Quickstep Manual for Version 4 with PulsePacer

10/11/2005

Introduction - What is “Quickstep” 1
Quickstep Setup 4
Hardware Hookup 4
Software Setup 6
Procedure to setup stepper motor parameters 7
Quickstep Commands 10
G Code Command Descriptions 16
Excellon Drill File Import 20
DXF to G code ACE converter 21
The Layer Dialog Box 21
The Priority Dialog Box 22
Multi-pass Milling 22
Quickstep 4 Notes 23

Introduction - What is “Quickstep”

Quickstep is a very easy to use program for controlling X,Y, Z motion from a basic Windows PC environment. You can use this program to control milling machines, drilling machines, router tables, engravers, plasma cutting tables or any other equipment that has stepper motors on X, Y and Z axis. You can also use this software to control servomotors if you have digital type servomotor drivers such as Geckodrive 320 or Rutex drivers. Quickstep software provides first time users with a simple interface to process DXF type CAD files into G code commands. For users interested in making your own circuit boards, Quickstep also imports Excellon drill files and converts them into G code commands. The G code commands can be plotted on screen at any viewing angle or zoom level. G codes can be edited, re-plotted and saved as desired. The G code commands can then be translated into stepper motor commands that are used to control three stepper motors through the PC printer port.

This is Version 4 of Quickstep that is specifically designed for operating with the “PulsePacer” circuit attachment that plugs into your PC printer port. This version of Quickstep runs on all platforms of Windows (Win 95, 98, Me, 2000 and XP). The program will run on any Windows system but the PulsePacer is required to actually control your stepper motors. Instructions for installing the PulsePacer are included with the device when shipped. It is important that you follow the installation instructions in the exact sequence described.

The function of the PulsePacer is to precisely control all stepper motor pulse signals with an independent timing circuit that is buffered from your computer operating system. This enables Quickstep to run motors at much higher top speeds than are otherwise possible in a Windows operating environment. Depending on your specific system and whether or not you are using full steps or microsteps you should expect to see a 100-300% increase in top speed with the PulsePacer compared to earlier versions of Quickstep without the PulsePacer. Typical linear speeds (G00 and G01 moves) obtainable for various system platforms are below:

Pentium II 300 MHz, Win 95 system approx. 30,000 steps/sec for each axis simultaneously.
Pentium III 850 MHz, Win Me system approx. 60,000 steps/sec “
Pentium IV 2.4 GHz, Win XP system approx. 80,000 steps/sec “

This speed capability should satisfy all but the most demanding industrialist who doesn’t mind spending thousands more $$ for all out speed capability, encoder feedback, servomotors etc.

The program runs by sending data from the printer port to the PulsePacer. The PulsePacer then sends precisely timed signals to the DB25 port plug on the PulsePacer to your stepper drivers. The PulsePacer output controls motor steps, direction and enable for all three axis in real time as the program processes the G code commands and plots the moves on screen. Stepper drives such as Geckodrive, Xylotex, Vextra, Larken and most other commercial brands can be controlled through the PulsePacer with Quickstep software.

With this most recent version of Quickstep and the PulsePacer circuit the program can now deliver higher speed than most users will ever need.

The program commands were designed to be as intuitive as possible, at least to me, and hopefully to others as well. Some of the basic features include the ability to move on any axis manually at three user defined speeds or go to specified coordinates for common positions. As with all stepper motor controls, the motors start at a user definable start/stop speed (steps/sec) and then accelerate at a desired rate in steps/sec per sec. Each axis of motion can be calibrated to the appropriate number of steps per unit of length. The length units can be anything --- in, mm, cm or whatever you want. So the program has only a few basic units it functions with, units of length, seconds and steps. Other units that need to be specified on the "Setup" table are start/stop speed for each axis, acceleration rate, steps per unit of length for each axis and backlash for each axis. Rapid traverse speed, and cutting speed, are defined by the user as well in the "Setup" table. An optional “F” parameter can be specified at the end of any G code line to set a new cutting feedrate on the fly.

To date the program recognizes only absolute coordinates for lines, and relative coordinates for the center of arcs and circles. Straight lines and circular arcs are fully accelerated and decelerated to the maximum speed obtainable as specified in the setup table. Arc moves are true arcs (i.e. not broken into straight line segments). Software supported G and M code commands include:

G00 **(interpolated straight lines at rapid speed)**
G01 **(interpolated straight lines at cutting speed)**
G02 **(circular interpolation clockwise direction at cutting speed)**
G03 **(circular interpolation counterclockwise direction at cutting speed)**
G04 (this will initiate a pause in program execution that requires a continue command to restart, or just pause for 0 to 60 seconds as you specify then resume motion)
G17 (arcs, circles and helical moves on XY plane)
G18 and G19 (arcs and circles on XZ and YZ plane)
G28 (move to a hardware switch confirmed "Home" position for X, Y or Z axis)
G43 & G49 (tool length offset compensation on Z axis)

G51 & G50 (rescale G code to change size on all axis)
G54 & G53 (fixture offsets for X, Y and Z axis)
G81 and G83 (for drilling and "peck" drilling deep holes)
M03 & M05 (start and stop spindle motor)
M07 & M09 (start and stop coolant)
M010 & M011 (start and stop a device of your choice)
M47 (automatic repeat cycle from first line of G code)
F (feedrate control for all subsequent cutting moves)

Note: All user comments in G code file should begin with a left parenthesis “(“.   The G code interpreter will then ignore that line**.**

A detailed description of these codes and how they are used can be found in the Help\Commands section.

Since most CAD programs can export into DXF type files no other translator is included in this program. Other file types such as HPGL or WMF can be used if they are imported into your CAD program and then saved as DXF type files.

Some G code files use G1, G2 and G3 rather than G01, G02 and G03. To use these files you can use the Edit – Find – and “Replace All” command. Use to change all occurrences of “G” to “G0” and everything should work fine. It will change all instances of G1 to G01, G2 to G02 etc. in just a fraction of a second.

