have available. All you need are two sides or a side and an angle other than 90°.

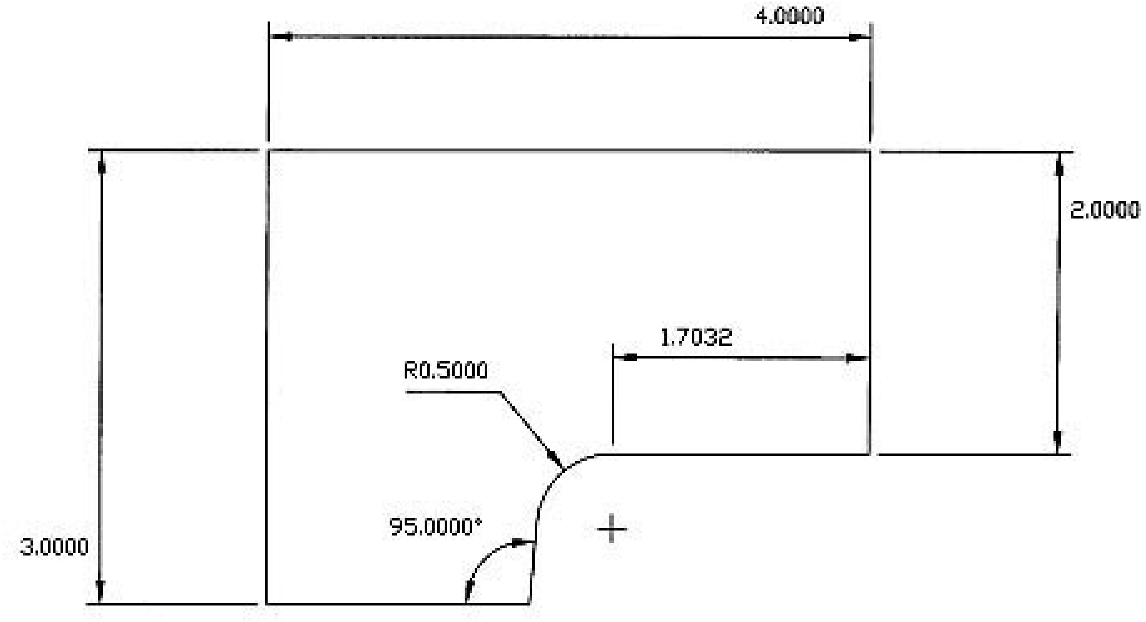


Figure 9

The first example I'm going to explain looks deceivingly simple and is typical of what you have to start with, and believe me it isn't easy, so pay attention. I started off this drawing by putting a circle in a rectangle in no particular position on the X-axis and then putting a 5° line against the tangent point of the circle. My next task was to figure out the missing points with a calculator. The only practical way to accurately figure these points out is with trig.

First we have to find and solve the missing values for the right triangles shown in Fig. 10 below. The key triangles are shown in solid black.

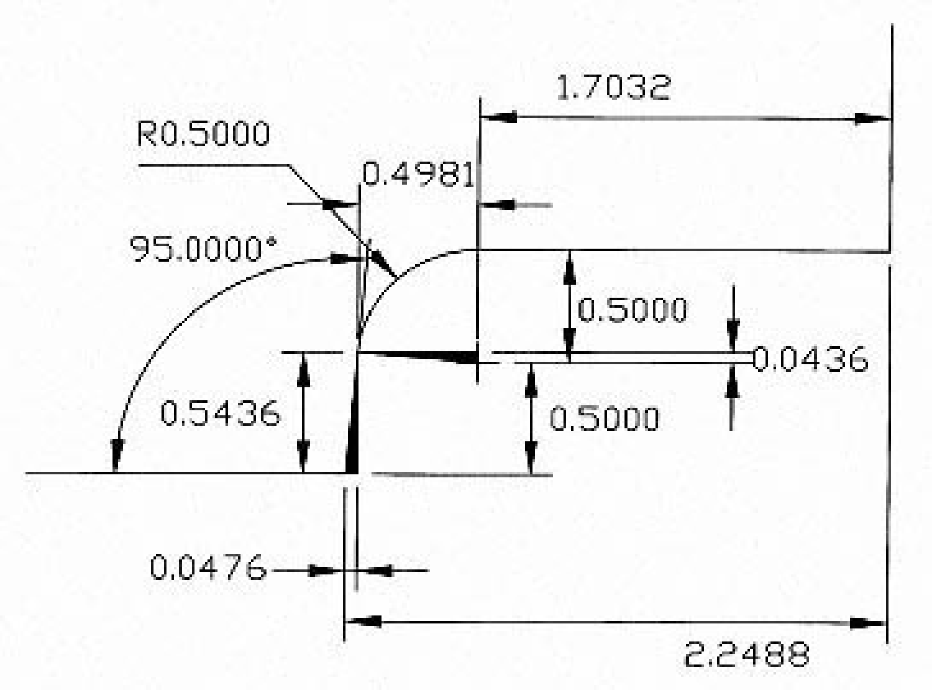


Figure 10

First, what two things do we have to solve this triangle? We have the hypotenuse because it is the radius of the arc and the 5° angle. From our tables we see that the 0.0436 was derived by multiplying the hypotenuse (0.5) times the sin 5°; and the 0.4981 was the product of 0.5 x cos 5°. This will give us our end-of-arc position.

The 0.0476 has to be calculated to give us the final X-axis point of the angle. We can easily calculate the 0.5436 because it is the sum of 0.500 and 0.0436. This gives us one side of the triangle that we know is 5°; therefore the dimension needed is the sin 5° x 0.5436. Adding up these 3 dimensions we end up with the 2.2488 X-axis dimension. We now have all the numbers we need to generate the code to produce this part.

Take one more look at the complete drawing before starting to write the program. The main thing I want you to learn is how important it is to solve for the correct triangle to get the answers you need, and the many times you have to solve more than a single triangle to solve what may seem as a straightforward problem.

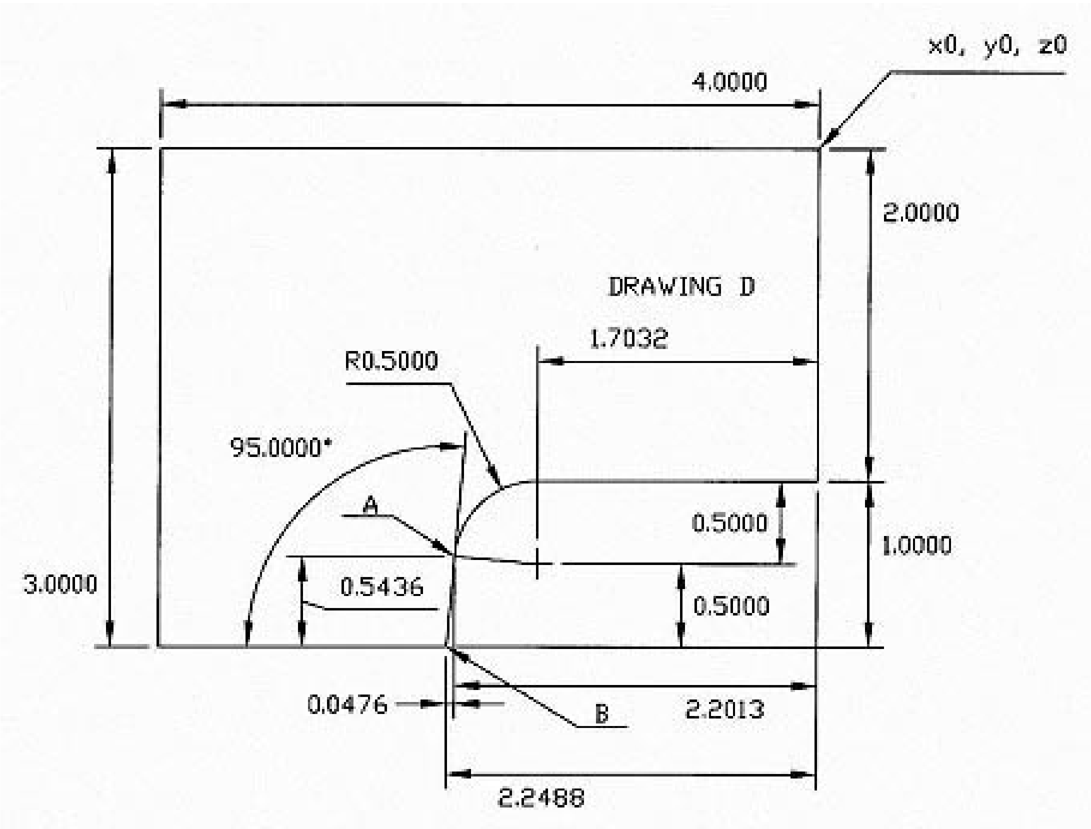


Figure 11

This program is just an exercise, and I'm not going to write any Z-axis code. The 0 starting point is located in the upper right-hand corner. Of course, we'll also use cutter comp to generate this shape. Now, read the code I wrote and examine Fig. 11 again and visualize where the starting and stopping points are in your mind before you run the program in backplot. This is what we are striving for, because a real professional programmer can read code like a letter.