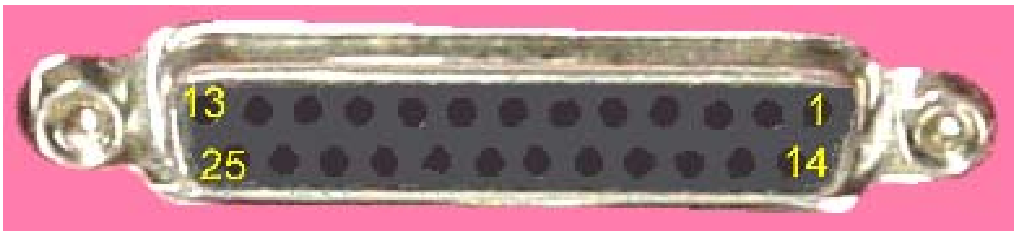
Some G code files also omit the G01 command when designating a long series of destinations. If no “G01” is on a line of G code then Quickstep assumes it is a cutting move and will then send out motor signals that go to whatever X, Y and Z destinations are specified on that line. That move will be at the default cutting speed specified in the “Setup table” or the last “F” cutting speed if so specified.

This is Version 4.2 of Quickstep and I expect that as user feedback dictates I will add more features.

Quickstep Setup

Hardware Hookup

Assuming you have your stepper motors and drivers already, you must be sure that all this stuff doesn't overload your PulsePacer that plugs into the printer port. The enable, direction, and step inputs should draw less than 1 ma each. You can check this with a multimeter if you are not sure. The PulsePacer output connections can only have two output states at each pin, 0 volts or +5 volts and your stepper drivers must respond to this signal level. If your stepper drivers draw excessive current then you can damage the PulsePacer or your printer port. All this being said the connections to the PulsePacer output port are as follows:



PulsePacer Output Plug Pin Numbers

Note that these are the same pin connections used to the printer port plug in earlier versions of Quickstep so you will not need to change your wiring to the male DB25 plug that plugs into the PulsePacer if you have already been using earlier versions of Quickstep.

pin 2 = x step
pin 3 = x direction
pin 4 = enable for all three motors
pin 5 = y step
pin 6 = y direction
pin 7 = z step
pin 8 = z direction
pin 10 = home limit switch (optional)
pin 11= move interrupt and axis limit switches (optional)
pins 18-25 = logic ground (you only need to connect to one of these pins)

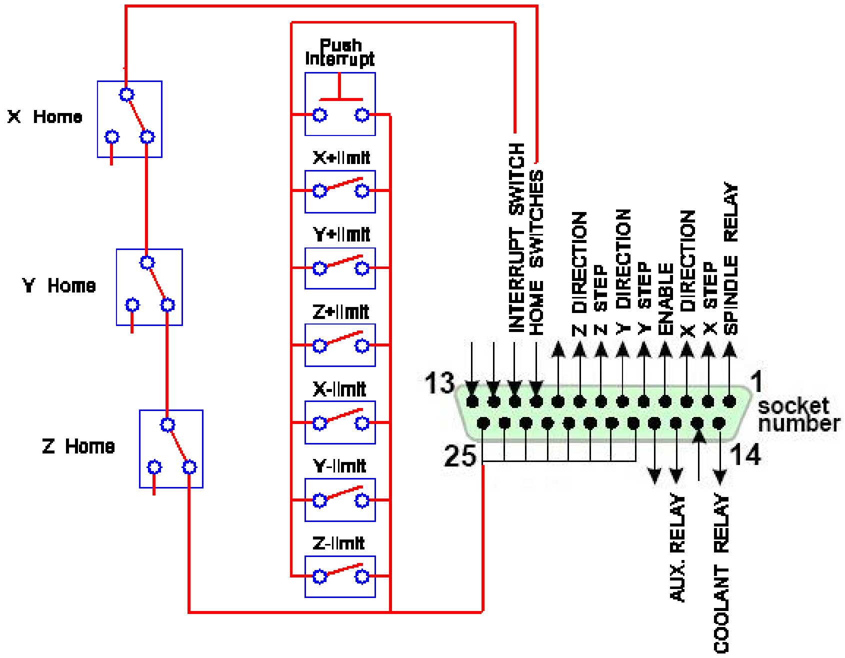
In case you wondering, no these pin assignments can not be changed.

You don't have to worry about polarity on pins 2-8 since this is taken care of in the "Setup Table". Check boxes in the Setup Table allow you to reverse the signal polarity. If your stepper drivers step when the step input signal goes low then uncheck the step signal polarity check box. If your motors are enabled when their enable input is low then you should check the enable signal polarity checkbox.

Home limit switches can be hooked up to one, two or all three axes. The limit switches need to be normally closed type switches. Each switch must be positioned so that when an axis moves in the negative direction then the switch is tripped at home position and creates an open circuit. For two or three axis, the switches are hooked up in a series X –Y and Z. One end of the series of switches is connected to logic ground (pins 18-25). The other connection is made to pin 10. Note that the switches need to be installed so that when the axis heads toward the home position at full speed and trip the switch it will stay tripped open and not get crunched within the travel distance it takes to decelerate from full speed to the start/stop speed. Remember, each axis you "Home" will move at full speed in the negative direction until the switch is tripped so beware.

Move interrupt and axis limit switches all connect up to pin 11. The move interrupt should be a momentary contact switch that is within easy reach. One terminal of the switch connects to ground (pins 18-21) and the other terminal of switch connects to pin 11. If you have axis limit switches then they all connect up the same way. One connection goes to ground, the other to pin 11. If pin 11 goes to ground from any axis limit switch, or the move interrupt switch, then your G code move will immediately decelerate to a stop. Note that if any of the axis limit or move interrupt switches are tripped (closed) then you can only move in the manual mode with the push buttons to deactivate the axis limit switches.

Quickstep hookup diagram for "Move Interrupt", "Axis limit" and "Home" switches. Home switches are normally closed type. Interrupt and axis limit switches are normally open type switches. Axis limit switches and Home switches are all optional. All pins not identified on printer plug are to be left unconnected.



Printer plug as seen from back of PC or DB25 female plug on PulsePacer.

For the spindle, coolant, and auxiliary motor, the pin connections are as follows:

pin 1= spindle motor (+5 volts is off and 0 volts is on)
pin 14 = coolant motor (+5 volts is off and 0 volts is on)
pin 16 = auxiliary motor (+5 volts is off and 0 volts is on)

Note that for all three motors the off state is always +5 volts. This is because when most computers boot up the pins are at +5 volt state already and you don't want motors to automatically come on every time you turn on the computer. If you intend to use computer signals for control of spindle motor you will need a solid state relay capable of handling the necessary high voltage and current of large motors. An inverting driver IC will also be needed to interface between the printer port signal and the solid-state relay. Note that there are some solid state optically isolated 120 volt relays now that will accept 5 volt logic signals and draw less than 1 ma so they can be connected directly to the output pins on the PulsePacer without damaging the Pulsepacer or printer port.