The entire program:

```
%
(rectangle, arc and tangent)
g00 g40 g90 x0 y1 z0
g41 d1 x0 y.5
g01 y0 f16
y-2.000
x-1.7032
g03 x-2.2013 y-2.4564 i0 j-.5
g01 x-2.2488 y-3.000
x-4.000
y0
x0
```

```
x.5
g00 g40 g90 x0 y1 z0
%
```

With comments

```
 %
g00 g40 g90 x0 y1 z0
g41 d1 x0 y.5
g01 y0 f16
```

Last time, I started at y1 position so I could enter into the cutter comp mode in two moves and coming in from the correct direction (remember - gear down-touch down).

```
y-2.000
x-1.7032
```

g03 x-2.2013 y-2.4564 i0 j-.5 –Last time, The x and y positions where the arc ends.

```
g01 x-2.2488 y-3.000
x-4.000
y0
x0
```

x.5 –Exiting g41 mode in correct direction.

```
g00 g40 g90 x0 y1 z0
%
```

Work out a couple of similar problems on your own while I get ahead of my students again. Boy, you guys are smarter than I thought you were.

The plot is about to thicken with your next problem that I called the pulley program.

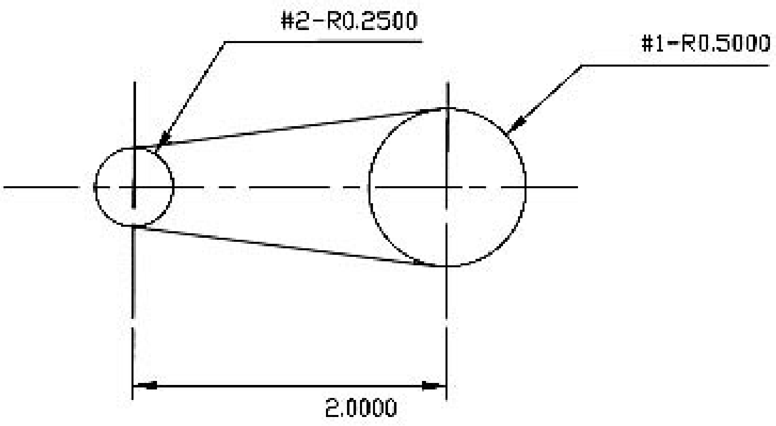


Figure 12

In this problem all we know is the radii of both circles and the distance between them, yet we have to arrive at the tangent points.

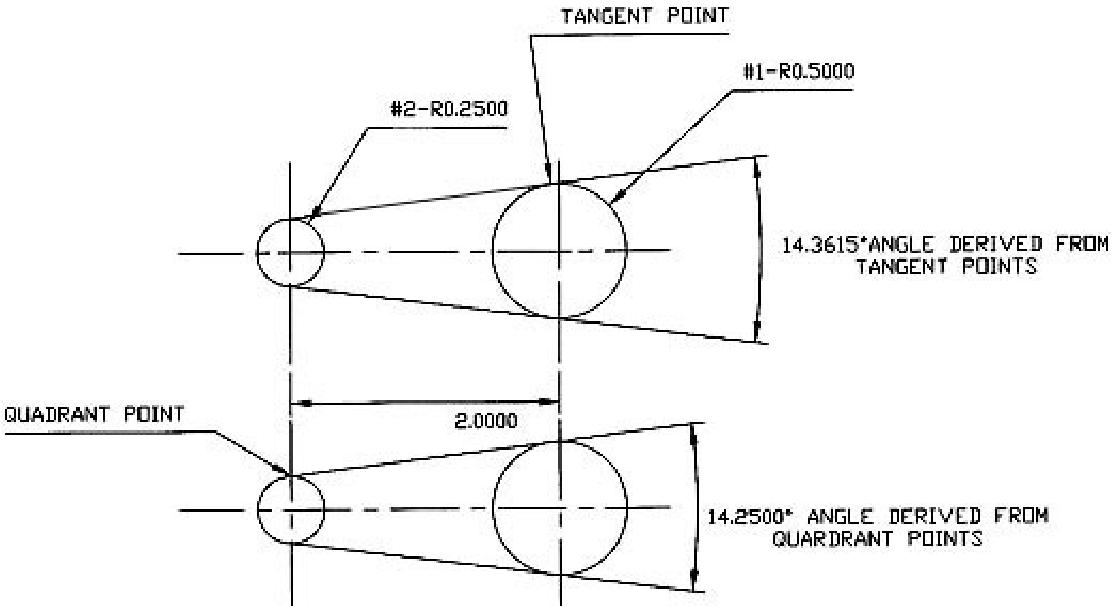

TANGENT POINT
#1-R0.5000
#2-R0.2500
14.3615°ANGLE DERIVED FROM TANGENT POINTS
QUADRANT POINT
2.0000
14.2500° ANGLE DERIVED FROM QUARDRANT POINTS

Figure 13

Here is the real problem. The seemingly obvious way of solving this problem would be to solve the angle that can be generated with the differences between the radii and the distances between centers and then use this information to come up the needed points. Wrong. Compare the two angles in Fig. 14. Although the error isn't that great for laying out sheet metal work, it's a gigantic error for a computer to swallow, and you have to be accurate to four places.

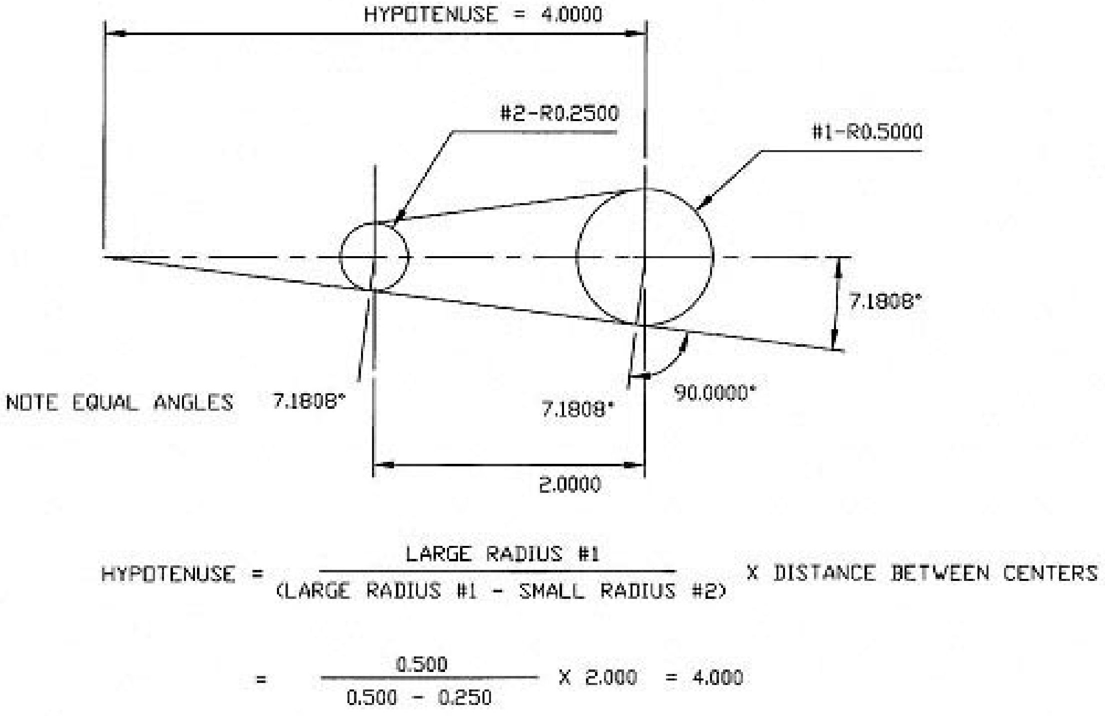

HYPOTENUSE = 4.0000
#2-R0.2500
#1-R0.5000
7.1808°
NOTE EQUAL ANGLES
7.1808°
7.1808°
90.0000°
2.0000
HYPOTENUSE = LARGE RADIUS #1 / (LARGE RADIUS #1 - SMALL RADIUS #2) X DISTANCE BETWEEN CENTERS
= 0.500 / 0.500 - 0.250 X 2.000 = 4.000

Figure 14

Believe me, I studied this problem for a long time before I was able to solve it. I also came up with unusually simple way to arrive at the second side of the right triangle so the problem was solvable. I don't want to blow my own horn, but it is extremely rare that an individual like myself, who hasn't studied high math, will ever come up with a complex mathematical rule on his own that works, whether mathematicians know about it or not.