Software Setup

From "Setup" on the main menu you get to the CNC setup table. Here you enter all the information required to match your particular machine/robot/or whatever you have got. This enables you to enter the steps per unit of length on each axis, instantaneous start stop speed, acceleration rate and maximum step rates. You can invert the signal going to the step or direction for each axis. You can also invert the enable signal for all the motors. Cutting speed and rapid travel speed are entered here. Three speeds are specified to select for manual moves with the mouse activated command buttons. Acceleration and deceleration are automatic whenever a specified speed is higher than the instantaneous start stop speed. Backlash is always automatically taken into account on all linear moves whenever the direction changes. For circles and arcs, however, you have an option check box that disables backlash at quadrants changes if you need speed, as opposed to precision. Otherwise, the speed of arcs cannot exceed the maximum start stop speed. This is necessary to correct for backlash instantly at each quadrant change while moving through arcs.

You need to select the printer output port address from the selection options. Almost all computers have the printer port on address &H378. If there is a second printer port it is usually at &H278. Actually you can enter any number here as long as it corresponds to the port address you intend to plug the PulsePacer into. Just don't forget to include the "&H" symbols before the number to indicate that the address number is a hexadecimal number.

PulsePacer switch setting is the numerical value corresponding to the positions of the four mini-switches inside the PulsePacer box. Each PulsePacer comes with a data sheet that has the numbers corresponding to all available switch positions. To find the correct switch setting for your computer you need to measure some information on your system. To do this you first need to unplug the PulsePacer or anything else that you may have plugged into the port you intend to use for the PulsePacer. Then you press the "Calculate PulsePacer rate" button on the Setup screen. Now wait, in about a minute you will see further instructions on how to complete the setup. It is essential that you follow the instructions to set the PulsePacer switches correctly and enter the correct value for your switch settings on the Setup screen in the PulsePacer switch setting box. Remember to press the "Save settings as default" button after you enter the value corresponding to your switch settings.

Cutter width is the diameter of the milling cutter that you intend to use. When you check the

"Show cut width" option in the G code plotting window the program will draw all cutting moves with a line width equal to the cutter diameter multiplied by the cosine of the tilt viewing angle. It will only be shown true to scale when the tilt angle is zero and you are either at 1024 x 768 or 1280 x 1024 pixel resolution.

Procedure to setup stepper motor parameters

Many people don't know where to begin when it comes to setting the motor parameters for acceleration rate, maximum step rate and start/stop speed. The following procedures should be sufficient to get most systems operational.

Motor speed is limited by available torque of motor, friction, inertia of the rotating mass, motor driver voltage, and computer speed (if you have an older computer less than 1GHz). Every system needs to be "tuned/tweaked" if you want to get optimum speed capability.

If your computer is a PII 300 MHz then it will limit your output signal to about 30000 steps per second. For a PII 900 MHz this goes up to about 60,000 steps per second for each of the three axis. For a 2.5GHz Windows XP machine maximum speed will be approx 80,000 steps/sec. If you are running microsteps like 2000 steps per revolution then you may need these very high step rates.

If you have set the PulsePacer switches correctly for your specific machine and entered the exact value corresponding to the mini-switch positions then the PulsePacer will put out the exact pulse rates you specify in the Setup Table. If you want to double check the output speed then you can make a test move and time it with a stopwatch as follows.

To check speed set the maximum step rate to 1000 steps per second, the start stop speed for all three axis to 200 steps per second and the acceleration rate at (1000 steps per second) per second. Now you need to make a long move on the X or Y axis equal to some even number of steps that you can time with a stopwatch. This needs to be a "rapid traverse" move (i.e. G00) that you can execute with the "Move to" command. For example, let's say your system is setup with a 200 step per revolution motor and it is direct drive to a 10 thread per inch leadscrew. This works out to 2000 steps per inch (200 x 10). With the "Move to" command make a 5.000 inch move. Neglecting acceleration and deceleration time, the move should take ~10.0 seconds. You will have moved 10,000 steps at a rate of 1000 steps per seconds so your stopwatch reading from start to finish should be ~10.0 seconds. If you know the details of all the physics and take into account all the acceleration and deceleration time it should take exactly 10.64 seconds.

Remember that stuff you learned in high school, distance=speed x time
for accelerating bodies distance=1/2 acceleration rate x time x time.

Acceleration time = (1000-200)/1000 = 0.80 seconds
Deceleration time = 0.80 seconds
Acceleration distance =(200 x 0.80) + (0.5 x 1000 x 0.8 x 0.8) = 480 steps
Total time for 10,000 steps = 0.80 + 0.80 + (10,000 – (480 x 2)) / 1000 = 10.64 sec

If the speed is not correct or the speed is not very smooth then your PulsePacer was not set correctly or the value of the PulsePacer switch setting was not entered correctly in the Setup

Table.

Now you need to get into the more difficult phase of tuning. Change the maximum step rate to 700 and acceleration rate to 500 steps per second per second in the Setup Table. Using these parameters in the Setup Table you will need to make moves at increasing maximum step rates until you see and hear the motor go out of phase and lose steps. You need to do this part of the tuning with the axis that has the most inertia/friction load. For something like a milling machine this is always the long X axis which has a much longer screw to turn. Zero your micrometer dial on the machine and then make the 5.000 inch moves at increasing maximum step rates until you lose steps. In the Setup Table you also need to set the “Rapid traverse speed” to something ridiculously high like 20.0 units per second so this parameter will not limit the rapid traverse speed (i.e. maximum speed will be limited by maximum step rate parameter only). If you are running motors with full steps I suggest increasing the max step rate from 700 to 800 then 900 then 1000 and so on until steps are skipping.

If you are running micro steps like 2000 steps per revolution, then you have a long way to go. You should change the start stop speed to 1000 and acceleration rate to 5000 in the Setup Table. Now you need to increase the maximum step rate in increments of 500 like 1500 then 2000 then 2500 etc until you lose steps.