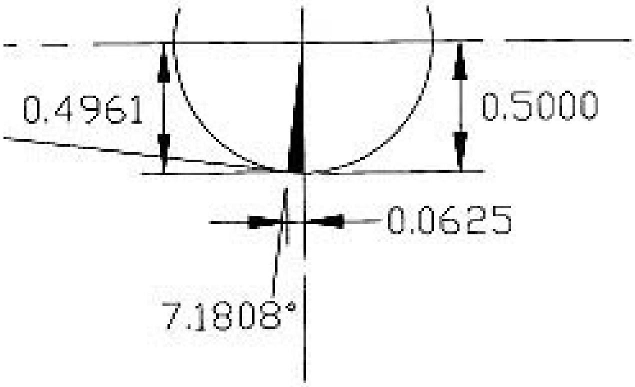


Figure 15

Now we can calculate the angle because the sin B = hypotenuse ÷ side b. With this angle we now know the remaining missing tangent points can be solved with your calculator. Note that once we know that angle, it should be noted the same angle falls in two other places as shown in Fig. 15. This allows us to solve the remainder of the problem in Figure 16 in the same manner as we did in Fig. 15.

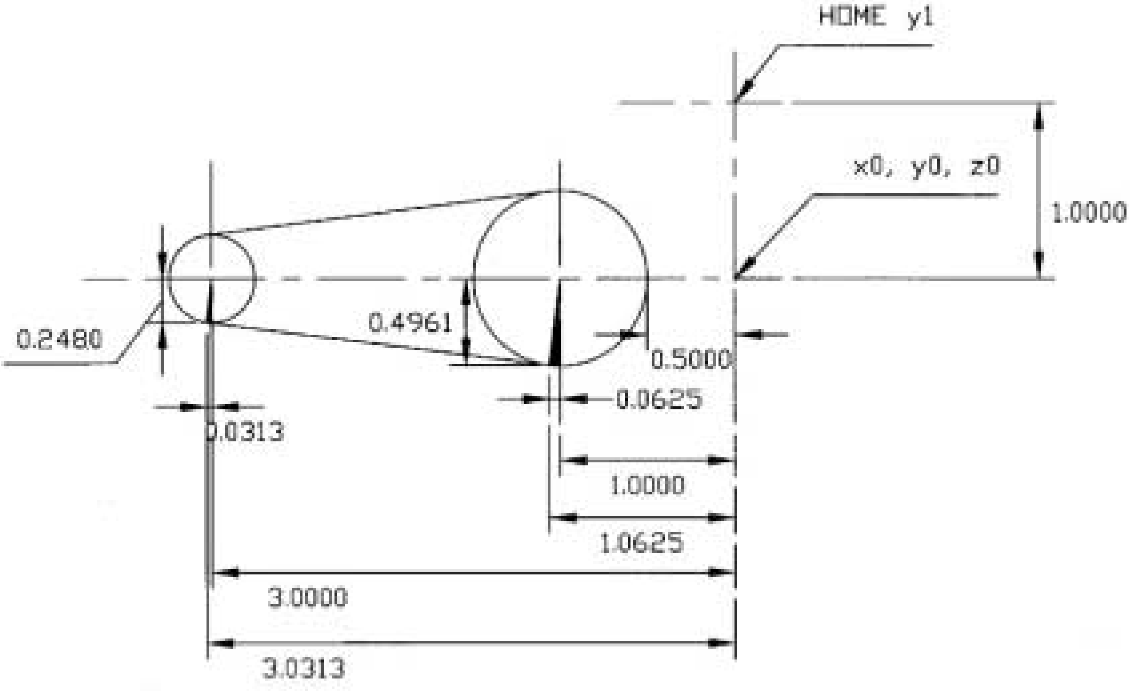


Figure 16

Let's write the code for this simple yet not-so-simple shape. One thing I must state again is that I'm having a terrible time with careless errors. I think the problem is that I'm more worried about how I'm going to explain each problem than I am in entering the correct numbers. Don't fall into my trap. You don't have to explain to me how you did it, so learn from my mistakes. Become a hermit while you are studying so you don't make careless errors.

The entire program:

```
%
(pulley program)
g00 g40 g90 x0 y1 z0
x-.500
g41 d1 y.500
g01 y0 f12
g02 x-1.0625 y-.4961 i-.500 j0
g01 x-3.0313 y-.2480
g02 x-3.0313 y.2480 i.0313 j.2480
g01 x-1.0625 y.4961
g02 x-.500  y0 i.0625 j-.4961
g00 g40 x0
x0 y1 z0
%
```

I believe that you should be able to follow the pulley program without comments now that you have calculated the tangent points.

Our next problem is linking to arcs together, and in this case the arcs are going in opposite directions. What I want you to notice in this particular program is the fact that you enter cutting the second 0.300 radius from the opposite direction as you did in the same radius on the opposite side. This changes the i and j values going in and coming out.

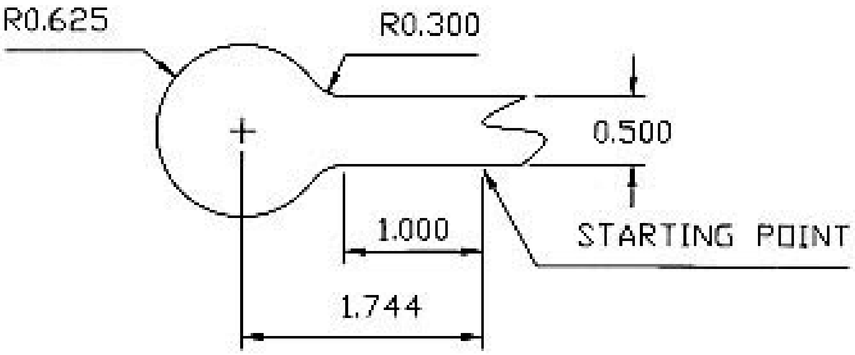


Figure 17

When you have to solve a problem like this, the first thing you have to do is determine what you have to work with. In this case, we know the two radii of the arc and the width of the straight section leading to it. Studying the drawing you will find that with the information known there is only one position where we can begin to solve for the missing dimensions to write our program.

The radii will give us the hypotenuse of the triangle we need to form, and we can determine the second side of the triangle by adding together the radius of the first arc and the distance the second arc's center is from the edge. Now we can calculate not only the starting point of the first arc ($b=\sqrt{c^2 - a^2} = 0.7437$), but also the angle of that triangle ($\sin A^\circ = a \div c = 36.4837^\circ$). With the angle now known we can calculate the x, y ending points of the arc, and we have already calculated the i and j values for that radius.

The ending point of our first arc is also the starting point of our second arc; therefore, because the part is symmetrical, the ending point of the second arc is located the same distance from the center point of that arc. To start the third arc new i and j values must be determined, because we are approaching this arc from the opposite direction even though this arc is identical to arc one.

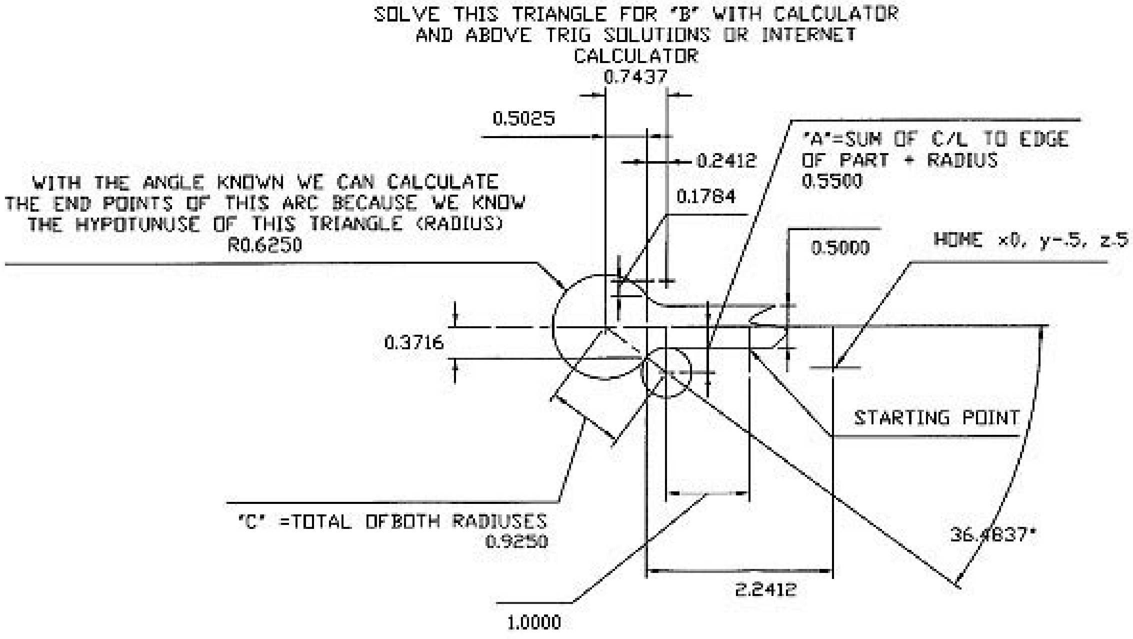

SOLVE THIS TRIANGLE FOR "B" WITH CALCULATOR AND ABOVE TRIG SOLUTIONS OR INTERNET CALCULATOR
0.7437
0.5025
0.2412
0.1784
"A"=SUM OF C/L TO EDGE OF PART + RADIUS
0.5500
WITH THE ANGLE KNOWN WE CAN CALCULATE THE END POINTS OF THIS ARC BECAUSE WE KNOW THE HYPOTUNUSE OF THIS TRIANGLE (RADIUS)
R0.6250
0.5000
HOME x0, y-.5, z.5
0.3716
STARTING POINT
"C" =TOTAL OFBOTH RADIUSES
0.9250
36.4837°
2.2412
1.0000

Figure 18

Don't just look at my answers and assume they are right. In the real world you don't solve problems you know the answers for. I'm writing this stuff for you to teach you how to do it. I already know. Be original and change a few basic dimensions and see if you can be confident that your answers are correct. That's what it is all about. Don't find out that you don't know how to do it by making bad parts. We are machinists who don't have erases to work with. Don't become one of those "Oh shit" machinists who can only have their mind work at full potential after he screwed up! Another free lecture by J. Martin.

Now let's write the code for this beauty. Also, I added a couple of Z-axis movements that might be used in the real world.

```
%
(rod-end)
g00 g40 g90 x0 y0 z0
g41 d1 x-.75 y-.25
g01 x-1 y-.25 f25
z-.5
x-2
g03 x-2.2412 y-.3716 i0 j-.3
g02 x-2.2412 y.3716 i-.5025 j.3716
g03 x-2 y.25 i.2412 j.1784
g01 x-1
x-.75
z0
g00 g40 x0 y0 z0
%
```

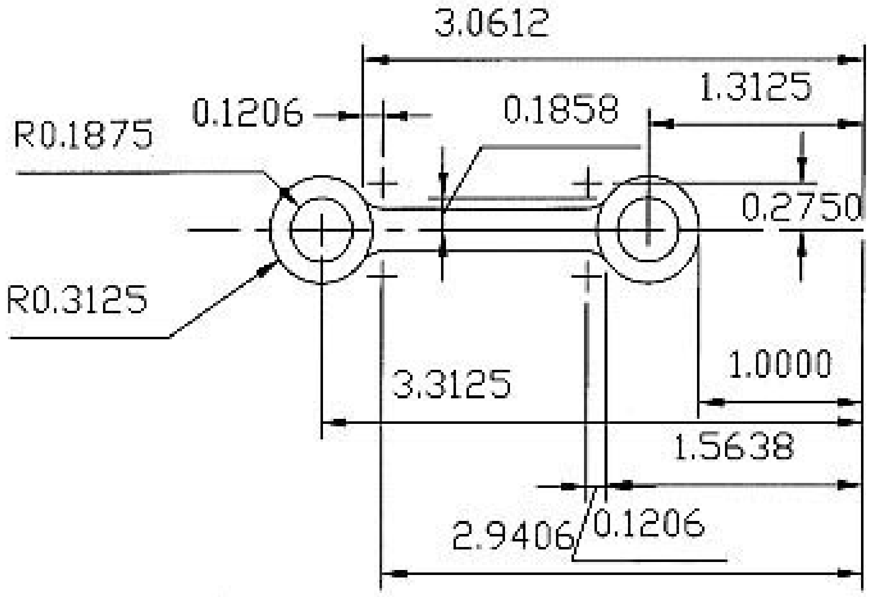


Figure 19

The rod program adds a useful new tool to our collection. The "r" letter designates "radius" when it's used in the same block as the circle command g02 or g03. This simplifies code writing because it eliminates the need to calculate the i and j points that are always changing as seen in our last example; however, the start and end points of an arc must be calculated to the same degree of accuracy. I felt obligated to teach you the hard way first because many older CNC systems don't have the advantage of the r command, and that's the type of systems that many beginners will end up running. You will see in the rod program just how much tedious code writing it eliminates.

```
%
(Connecting rod)
g00 g90 g40 x0 y0 z0
g01 x-1 y.5
g41 d1 y.1
g01 y0 f15
g02 x-1.5638 y-.1858 r.3125
g03 x-1.6844 y-.125 r.15
g01 x-2.9406
g03 x-3.0612 y-.1858 r.15
g02 x-3.0612 y.1858 r-.3125
g03 x-2.9406 y.125 r.15
g01 x-1.6844
g03 x-1.5638 y.1858 r.15
g02 x-1 y0 r.3125
g01 y-.1
g00 g40 x-.5
x0 y0 z0
%
```

This concludes your basic course on CNC g-code programming. You'll know if you passed this course as soon as you attempt to write your own code to make a particular part that you need. You now have the knowledge to produce a very complex part. The next stage of codes to learn will only simplify the writing of the code you now know how to write. I believe that working before you move on you should build up your confidence by producing some parts that you can use before traveling on.

Using the optional 4th axis

The Sherline driver board was designed to accommodate the fourth (A) axis; therefore, all that is needed is P/N 8730 rotary table that includes both the proper stepper motor and the stepper motor mount. It sells for $370 (6/03). The A-axis is programmed in degrees. A minus entry will rotate the table in a counter-clockwise direction. The A-axis can move in unison with the X-, Y- and Z-axes. Use slow feed rates to avoid a situation where the A-axis cannot keep up because of the 72 to 1 worm ratio of the rotary table. There isn't any EMC software available at this time to write the complex code associated with engraving or machining on a round surface; however, I believe the type of parts that will be made by Sherline machinist are straightforward and can be easily programmed. Engrave parts such as jewelry will need special programs. I'm sure I'll be writing more on this subject when I find out just what type of work you want to run with this device.

The Sherline 8730 CNC Rotary Table with stepper motor can be plugged directly into the A-Axis cord pre-wired into your computer to operate as a 4th axis.

Conclusion

You're ready to take off on your own

The next stop on the EMC highway is to visit Linux-CNC. The road to this town requires a hookup to the internet to travel on. If you're not connected to the internet you're missing one of the great advancements of civilization. It's like having all the great libraries of the world on your personal bookshelf. I have taught you the basic language of g-code programming, and much more is available on the http://www.linuxCNC.org/ site. A couple of members from this group have written some programs for simplifying code-writing that I'm sure you'll find interesting. They have a users group that can be of great help in solving code writing problems.