Once you establish what speed is too fast then I suggest that you set your maximum step rate to about 70% of what you know is too fast. Make a few long moves and check your exact position with the graduations on the handwheel dials at the end of each move. Remember your backlash compensation will also affect handwheel position.

Now that you have the maximum step rate set you need to go through a similar procedure of progressively increasing the acceleration rate until the motor will not keep up with step signals. For systems running at full steps i.e. 200 steps/revolution, I suggest increasing acceleration rate from 500 to 550 then 600 etc until steps are lost. For microstep systems running at 2000 steps per revolution, I suggest increasing acceleration rate from 5000 to 5500 then 6000 then 6500 etc until steps are lost.

When you know what is too fast then back off to about 70% of that rate and you should be safe at that acceleration rate.

Now that you have max step rate and acceleration rate set you need to use a different trial and error method for determining the maximum start/stop speed. You can use very short moves for this procedure. Actually a series of short moves will be used to tune this parameter. For a full step system, start at 200 steps/sec. For a microstep system, start at 1000 steps/sec.

Make sure you have the proper value of backlash entered in the Setup Table for all axis. Now with the “Move to” command move in the negative direction -0.10 units on the X axis. This move is just to take up the backlash going in negative direction. Press the “Zero” button for X axis to zero the position in program counter. At this point you should also zero your position counter on the machine handwheel dial, DRO, or a dial indicator set up to measure true X position. With the “Move to” command make a series of moves in the positive direction on the X axis, such as 0.10, then 0.20, then 0.30 then 0.40. Now with the “Move to” command go back to 0.0000 position. It should take you back exactly to the 0.0000 reading on your handwheel dial, DRO, or dial indicator. If that position is not correct then slow down the start/stop speed about 10% and repeat the full procedure until position error is eliminated. If you had no error, then you can probably increase the start/stop speed about 10% and repeat the procedure until you know what the limit is for the X axis start/stop speed.

When you know what your maximum start/stop speed is then back off your final setting to about 70% of what you know is too fast and you should be safe. You may also want to do the start stop speed tuning for each axis. Usually an axis that has less inertia/friction will start a lot faster and you might as well get each axis to start as fast as possible.

Quickstep Commands

Note: G codes are described in the last half of this section. To print or edit Help sections see the Help\User Notes section.

"Plot" – This plots the G code shown in the text window onto the screen at the desired rotation and tilt angle. This plot represents the tool path that will be generated when the G code is executed. Plot will always start the first line at 0.0, 0.0, 0.0 X, Y, Z coordinates regardless of where the machines current position may be. Pause commands in the G code are ignored when plotted.

"Z ++" – Zoom control. Just select a point on the drawing plotted and drag a box down and to the right, then release left mouse button. This will zoom the box area you just defined to full screen magnification. Zoom is limited to a maximum value of 40.

"Pan" – Select a point of interest with left mouse button and drag that point to a new destination then release mouse button.

"Show cut width" -- Checking this option box will display all G code cutting moves (G01, G02, G03 and the vertical feed of drill cycles G81 and G83) as lines equal in width to the cutter diameter entered in the Setup Table. Note that the line width is only to scale when the tilt angle is set to zero degrees. Otherwise it is not to scale but is simply displayed as the cutter width times the cosine of the tilt angle.

"Position" -- This displays the X, Y, Z positions of each axis. When in manual mode and at slow speeds the position is updated in real time as the move progresses. When the move is faster where acceleration and deceleration are required then the position is updated when deceleration has completed. When in CNC mode the destination position is displayed when each move is completed. This display always shows the absolute position even if tool or fixture offsets are active.

"Offset" – Whenever you Plot, Simulate or Run G code that has commanded a tool height offset (G43), or fixture offset (G54) a view frame will open to the right side of the Position display. This frame will display the current offset for each axis specified by your G43 and G54 code. Use G49 and G53 commands at the end of your G code to remove that view frame. When both G43 and G54 are specified then the sum of the tool height and Z axis fixture offset will be displayed. Unless G43 or G54 commands are in your G code all offsets are assumed to be zero each time you Plot, Simulate or Run G code.

"**Zero**" --These buttons will zero the position counter for each axis. They do not move the motor positions.

"**Set X Y Z**" -- This button sets the position counters to whatever the display position has been edited to read. For example, you can align the tool to a hole centerline then simply enter the X, Y, Z coordinates of that hole in the Position display and hit "Set X Y Z". Now the correct center of hole is set in the machine coordinate system.

“Home” – When you depress the home button for any of the axis, Quickstep will automatically send that axis in the negative direction at maximum speed until the home limit switch is tripped open. Then it will decelerate to a stop and reverse direction of that axis. It will move in the positive direction at the start/stop speed until the limit switch returns to its normally closed status. This is the precise “Home” position where the backlash has been taken up and the switch just returns to its normally closed state. When “All Three Axis” button is depressed this sequence of moves is made for the X axis, then the Y axis and then the Z axis. Warning ----- Do not try to home an axis that is not connected to a home limit switch or it will not stop going in the negative direction.

"**Move to - ?**" -- This input box opens a Favorite Position table that you use to make manual moves to commonly desired positions such as Home, tool changes etc. You simply enter the x, y, z coordinates and depress "Go to it Now" button. Travel speed is rapid traverse speed by default.

One of the “move to” options is labeled “Up & Over”. This option is used for making two consecutive moves such as you may need for tool changes. Selecting this option enables you to move the tool up above the work piece and then immediately traverse over to a tool change position. For this option the first X and Y positions are automatically filled in to whatever the current position is at that time. The Z position can be set to whatever you want to raise the tool to on the Z axis. The second set of X, Y, Z coordinates define the next position you go to immediately after the Z axis moves up.

The bottom favorite position labeled “Current Pos” is always automatically updated to the current position so that if you want to move on only one axis then the coordinates for the other axis are already entered in the input boxes. You just edit the value for the axis you want to move on, then hit “Go to it Now”. In the registered version of the program all the “favorite position” values are saved when the setup table values are saved with the “Save Settings” command button.

If you have invoked an “Interrupt Move” (not available in Demo version) the position at that interrupt is also available as an option so that you can return immediately to that position and restart your G code run at that point.