If you're like me you will find a lot of satisfaction with this new ability to write a program and then watch your faithful robot make parts for you at the push of a button. Though the parts you make initially on this machine may be small, what you have learned is not. When you master this, you will have gained the ability to make parts in shapes and in numbers that cannot be matched by the world's best machinists working by hand. If you want, you can apply what you have learned to larger machines and larger parts later on. Whether you took on this challenge just for the fun of it or as a way to solve an immediate practical need for small precision parts, you are on your way to joining the leading edge of today's modern machinists.

If you found my instructions useful and you are not reading them to operate a Sherline machine may I suggest a way to show your appreciation and help a good cause would be to donate a few dollars to my foundation that is dedicated to getting great craftsmen the respect they deserve in this world. Check out http://sherline.com/jmfound.htm and be sure to visit the Joe Martin Foundation's Internet Craftsmanship Museum at http://www.craftsmanshipmuseum.com/ where the marvelous work of these great craftsmen can be viewed and enjoyed 24 hours a day.

Definitions and Codes

The word definitions listed below were copied from the Linux CNC website. It's where I gathered the information to write part two of these instructions. It's worth reading and using to go to the next step in programming, but for the time being stick with just my instructions. I never intended to write a complete programming manual and was only interested in writing enough instructions to get you up and running in the marvelous world that we call CNC.

http://linuxCNC.org/handbook/gcode/g-code.html#anchor259197

I have highlighted in **red bold face** the words we'll be working with when using a Sherline machine, and, although many of these words we will not directly use, I wanted you to be aware of the codes used in industry. A CNC program "word" is defined as an acceptable letter followed by a real value.

Table 1
Words acceptable to the EMC interpreter

D	tool radius compensation number	M	miscellaneous function (see below)
F	feedrate	N	line number
G	general function (see below)	P	dwell time with G4 and canned cycles
H	tool length offset	Q	key used with G10 Q feed increment in G83 canned cycle
I	X-axis offset for arcs X offset in G87 canned cycle	R	R arc radius, canned cycle plane
J	Y-axis offset for arcs and Y offset in G87 canned cycle	S	S spindle speed
K	K Z-axis offset for arcs and Z offset in G87 canned cycle	T	T tool selection
L	L number of repetitions in canned cycles and key used with G10	X	X-axis of machine
		Y	Y-axis of machine
		Z	Z-axis of machine

Miscellaneous words

M words are used to control many of the I/O functions of a machine. M words can start the spindle and turn on mist or flood coolant. M words also signal the end of a program or a stop within a program.

Table 2—M Word List

M0 program stop	M8 flood coolant on
M1 optional program stop	M9 mist and flood coolant off
M2 program end	M26 enable automatic b-axis clamping
M3 turn spindle clockwise	M27 disable automatic b-axis clamping
M4 turn spindle counterclockwise	M30 program end, pallet shuttle, and reset
M5 stop spindle turning	M48 enable speed and feed overrides
M6 tool change	M49 disable speed and feed overrides
M7 mist coolant on	M60 pallet shuttle and program stop

Preparatory Words

Some g words alter the state of the machine so that it changes from cutting straight lines to cutting arcs. Other g words cause the interpretation of numbers as millimeters rather than inches. While still others set or remove tool length or diameter offsets. Most of the g words tend to be related to motion or sets of motions. Table 3 lists the currently available g words.

Table 3—G-code List

g0 rapid positioning	g58 use preset work coordinate system 5
g1 linear interpolation	g59 use preset work coordinate system 6
g2 circular/helical interpolation (clockwise)	g59.1 use preset work coordinate system 7
g3 circular/helical interpolation (counterclockwise)	g59.2 use preset work coordinate system 8
g4 dwell	g59.3 use preset work coordinate system 9
g10 coordinate system origin setting	g80 cancel motion mode (includes canned)
g17 xy plane selection	g81 drilling canned cycle
g18 xz plane selection	g82 drilling with dwell canned cycle
g19 yz plane selection	g83 chip-breaking drilling canned cycle
g20 inch system selection	g84 right hand tapping canned cycle
g21 millimeter system selection	g85 boring, no dwell, feed out canned cycle
g40 cancel cutter diameter compensation	g86 boring, spindle stop, rapid out canned
g41 start cutter diameter comp. left	g87 back boring canned cycle
g42 start cutter diameter comp. right	g88 boring, spindle stop, manual out canned
g43 tool length offset (plus)	g89 boring, dwell, feed out canned cycle
g49 cancel tool length offset	**g90 absolute distance mode**
g53 motion in machine coordinate system	**g91 incremental distance mode**
g54 use preset work coordinate system 1	g92 offset coordinate systems
g55 use preset work coordinate system 2	g92.2 cancel offset coordinate systems
g56 use preset work coordinate system 3	g93 inverse time feed mode
g57 use preset work coordinate system 4	g94 feed per minute mode
	g98 initial level return in canned cycles

Tool diameter compensation (g40, g41, g42) and tool length compensation (g43, g49) are covered

Table 4—G- and M-Code Modal Groups
Group 1 = {g0, g1, g2, g3, g80, g81, g82, g83, g84, g85, g86, g87, g88, g89} —motion
Group 2 = {g17, g18, g19} — plane selection
Group 3 = {g90, g91} - distance mode
Group 5 = {g93, g94} - spindle speed mode
Group 6 = {g20, g21} - units
Group 7 = {g40, g41, g42} - cutter diameter compensation
Group 8 = {g43, g49} - tool length offset
Group 10 = {g98, g99} - return mode in canned cycles
Group12 = {g54, g55, g56, g57, g58, g59, g59.1, g59.2, g59.3} coordinate system selection
Group 2 = {m26, m27} - axis clamping
Group 4 = {m0, m1, m2, m30, m60} - stopping
Group 6 = {m6} - tool change
Group 7 = {m3, m4, m5} - spindle turning
Group 8 = {m7, m8, m9} - coolant
Group 9 = {m48, m49} - feed and speed override bypass

There is some question about the reasons why some codes are included in the modal group that surrounds them. But most of the modal groupings make sense in that only one state can be active at a time. *Note: It isn't necessary to try and remember the words I highlighted. You'll learn these as we go.*

Installation or Re-installation Instructions for Linux

If for some reason your computer crashed (probably a loss of power) when it was writing to the hard disk, the computer may require re-installation of the basic operating system or programs. This is relatively easy to accomplish; however, the data that was stored on the hard drive could be permanently lost. This is why you should always save a copy of your programs and documents on a floppy disk, for they are the most valuable information stored in your computer.

Insert the CD: First you will need to boot off of the CD. This should work by default. Just place the Linux BDI CD (Disk 1) included with your system into the CD-ROM drive and turn on the computer. The computer will boot off of the CD and start the EMC installer.* It should scroll some text past the screen and then come up with a text based color screen that says "Welcome to the BDI Installer." Here you need to press the [Enter] key.

**NOTE: If you don't insert the CD into the drive and close the tray quickly enough, it may not boot. It might be necessary to turn the computer off once the CD is installed in the tray and then turn the computer back on. This will cause to computer to boot up from the CD.*

Language: Next you will see the graphical portion of the installer, and it will ask you to select a language. It defaults to English, but if you happen to be using another language select it and press [NEXT].

Keyboard and Mouse selection: You should be using the keyboard and mouse that came with your system. If you are, click [NEXT], otherwise select the appropriate mouse and keyboard from the list and click [NEXT].

Click [NEXT] again on the screen featuring the Linux mascot, “Chips.”

Installation options: The next option is what type of installation you want. Select the “EMC Developers System” and click [NEXT].

Depending on how badly damaged your system is, the next screen might say that the hard drive is corrupted. If so, click the “Initialize” option. A few other options may come up at this point, asking if you want to destroy the existing partitions, etc. You want to choose “YES.”

The next option will give you two choices: “Remove Data,” and “Manually Partition.” You want to do this automatically, so choose “Remove Data” and click [NEXT].

Now it will ask you for the time zone. Move the mouse over your part of the world until the arrow points to your city, or a city with the same time zone as the one you are using. Then click [NEXT].

You will now need to setup account information. If you want a default configuration just click [NEXT] without giving any input.

Monitor Selection: The next part will be different for each user, but the monitor selection is crucial. If you select a monitor type that is not compatible with the one you are using, the graphical interface will try to project an image that your monitor will not be able to handle. Therefore, it is imperative that you select a monitor that is as close to the one you are using as possible. At Sherline for the default configuration we use “Generic Non-Interlaced SVGA monitor.” This will work with most monitors. Once you have selected the appropriate monitor click [NEXT].

Configuration: The following screen will be the X configuration screen. Under “memory selection” click on “16384k,” and also select “Use graphical login.” Then click [NEXT].

Click [NEXT] one more time.

Now it is going to install everything, which usually takes about 5 minutes.

Once the install is done it will ask you to create a boot disk. You want to click “skip boot disk creation,” and then click [NEXT], and finally [EXIT]. The CD-ROM should eject. When it does, remove the CD and close the CD tray so that the installer doesn’t start again.

Installing the Linux OS with the EMC on a different computer

The first thing you must understand is that the Linux operating system is not compatible with the MS Windows® based operating system; however, it will read many files that may have been created in other operating systems. Be absolutely sure that when you insert the Sherline EMC disk into your computer that you have transferred programs and data to another computer that you want to save. Your hard disk will be completely overwritten with the Linux operating system. Installation is simply putting the CD into your computer and rebooting on most computers. (Boot means turning the computer off and on) If the computer doesn't want to install the program in this manner go to EMC Install section and "read installing from a floppy."

If you didn't purchase a Sherline system that included a computer, I'd like to suggest purchasing a used computer and dedicating it to your CNC setup. Another choice could be upgrading your present computer system and using your existing computer for this CNC project. The requirements are simple: For 400 MHz or better use the "inch-freq" version that I prefer; however, if you have an old 233 MHz around it will run with the "inch_step" version. A computer must have a hard drive, CD and floppy. I like laptops because they don't take up to much room on your workbench. Use whatever you're comfortable with. Don't waste a lot of money buying a computer that's too good for the job. Spend the savings on Sherline stuff.

One more time. Installing the Linux EMC® program is going to eliminate all programs and data from your computer, and you'll be starting over with the Linux operating system. You'll also be loading many other useful programs that are free and perfectly legal to own and operate. I'm not going to get into that end of it, but there is much information about these programs on their website at www.linux.org.

It is also possible to partition a hard disk in such a way that you can run both operating systems, but it would be far beyond the scope of my instructions to instruct you how to do so. A smart computer hacker friend could help you accomplish this. Things change so fast in the computer world whatever I wrote on the subject would be wrong next month.

Linux Installation on a different computer

The following installation section is taken directly from the www.linuxCNC.org site. It is reprinted here in case some of it might prove to be helpful if you are installing on a computer other than the one provided with your Sherline system. The most up-to-date version of this section can always be found at the web address listed above.

Installing EMC from BDI 2.xx

Installation can be a daunting task to people new to Linux. The hardest part is getting the Real Time Linux patch up and running. After that, installing EMC is pretty easy. With that said, most people have found that the easiest way to install Real Time Linux and EMC is to go to Tim Goldstein 's web page and use his install script for EMC on the Linux 2.0.36 release. Or you can find or order a copy of the BDI install CD-ROM that was produced by Paul Corner. This disk installs Linux 2.2.18 and Real Time Linux 3.0 and comes with a recent EMC release.

2.1 Setting up to install

A boot is what happens when you start your PC. Normally, when an operating system (OS) has been installed, your PC will boot without any effort on your part. The trick here is that you need to get the PC to boot into a temporary Linux system in order to install the

BDI and the EMC. Let’s run a quick experiment so that we know how your PC wants to boot. Turn off the PC. Place the BDI disk into the CD-ROM and turn it back on. If you see “Welcome to the Brain Dead Install for EMC” at the top of the screen after a bit, you’re there and you should skip down to install questions. If you see the old OS you need to make an adjustment to the BIOS. The BIOS is the first little program that wakes up when a PC starts. We need you to change this little program so that the PC wakes up and reads the CD-ROM before it reads any other input device.

2.1.1 BIOS setting

The BIOS is the first thing that works when a PC starts. In most cases, you can get into bios configuration screens by pressing <Delete>, <F1>, or occasionally <F2>. These F keys are the function keys along the top of a standard keyboard

Once you get into the configuration screen, the boot sequence is usually stored on the second page of stuff. It might look something like figure xxxxx. If the CDROM is listed, make it the first boot device. In most cases you can use hd0 for the second without problems. If the PC still will not boot from the CD but keeps trying to run its old OS, you will need to make the CD-ROM the only boot device listed. If this works you can skip the boot floppy section and read the install.

Some PC's are not set up to allow you to do this. If that is the case with yours, you will need to build a boot floppy as outlined in the section below.

2.1.2 Boot from Floppy

Most of the newest PCs, the ones that do not have a floppy drive will boot from CD-ROM. Many older CD devices did not allow you to boot from them. If you have a floppy drive, you can boot from it if you make a disk first. You will need to set the bios to boot from floppy using the same procedure indicated above.

You will also need to make a boot floppy. For this you will need a good 1.44 Meg floppy and a working computer that you can copy a file from the BDI CD-ROM to that floppy.

Making a boot floppy using MS-Windows or DOS.

Your BDI CD has both the file that you need on the floppy and the program that is required to write that file to a good disk. You need to use a formatted disk for this. It does not need to be empty but the old stuff will be overwritten with the new file.

There are two directories that you need to work in to make a floppy. The file you need to copy is in the “images” directory. It's called *images/boot.img*. This image takes a full floppy (1.4 Meg). This is a special file because it includes everything that you need to start up a computer. The system is there, the OS is there, and there are hardware drivers that will read your keyboard, send text to your display and read your CD-ROM drive.

The second directory is *dosutils*. You can't just drag and drop boot.img onto a floppy icon and have it work. You need to write boot.img to your floppy using a little program named dosutils/rawrite.exe. The exact process will depend upon the version of MS-Windows that you are running. MS-Windows 2000 let me use the file explorer to find the correct drive name (E:) for the CD-ROM and (A:) for the floppy. It also let me start rawrite.exe by navigating to E:dosutils and double clicking on that executable.

Rawrite said, “Enter disk image source file name:”

I answered, “E:imagesboot.img”

Rawrite said, “enter target drive:”

I answered, “A:”

Rawrite said, “Please insert a formatted diskette into drive A: and press [Enter]:”

I did that and after a minute or two the terminal window went away and I had a boot image on a floppy.

With some of the older MS-OS’s you may need to start a dos terminal from the start menu rather than simply clicking on rawrite.exe. Under the start menu you should find an MSDOS line. Once you have the dos window up you can enter text. The command line is: (your CD-ROM drive letter):dosutilsrawrite.exe

Rawrite will prompt you just as it did in the test above. When you start up the install with a floppy, you should have the BDI CD disk in the CD-ROM drive.

2.2 Install Questions

When you see the “welcome to the BDI” press [Enter]. Nothing to fear here. The install routines used by the BDI are based on some of the best install programs available today. These programs recognize thousands of devices, and, for the most part, ask only that you confirm its choices. Many of these choices can be changed after the install if you find that they do not work just right. There are a few critical parts, so read through this section while you answer each of these questions

As the BDI begins to install you should see a full screen picture of a mini-mill. Seeing this tells you that install has recognized at least a few things about your display driver and will run using graphics rather than text. If this image is not clear or there are several of them you probably have a multisync monitor, and you will need to request a text install. In this case you can change that setting later. Start over by shutting off the computer and restarting it. When you get to the boot prompt type, “linux text” and press [Enter]. You will be able to move around the screen with the [Tab] key. The up/down arrow keys will move the highlight through a list. Some lists, usually those with [ ] use the spacebar to select your choice from a list. The questions used in the text install are about the same as the graphic one.

2.2.1 Language Selection

The default is English. Select your language and press next. The next button is the key to moving through the install menus.

2.2.2 Keyboard Configuration

There are three variables set here. Model usually defaults to generic 101 keys or similar. Unless you have a special keyboard the default will do the job for you. The layout of a keyboard is different for laptops and for different countries. It may point you the correct one, but you might just look through the list for one that is closer to what you are using. I'd leave the dead keys part alone.