"**Simulate**" -- This runs the current G code at the relative speed specified. The X, Y and Z drive motors are not enabled in this mode so you can use this option to watch G code being run line by line to check everything out before you ruin something. When speed is set at 1 then it runs at actual cutting speeds specified in "Setup" table and any F commands in your file. When speed is set at 10 it runs ten times faster. During a Simulation run the position display updates the G code destinations as they are in progress but at the end of simulation the display positions go back to the actual current positions since nothing has really moved. In “Simulate” and in “Run G Code” mode the plot on screen will start the first line at the machine position shown in the X, Y, Z position counters. After you start a Simulation run you can stop and start it one line at a time by checking the “One Line” option button.

"**Run G Code Now**" -- This is what starts the real G code execution. When you are sure that everything is ready to go, this is what starts the execution of G code moves. The first move starts from the position shown on the X, Y, Z position counters. Make sure the Z position is correct or you just may have a collision on the first move ! Yes, I have made this mistake more than once.

“One Line” – Checking this option will enable the motors and run your G code just one line and then go into a Pause state. You can then continue running your G code one line at a time by rechecking this option for each line. You can also check the “Continue” option to run the remaining lines of G code non-stop.

"**Pause Run**" -- This stops G code execution at the completion of the move currently in progress. It does not immediately stop the move (see “Move Interrupt” below for immediate stops) so you may want to have your own emergency "Kill" switch going to motor enables and any other moving parts such as spindle etc. Of course you can also depress “Ctrl Alt Delete” to stop the whole program for an emergency. “Pause Run” option can be used in Simulation mode also. A G code of G04 on any line will automatically initiate a “Pause” that is restarted by the user clicking on the “Continue” button. A G04 command can therefore be placed in the G code file on any new line for desired tool changes etc. A line containing “G04 sec=4” will cause a momentary pause in the motion for 4.0 seconds, then the next move will proceed. Use all lower case letters for “sec”.

“Interrupt Move” – This stops all stepper motor motion immediately. High speed moves are decelerated to a controlled stop and the true position is displayed in the Position readout display. When an interrupt has occurred you have two options. You can depress the “Yes” button in the message box which puts program into the “Pause” state where you can make feedrate override adjustments or change tools using the “Move to ?” button. If you choose not to go into the Pause state then the “No” option will end the G code run that is active. “Interrupt Move” is activated by a momentary push button that is intended to be mounted on the machine within easy reach at a seconds notice. If you have wired up your axis limit switches these switches will invoke the same response. Note that this feature is not available on the Demo program. See “Help/Setup” for hookup of this feature.

If you have invoked an “Interrupt Move” and are ready to finish that move and subsequent G code steps then hit “Continue” to restart where you left off. If however you made a manual move after the interrupt stop then you must go back to the exact position that the interrupt stopped at before you hit “Continue”. A Warning message box will appear and prompt you to go back to the interrupt position. This interrupt position is automatically displayed as one of the options in the “Move to ?” window. Note that when you go from the position you are at and head toward the last interrupt position it will be in a straight line so you may need to make sure that the Z position does not cause a crash if your interrupt occurred with the cutter in a pocket.

If an interrupt has occurred from an axis limit switch then no further moves can be made unless you are in the “Manual Mode” and use the manual move push buttons.

"**Continue**" -- This restarts G code execution if a "Pause" or “Interrupt Move” has been enacted.

"**End Program Execution**" -- This ends G code execution after the current move has been completed. Again you may want to have an emergency “Kill” on your machine motors if you anticipate emergency stops without deceleration etc. A push button connected in series with the motor enable connection works fine for this. Of course you can also depress "Ctrl Alt Delete" to stop the whole program for an emergency.

"F" – Feedrate, is an optional parameter to reset the default feedrate for all subsequent cutting moves (G01, G02, G03, G81, G83). The F feedrate, in units of length per min can be set with a parameter like "F2.2" at the end of any line of G code. This will set the feedrate to 2.2 units of length per min. If no F parameter is specified then the default speeds from setup table will be assumed. On the left side of the "Feedrate Override" scrollbar you will see the currently active feedrate displayed. If a new F is specified in your G code, then that new F value multiplied by whatever feederate override is set on the scrollbar will be displayed as you run or simulate the G code. The "F" must be on a line containing G00, G01, G02, G03, G81 or G83 command or it will be ignored.

"Feedrate Override" – If you want to override the current feedrate then you can invoke a stop by using either the "Interrupt Move" or "Pause" command and then move the feedrate override scrollbar. When set at 100% the feedrate = F. At 200% feedrate = F x 2. At 20% feedrate = F x 0.20. After setting the scrollbar hit "Continue" to restart again at the new feedrate. The current feedrate as modified by the value of override is displayed to the left of the feedrate override scrollbar. You will see this value change as you move the scrollbar.

"Teach Mode" – This is a mode of operation that allows users to make simple moves of their machine and Quickstep will automatically turn these moves into G code for future repeated use. To use Teach mode select the "Teach" mode option. Now you simply make a move with the manual push buttons or one of the "Move to ?" option buttons. When you are at the desired destination then you depress either the G00 or the G01 button in the Teach Mode window and the corresponding G code will appear in the G code view window. If you desire a change in the cutting speed just enter F parameter before you hit G01 button. Now make the next move with mouse buttons and hit G00 or G01 button. Again the corresponding G code will appear in the G code view window after the last line of G code. You can preview the G code moves at any time by hitting the plot button. At any time you can also simulate or actually run the G code after it has been plotted. Could it be any easier ?

Alternate Teach method

If you want to write lines of G code without actually running any of the motors you can also just type in the desired final destinations in the "Position" displays and then use the Teach window buttons to write the G00, G01, G02 or G03 lines into the G code window. This method saves a lot of typing time since you only need to change the position values for the axis that you want to move on. The Teach mode functions by assuming the values in the Position display, are also the final position values that you want that G code move to end at. Note that you do not need to hit the "Set X Y Z" button each time you change the values in the Position display.

Arcs and Circles on XY Plane

Now for arcs and circles. This is a bit more tricky if you are not accustomed to G code I and J arc parameters. First you move to the desired final arc destination but then you must also enter the I and J parameters. I is the distance on the X axis from the starting point of the arc to the center of the arc. J is the distance on the Y axis from the starting point of the arc to the center of the arc. I and J are both relative distances from the starting point of the arc. The starting point of the arc is the position that the previous line of G code took you to. Of course the final arc destination point is the point you just moved to. I and J can be positive or negative values but they can not both be zero since you would not have an arc if you did not have a radius, right ? If you try arc/circle moves and the final arc destination does not lie on an arc from the I, J center point that you defined then Quickstep will automatically adjust the end point of arc to the closest X , Y point that intersects the arc. This is no simple mathematical feat but it does work. For consecutive full circles from the same starting point see special note ** below. If you have a project with lots of odd angular tangents to arcs and circles it is easier to use a CAD program and let it do the trig for you but Teach mode works very well for simple spur of the moment machine paths

Helical Interpolation

When completing arcs and circles on the XY plane you can modify the path into a 3 dimensional helix simply by adding a final Z coordinate after the value of J (i.e. G02 X1.0 Y1.0 I0.5 J0.0 Z0.5). This specifies that the final Z coordinate of the helical move will be at Z=0.5. If you want the helix to be more than one full 360 degree revolution you can also specify another parameter after the Z value to turn the move into as many revolutions as you like to get to the same final X Y Z destination. You need to specify an optional "A" parameter which stands for "**A**dditional full revolutions". A value of "A4" would convert a one half revolution helix into a 4.5 revolution helix. A value of "A4" would convert a full circle helix into a five revolution helix ending at the same X Y Z coordinates specified. The value of "A" must be positive, and a whole number (i.e. G02 X1.0 Y1.0 I0.5 J0.0 Z0.5 A4). Helical paths can only be specified when working in the G17 mode with arcs on the XY plane. See "Demohelix.text" for examples of these arcs.

Arcs on XZ and YZ Planes

To teach arcs and circles on XZ or YZ planes you select the XZ or YZ option in the "Teach" window. You will see that G18 or G19 automatically is written into the G code window and the input boxes for values of I and J are now input boxes for I and K or J and K relative coordinates to the center of the arc you wish to draw. Like I and J, the value of "K" is the relative distance, but in the Z direction, of the center of the arc from the starting position of the arc you are about to draw. Unless otherwise specified by a G18 or G19 command then the G17 mode is assumed and the arcs and circles will be on the XY plane at whatever Z value is current. You must always invoke a mode change with G17, G18 or G19 commands when you specify arcs on a different plane. See "Demohelix.text" for examples of these arcs.

** Note – When you have two or more consecutive 360 degree circles all starting from the same point it may appear that the later circle command is being ignored. No, it is not being ignored. It is actually being misinterpreted as a very tiny arc segment that extends from a miniscule position error from the previous circles end point to the next circles end point. This is a rounding error that may occur in the 6th or 7th decimal point. If this occurs then enter a G00 Xxx.xxx Yyy.yyy Zzz.zzz line of G code between the first and second circle commands. In this line of G code the xx.xxx, yy.yyy and zz.zzz need to be the identical values of X, Y and Z position where the second circle end point is to be. This basically confirms that the starting point of the circle is exactly identical to the end point of the circle.

Drill Cycles G81 and G83

For drilling cycles the move you made just before any G81 or G83 command will preset the Z elevation where the drill cycle does the initial rapid traverse on the XY plane and the final Z height where the drill cycle ends. The input boxes for drill cycles in the "Teach" mode require you to specify the X and Y coordinates where the hole is to be drilled, the retract height on the Z axis "R", and the Z depth of the hole to be drilled. For "Peck" drilling (G83) you must also specify the depth of each peck "Q" on the Z axis. Q must be a positive value. The drill demo files show these commands in action so you can plot and simulate them to visualize what occurs.

Example of G81:

Assume that your last move completed was G00 X4.0 Y4.0 Z2.0
Next line of G code reads G81 X5.0 Y6.0 Z0.0 R1.0 F1.0
The first move that occurs will be a rapid traverse to X5.0 Y6.0 Z2.0
Next move is a rapid traverse down to Z1.0 (this is the "Retract" height)
Next is a slow feed down to Z0.0 at F=1.0 units/min
Final move is rapid move up to X5.0 Y6.0 Z2.0

Example of G83 "Peck Drilling":

Assume that your last move completed was G00 X4.0 Y4.0 Z2.0
Next line of G code reads G83 X5.0 Y6.0 Z0.0 R1.0 Q0.34 F1.0
The first move that occurs will be a rapid traverse to X5.0 Y6.0 Z2.0
Next move is a rapid traverse down to Z1.0 (this is the "Retract" height)
Next is a slow feed down to Z0.66 at F=1.0 units/min
Next move is rapid up to X5.0 Y6.0 Z1.0
Next move is rapid down to Z0.66
Next move is slow feed down to Z0.32
Next move is rapid up to Z1.0
Next move is rapid down to Z0.32
Next move is slow feed down to Z0.00
Final move is rapid up to X5.0 Y6.0 Z2.0

Note: If you need to change drills at any point then insert a G04 command with your reminder message of drill size just before the G81 command. This will stop the execution of G code and allow you to change drills. Then you can hit the "Continue" option box to continue running subsequent lines of G code.

G Code Command Descriptions

G00 Interpolated straight line move at rapid traverse speed. The speed will be as specified in the Setup Table. If that speed exceeds the maximum step rate in the Setup Table then the maximum step rate will be used. Values for X, Y and Z are all required to be in absolute coordinates. X, Y and Z should be in upper case fonts. If only Z value is specified on the line then it is assumed that X and Y values do not change for that move. Examples – G00 X2.0 Y2.0 Z2.0 or G00 Z4.00 or G00 X2.00 Y4.00 If no G00 is on any line but an X, Y or Z character is found on the line then it will be interpreted as a G01 type move. Example – X2.0 Y will be interpreted as G01 X2.0 Y0.000 No value specified for Y is interpreted as Y=0.000

G01 Same as G00 except the speed will be as specified in the Setup Table for cutting speed in units per min. An optional "F" parameter can be included on the line to change the cutting speed for that move and every other cutting move thereafter. Example-- G00 X4.0 Y6.0 Z2.0 F22.2
If no G00 or G01 is on any line but an X, Y or Z character is found on the line then it will be interpreted as a G01 type move. Example – X2.0 Y will be interpreted as G01 X2.0 Y0.000 No value specified for Y is interpreted as Y=0.000

G02 Circular or helical moves in a clockwise direction at cutting speed. This clockwise direction is as viewed from looking down the +Z axis onto the XY plane. Starting point for the arc is where your previous move ended. The X and Y coordinates specified are the end point of the arc. The I distance is the relative distance in the X direction from the starting point of the arc to the center point of the arc. The J distance is the relative distance in the Y direction from the starting point of the arc to the center of the arc. Example -- G02 X2.0 Y2.0 I1.0 J0.0 If your previous move left you at X0.00 Y2.0 then this will make a half circle with the center point at X=1.00 Y=2.00

If an additional Z coordinate is specified then the arc will be a helix ending at the X, Y and Z coordinates specified. For a helix with more than one full revolution an additional "A" parameter can be specified to add full circles of revolution. "A" must be a whole number. For example, a half circle helix with an optional A3 parameter will produce a helix of 3.5 revolutions.
Example – G02 X2.0 Y2.0 I1.0 J0.0 A3 See Demohelix for an example of this code.