2.2.3 Mouse

If this is a PS-2 mouse—the one with the little round connector that plugs directly into the back of the computer—you should be able to get by with either a 2 or 3 button generic definition. A serial mouse may not be recognized during the graphical install, and you may want to try text install instead. If you know which serial port the mouse is connected into, you can set it on this screen. Mouse configuration is not an exact science. You should expect to experiment for best performance after you complete the install. (See setup section.)

2.2.4 Graphic

Penguins are sort of a Linux trademark. This little guy, named Chips, is the EMC's mascot. When you're through admiring the artwork, press [Enter].

2.2.5 Install Type

If you want to run a machine and carve that hunk of material into your dream part, select the BDI EMC Install. This will take less than 500 Meg of hard drive space and should install if a few minutes. If there is a chance that you will want to change things in the program, once you've gotten comfortable with it, select EMC Developer's system. This install will take about 1.3 Gig. It should complete in less than 15 minutes.

2.2.6 Install Window

Default here is “Remove data.” To me this is the way to go. You set aside that old PC for this job, let it do it. It is possible to manually partition a drive and dual boot with other OS's but I'm not even going to go into it here. For more information on dual boot, consult your favorite internet search engine.

2.2.7 Time Zone

Most computers run on local time. Just pick yours from the list. The fancy stuff's under the UTC offset tab.

2.2.8 Account Configuration

Linux works on accounts. It needs to know who is doing what with its files. You will at least need a superuser account. This is the one that can trash the system, or reach into and use all of the devices on your computer. The name of this super user is “root.” It's all lower case. Typing in “Root” doesn't get you a root account. The EMC needs superuser access to things like the parallel port and the kernel to install special modules that it needs to plan and execute motion commands. You do not have to enter a password for root. You can click on the root password box and press [Enter} [Enter] and that keystroke will become the password used when you want to login as root.

You can also set up user accounts here. I like to set up an account for “operator” because I'd rather not leave all of the BDI exposed to passers by. The operator password can be as simple as the first digit of the machine it's running. In my case that is an “S” for Sherline. Now I press [Add] and [Next].

2.2.9 X-Configuration

There are a couple of screens here. I'll lump them together in this section. One caution, don't be in a big hurry to get past this.

X is the name of the window manager that is used with the BDI. Like MS-Windows, X-Windows is a way of making your monitor display pictures instead of just text. Here we need to work with the X configuration to confirm that it knows what your display driver (the chips inside your computer) and the monitor you have connected to it is capable of. It may show you the name of your monitor on the first page here. That may be good enough. If it does not show the name of your monitor you can look for it by typing the first letter of its name. V for Viewsonic or S for Sony. I've found that “generic non-interlaced SVGA 1024x768” is a good all around selection. It's enough resolution to display all of the setup screens. It is enough to display the mini or tkemc graphical interfaces to the EMC. [Enter]

The BDI install is pretty good at finding your driver. Most laptops use NeoMagic. There is a great variety in desktop machines. This is where you will want to test your configuration using the “Test this configuration” button. Don't quit the first time you see a tolerable display. Don't be afraid to use the back button and change monitor type as well as try different memory settings. You can also select the “customize X settings” button and try different resolutions and color depth. I found my laptop liked the combination of:

> Generic laptop display panel 124x768
>
> NeoMagic (laptop/notebook)
>
> 1024k memory

The text did not look real clear in the test screen. For other settings it mostly just came back to the configuration screen. I tried the “Customize X Configuration” button and [Next]. On that page I clicked the 1024x768 in the 8 Bit column. The text in the test screen was very clean, so I chose [Accept] and [Next] when it returned to the configuration page. I had to go back another page and click the “Use Graphical Login” button in order to be certain that it would start up in X-Windows. (Your install may vary here because some tests will ask this.)

2.3 Installing Packages

This is where the real installer work begins. [Next] will format the hard drive and put all of the necessary software on it. This display will show the progress of each package with the bar on top. It shows the progress of the whole install on the bottom bar. When the installer had done its job it will ask if you want a boot floppy. If you've got one, fine. If not, press the “skip boot disk” and [Enter].

Press [Exit] and after a moment it should eject the CD. Remove it and the floppy if you used one and let it reboot.

Supplemental guide for installing Sherline modifications when using the 8760 driver box package

If you are reading these instructions and have purchased a Sherline 8760 (A CNC External Driver Box) rather than a complete system with computer, you may be wondering why a Sherline logo mentioned in the instructions doesn't appear on your computer desktop after you install Linux. For this to occur, a few simple steps will need to be performed to add the Sherline modifications to the standard BDI install. First, you need to place the second Sherline CD (Disk 2) into the CD-Rom drive. Then open up a terminal. This is accomplished by clicking on the icon at the bottom of the screen that looks like a monitor. Once you have the terminal open, make sure that you are logged in as the root user. To do so type the following command:

```
whoami
```

Hit the [Enter] key. It should return with the value "root." If it does not, you are either logged in as the wrong user or the installation was not done properly.

Before going any further, make sure that you have run one of the programs on the desktop at least once. They have blue icons are labeled as "emc_mini" and "emc_generic." All you need to do is run one of them once and then continue.

In the terminal you opened previously, type the following series of commands, keeping in mind that everything is space and case sensitive. After each line, hit the [Enter] key.

```
cp /mnt/cdrom/sherline-mods.tar.gz /
cd /
tar zxvf sherline-mods.tar.gz
rpm -i --force emc-rtl3.0-2.18.i386.rpm
rm -fR /root/Desktop/emc_*
```

(Though it can be hard to see, there is a single space after "cp", "gz", "cd", "tar", "rpm", "-i", "rm" and "-fR"—in the lines above.)

Once you have done this, right click anywhere on the desktop. From the pop-up menu that appears, click on "Refresh Desktop". You should now see a Sherline logo on the desktop with the description "Inch_freq." This is Sherline's custom EMC program. Click on it. When it starts up and you see the main screen for EMC, you have successfully installed the Sherline modifications.

Frequently Asked Questions

By Mike Joyce

How hard is it to hook up the computer?

Sherline has made everything as easy as possible for you. The computer has the operating system and software pre-installed. The driver and power supply are also pre-installed inside the computer box. All you have to do is plug in some cords to get up and running. There are some good pictures in the printed step-by-step guide that came with your system. They show you how to plug in your new Sherline CNC computer and stepper motors. You can see these instructions in several places. In addition to the printed sheet, it is on your Sherline Documentation CD where it is called *quickstart.htm*. Here you can see photos of the system and the back of the computer to see exactly how to plug everything in. If you aren't familiar with computers, please be sure to pay attention, as you can damage the system if you force things into the wrong plugs.

I live outside the United States and our current is 230 volts. Can I still use my CNC system here?

Yes, but you will need to toggle two switches for this to happen. The back of the computer's power supply has a red switch for switching between 115 and 230 volts. You will need to flip this switch to 230. You can easily switch it with a small screwdriver or any other small flat device. Secondly, you will need to switch the driver board's power supply so that it will be set to use 230-volt power. There is a switch that is basically identical to the one on the back of the computer. You will need to open up the computer's side panel by unscrewing the two screws on the back. The side you will be unscrewing is the one with the power switch for the driver board. MAKE SURE that all power is unplugged, as even if the computer is off it will still have live power to the driver board. Once you have removed the side panel you will need to flip the switch in the middle of the power supply on the side facing you. It is somewhat embedded into the side, so you might need to get a screwdriver instead of just using your fingernail like you might be able to do with the switch on the back of the computer. Be sure that you have the right setting for the power in your area, as having it switch wrong it will cause damage to the system.

What format are the instructions in?

All of the instructions are at least available in HTML format [*.html*], this format is readable on just about any system without any troubles. Many of the files are also formatted in Adobe Acrobat [*.pdf*] and Microsoft Word [*.doc*]. It should be noted that PDF and Word files may not be readable on your system depending on the software you have installed. You will be able to open the HTML file in your favorite web browser.

Where can I find a current copy of all of the instructions on the Sherline Documentation CD?

An up-to-date version can always be found at http://www.sherline.com/CNCinst.htm.

What is Linux?

Linux is an operating system. An operating system is the basic set of programs and utilities that make your computer run. Some other common operating systems are UNIX (and its variants BSD, AIX, Solaris, HPUX, and others); DOS; Microsoft Windows; Amiga; and Mac OS.