G03 Circular or helical moves in a clockwise direction at cutting speed. Same as G02 except arc move is in counter clockwise direction.

G04 This will initiate a pause in program execution that requires a continue command to restart. When G04 includes a "seconds" parameter (i.e. G04 sec=4) to define a specific delay time, then a program delay is automatically processed. Delays can be up to 60 seconds max.

G17 This command is used to complete arcs and circles on the XY plane. An additional coordinate parameter for Z can be specified to convert an arc or circle into a helical path. A second additional parameter "A" can be specified to add full circles to a G02 or G03 move to produce a helical path with more than one full revolution. "A" must be a whole and positive number. See “Demohelix.txt” example. G17 does not need to be specified since it is the default mode unless otherwise changed by a G18 or G19 command.

G18 This command is used to complete arcs and circles on the XZ plane. Direction CW or CCW is as you would see the direction looking down the +Y axis onto the XZ plane. All arcs after changing to this mode must specify X, Z, I and K parameters. K is the relative distance from the arc starting point to the arc center in the Z direction.
Example – G02 X2.0 Z2.0 I1.0 K1.0

G19 This command is used to complete arcs and circles on the YZ plane. . Direction CW or CCW is as you would see the direction looking down the +X axis onto the YZ plane. All arcs after changing to this mode must specify Y, Z, J and K parameters. K is the relative distance from the arc starting point to the arc center in the Z direction.
Example – G02 Y2.0 Z2.0 J1.0 K1.0

G28 Move to a hardware switch confirmed "Home" position for X, Y or Z axis. Specify one axis for each G28 command. Speed is same as G00 rapid traverse. Example -- G28 X This will move to the X=0.000 Home switch position the same as if you had pressed the “Home X Axis” push button in the “Move to” menu. When you plot or simulate a G28 command it will be shown as a bright green line. To home all three axis use three G28 moves:

```
G28 X
G28 Y
G28 Z
```

G43 Tool length offset compensation on Z axis. You can specify + or – compensation. After this command all subsequent G code moves will be offset in the Z direction by that amount. The tool length offset will be summed with whatever fixture offsets G54 you may have specified and the summed distance will be shown in the Offset frame to the right of Position indicator. Make sure you cancel a G43 with a G49 command to make the Offset frame disappear after completing the run or simulation of your G code. Whenever you run or simulate G code the tool offset is 0.000 unless you specify otherwise with a G43 command. Example – G43 1.25 All subsequent G code moves will be shifted 1.25 units upward from the G code you are running. See Demo4.

G49 Resets tool length offset to 0.000. Use this and G53 at the end of your G code to make sure the Offset frame disappears at the end of your run. Example -- G49 See Demo4.

G51 Rescales G code to change size on all axis proportionately. Axis cannot be rescaled individually or arcs and circles would become elliptical and this is not allowed.
Example – G51 2.00 This would double the scale of all subsequent G code moves. See Demo4 for examples.

G50 Resets scale to 1.000. Example – G50 See Demo4 for example.

G54 Fixture offsets for X, Y and Z axis. After specifying distance offsets on one, two or all three axis then all subsequent G code moves will be shifted by that amount. When offsets have been specified then an Offset window will display the amount of offset being applied to each axis. If tool length offset is active then that will be summed with any Z axis offset specified. The Position indicator will always display your tool position in absolute coordinates. Make sure you cancel the G54 with a G53 command at the end of your G code to make the Offset window disappear. Whenever you Run or Simulate your G code it assumes that offsets are all zero until a G54 command is encountered. Example – G54 X1.00 Y2.00 Z0.00 See Demo4 for examples.

G53 Resets all fixture offsets to 0.000. This will make Offset window disappear unless a tool length offset is still in effect. Example – G53 See Demo4 for example.

G81 Drill cycle with retract move. You specify an X, Y position for the hole, a depth of the hole (Z), and a retract height (R). The drill cycle will always end at the same height on the Z axis that you started at. The retract height is where the downward speed transitions from a rapid move downward to your specified feedrate downward.

Note: If you need to change drills at any point then insert a G04 command with your reminder message of drill size just before the G81 command. This will stop the execution of G code and allow you to change drills. Then you can hit the "Continue" option button to continue running subsequent lines of G code. See Drilldemo file for examples.

Example of G81 X5.0 Y6.0 Z0.0 R1.0 F1.0:

> Assume that your last move completed was G00 X4.0 Y4.0 Z2.0
> Next line of G code reads G81 X5.0 Y6.0 Z0.0 R1.0 F1.0
> The first move that occurs will be a rapid traverse to X5.0 Y6.0 Z2.0
> Next move is a rapid traverse down to Z1.0 (this is the "Retract" height)
> Next move is a slow feed down to Z0.0 at F=1.0 units/min
> Final move is rapid move up to X5.0 Y6.0 Z2.0

G83 This is the same as G81 but allows you to "Peck" drill for chip removal in deep holes. You need to specify an additional "Peck" distance called Q. Q must be a positive value. This is the amount the drill is fed downward for each peck of drilling. See Drilldemo file for examples.