What is EMC?

EMC is the Enhanced Machine Controller originally developed by National Institute of Science and Technology (NIST). It is released under the General Public License (GPL). Currently NIST does very little work on the project, and it has been taken over by average Linux users and a handful of very talented coders. You can get up to date on the project and get more detail on the movement of it by reading its website http://www.linuxcnc.org.

What programming language does the EMC system use?

The Sherline CNC system and EMC uses *g-code*. G-code is the standard language used to program most CNC machines in industry and is probably supported by your favorite CAD/CAM program. For simple shapes you can write the needed g-code yourself. For complicated shapes you can draw your part in a CAD program and export it to g-code, which can then be used in EMC. (Exporting to g-code may require a second translator program if your CAD program can't do it.) More can be read about g-code and its uses in the *CNCinst.htm* file included on the Sherline Documentation CD-ROM. Included with your CNC system is a free* trial version of the Vector32 CAD (Computer Aided Drafting) program.

*After six months, the program will need to be registered with Vector for lifetime use. They currently charge a $15 administrative fee to register your copy for you. Their full CAD/CAM program which can output g-code is also on the CD as well as several other helpful programs that can be unlocked and registered for a very reasonable price. Sherline owners receive a special discount from their normal price. See www.vectorcam.com for more details on the software.

What do I do if I don't know how to use Linux or CNC?

There are a couple of options here. You can click on the *EMCinst.html* on your desktop. The same set of instructions is also on the Sherline Documentation CD. You can get to it by putting the CD into your CD-ROM drive and opening the *index.html* file located in the *getting-started* directory on the CD. You also might want to take a look at the official Sherline Instructions; they are located both on your desktop, and on the Sherline Documentation CD. The file is called *CNCinst.htm*. There is also a large amount of documentation on the internet. If our documentation doesn't provide the answer you seek, try your favorite search engine and search for your specific problem/question, chances are somebody else has had the same problem. There are also several user groups you can join that will most likely help you with support. While these are not Sherline specific groups, they will most likely be familiar with the system you are using, and if not something very close to it. Be sure to research your question before posting to these lists. Some people get frustrated when people ask questions before doing a bit of research first. You can learn more about these mailing lists here: *http://www.linuxcnc.org/links.html*. When all else fails Ray Henry is available for technical support. He can be reached at rehenry@up.net and 906-875-6692. With your purchase you are entitled to up to one hour of free technical support from Ray. Additional consultation is available for a reasonable fee.

Okay, I have all this Linux stuff down, now I want to code some CNC!

Great! Well if you are new to the CNC community, it is suggested that you read the CNCinst.htm instruction file located on the Sherline Documentation CD and on your Linux Desktop. It is a great way for those who aren't familiar with CNC or G-Code to get started in coding. By the time you are finished reading the instructions you should be able to write simple programs and at the very least understand other, more complicated instructions.

How do I transfer my g-code files to my CNC computer from my CAD/CAM computer?

There are a few ways you can do this, but the easiest way is to put your g-code text files onto a floppy disk or write them to a CD on your CAD computer, then take the floppy disk or CD and put it into your Linux computer. On the desktop there will be an icon that says "Floppy." Click on it. It will open up the floppy drive. Simply drag and drop the file onto the desktop. When you are in the EMC, you can open this file by browsing to it. The Desktop is located in the */root/Desktop* directory.

How do I use my inch CNC system in metric mode?

You can toggle between metric and inch modes by placing metric code under a *g21* code as opposed to *g20*. If you are adding CNC to a Sherline machine that already has metric leadscrews you will need to run the *MM_step* file on the desktop.

My stepper motors make noise, but nothing is moving. What should I do?

The likely problem here is your mill is binding or not properly lubricated. With the more significant stress of quick back and forth motions and the high number of turns possible in a short period of time your mill will need to be more frequently lubricated than hand driven applications. Under continuous CNC use we recommend that Sherline CNC mills be lubricated once every four hours. Refer to the mill manual or go to http://www.sherline.com/lubricat.htm for more detailed lubrication instructions.

I have read the entire Sherline Documentation set front to back, now what to I do?

Well, our aim was to just start you on your path of learning Linux and CNC. There is a whole lot you can learn beyond these instructions. For starters you can check out the EMC website http://www.linuxcnc.org and get on the mailing lists. There are also more complex books and documentation on g-code available at your favorite bookstore or library.

What is GPL and what does it mean to me?

GPL is a license that Linux and EMC are both released under. Basically (very basically) it states that all code released under this license can be used, modified or sold as a prepackaged product, *but* it must be released under the same license after you modify it. This helps the Linux and other open source communities grow as people contribute to it. What does this mean to the average user? Nothing. If you decide to hack up the kernel, however, you are required to submit those changes back to the community. More information about GPL can be found here: http://www.gnu.org/copyleft/gpl.html

Why are my slides going in the wrong direction when I Jog?

This means that your ini file is improperly set. To fix it you will need to open the file on your desktop *mill_inch_step.ini* (if you are using a metric lead screw, use the *mill_mm-_step.ini* instead). To open it, just click on it. You will need to change *INPUT_SCALE* and *OUTPUT_SCALE* values from *+1600* or *-1600* to the opposite value; i.e., if yours is *+1600* and it is jogging backwards, change it to *-1600* and vise versa.

Other than calling Sherline, where are some places I can go to get help on my specific questions on machining, CNC and using Sherline tools?

Sherline owner discussion groups can be found at http://www.yahoogroups.com. Just go to the yahoogroups.com web site, type in the name of the group you wish to monitor or join and select it from the list provided. You can click on "join this group" to become a member or you can just read the archives without joining. These groups can be a very helpful resource for anyone new to machining and a great place to go to get advice from experienced users on specific problems. Here are some of the groups:
Sherline (1500+ members)
Sherline CNC (New May, 2003, 350+ members)
CAD/CAM/EDM/DRO (not Sherline specific, 3800+ members)

Are these instructions updated periodically?

Yes, we are constantly trying to improve the instructions. You can find the most up-to-date version of these instructions at http://www.sherline.com/CNCinst.htm. A .pdf version is also available at www.sherline.com/CNCinst.pdf. Compare the version number at the very beginning of the instructions to see if it is newer than the version you have been using. If you have questions or suggestions about this or other documents included with your CNC system or have other Sherline specific questions, you can contact me by e-mail. I can be reached via mike@sherline.com or 760-727-5857.

Supplemental guide for installing Sherline modifications when using the 8760 driver box package

If you are reading these instructions and have purchased a Sherline 8760 (A CNC External Driver Box) rather than a complete system with computer, you may be wondering why a Sherline logo mentioned in the instructions doesn't appear on your computer desktop after you install Linux. For this to occur, a few simple steps will need to be performed to add the Sherline modifications to the standard BDI install. First, you need to place the second Sherline CD (Disk 2) into the CD-Rom drive. Then open up a terminal. This is accomplished by clicking on the icon at the bottom of the screen that looks like a monitor. Once you have the terminal open, make sure that you are logged in as the root user. To do so type the following command:

```
whoami
```

Hit the [Enter] key. It should return with the value "root." If it does not, you are either logged in as the wrong user or the installation was not done properly.

Before going any further, make sure that you have run one of the programs on the desktop at least once. They have blue icons are labeled as "emc_mini" and "emc_generic." All you need to do is run one of them once and then continue.

In the terminal you opened previously, type the following series of commands, keeping in mind that everything is space and case sensitive. After each line, hit the [Enter] key.

```
cp /mnt/cdrom/sherline-mods.tar.gz
cd /
```

```
tar zxvf sherline-mods.tar.gz
rpm -i --force emc-rtl3.0-2.18.i386.rpm

rm -fR /root/Desktop/emc_*
```

(Though it can be hard to see, there is a single space after “cp”, “cd”, “tar”, “rpm” and “rm”—the first items in each line above. There is also a single space after “-i” and after “-fR” in the two last lines.)

Once you have done this, right click anywhere on the desktop. From the pop-up menu that appears, click on “Refresh Desktop”. You should now see a Sherline logo on the desktop with the description “Inch_freq.” This is Sherline’s custom EMC program. Click on it. When it starts up and you see the main screen for EMC, you have successfully installed the Sherline modifications.