Example of G83 X5.0 Y6.0 Z0.0 R1.0 F1.0 "Peck Drilling":

> Assume that your last move completed was G00 X4.0 Y4.0 Z2.0
> Next line of G code reads G83 X5.0 Y6.0 Z0.0 R1.0 Q0.34 F1.0
> The first move that occurs will be a rapid traverse to X5.0 Y6.0 Z2.0
> Next move is a rapid traverse down to Z1.0 (this is the "Retract" height)
> Next is a slow feed down to Z0.66 at F=1.0 units/min
> Next move is rapid up to X5.0 Y6.0 Z1.0
> Next move is rapid down to Z0.66
> Next move is slow feed down to Z0.32
> Next move is rapid up to Z1.0
> Next move is rapid down to Z0.32
> Next move is slow feed down to Z0.00
> Final move is rapid up to X5.0 Y6.0 Z2.0

G90 This command is not required. It is assumed that all X, Y, and Z coordinates are absolute not relative. I, J, and K parameters for arcs however, are all assumed to be relative. These are the most common conventions for these parameters.

M03 Start spindle motor, pin 1 on printer plug goes to 0 volts.
M05 Stop spindle motor, pin 1 goes to +5 volts.
M07 Start coolant motor, pin 14 goes to 0 volts.
M09 Stop coolant motor, pin 14 goes to +5 volts.
M010 Start auxiliary motor, pin 16 goes to 0 volts.
M011 Stop auxiliary motor, pin 16 goes to +5 volts.
M47 This will automatically keep cycling the G code starting at the 1st line. It can be placed on any line in your G code file.

F Feedrate, use to change cutting feedrate for all subsequent moves, units are length per min. Applies to G01, G02, G03, G81 and G83 feed moves. The "F" must be on a line containing G00, G01, G02, G03, G81 or G83 command or it will be ignored.

All user comments in G code file should begin with a left parenthesis "(". The G code interpreter will then ignore that line.

Backlash is compensated for every time an axis moves in a changing direction. This occurs at every quadrant when a full circle is completed.

Some G code files use G1, G2 and G3 rather than G01, G02 and G03. To use these files you can use the Edit – Find – and "Replace All" command. Use to change all occurrences of "G" to "G0" and everything should work fine. It will change all instances of G1 to G01, G2 to G02 etc. in just a fraction of a second.

Some G code files also omit the G01 command when designating a long series of destinations. If no "G01" is on a line of G code then Quickstep assumes it is a cutting move and will then send out motor signals that go to whatever X, Y and Z destinations are specified on that line. That move will be at the default cutting speed specified in the "Setup table" or the last F cutting speed if so specified.

Excellon Drill File Import

Quickstep imports standard Excellon format files for drilling printed circuit boards. Importing these files is very easy from the File --- Import Excellon menu choice. By default Quickstep searches for all files ending with the **.th** file extension in the Quickstep directory. You select the file to open, and it loads into a window where it can be viewed.

You will see in the file view window that starting with the second line there are tool codes in the format of T1C0.025, T2C0.030 etc. T1 denotes the first tool to drill with. C implies a circular aperture i.e. a round hole made with a drill. The number, 0.025 specifies the drill diameter 0.025 inches. M47 line shows the name of the file you are viewing.

In the Excellon Drill viewing window are three input boxes where you enter the desired value of "Release Plane Height", "Drill Depth" and "Tool Change Height". The "Release Plane Height" is the height that the tool is raised to on the Z-axis for all rapid traverse moves going from hole to hole. "Tool Change Height" is the height on the Z-axis that the tool is raised to when a tool change is called. All moves are made at maximum traverse speed except for the downward drill move. By default, the downward drill speed is the same as specified in the "cutting speed" in the "Setup Table". If you want to do any last minute editing you can make changes in the viewing window then simply hit "Convert to G code". This converts the file into recognizable G code and loads it into the Quickstep G code viewing window for plotting on screen and execution of drilling activity. All the necessary G code up and down moves for drilling and tool changes are now visible for each hole location. The spindle is started and stopped also as required for tool changes. Of course you can start and stop the spindle manually if you don't have the motor control pin hooked up to your spindle motor control circuit.

You can plot the moves on screen and view tool path in simulation mode. You will see that when you plot in simulation mode, and in run mode, the program automatically stops for each tool change. This is where you make sure you insert the proper drill diameter then hit "Continue" check box to drill all specified holes with that tool.

To read the drill sizes and have the Excellon converter insert prompts in the G code to change drills you must define all the tool sizes and then follow that with an "M47" command. These should be the first few lines in your Excellon file such as shown below:

```
T1C0.035
T2C0.043
M47
```

DXF to G code ACE converter

This is a really nifty little file converter that is available as freeware compliments of Doug Yeager. If you don't already have your own favorite DXF to G code converter I highly recommend this one because it is so easy to use. It has been able to convert every drawing file I tried so far from Turbocad 7.0. If you want to convert text to G code you must break down the text into lines and polylines. In Turbocad 7.0 you use the "Explode" command. In AutoSketch you must use TrueType fonts then convert them to polygons and then explode the polygons into lines. It goes faster than you think though. I have tried several other converters and really like this one the best.

What is ACE converter?

ACE converter is an application that translates CAD (Computer Aided Design) files into tool path files. Most CAD applications today allow the user to export drawing files in various different formats. One of those formats is DXF (Drawing Exchange Format). This is done so that the same drawing can be read into any CAD application, no matter which one was used to create it. ACE converter reads DXF files and has the capability of converting those files into tool path or G and M code files.

ACE converter reads and supports DXF files that contain lines, polylines, arcs, circles, and points. Once a DXF file is read in, a variety of different code settings can take place using the Layer and Priority dialog boxes. The **Layer Dialog** box is used to set the properties for each layer of the DXF file. The **Priority Dialog** box is used to set the properties for each cutting priority. A cutting priority is one or a group of layers that are treated as a bundle and all share the same properties. For each priority there can be both pre-code user defined text and post-code user defined text. This allows the user to insert whatever commands they so chose into the code stream without editing the file after conversion.

The Layer Dialog Box

The layer dialog box is accessible by double clicking on any one of the layers listed in the layer column of the ACE converter window. This means, of course, that a DXF file must be loaded into ACE converter before accessing the Layer dialog box.

This dialog allows the user to:

> Turn the layer off. If this box is checked, no entities in this layer will be converted.
>
> Determine which direction to cut arcs and circles in this layer. If "Doesn't Matter" is selected, upon priority optimization the direction of each arc or circle will be automatically generated and optimized if the priority the layer belongs to is optimized. Otherwise, the arcs and circles default direction is counterclockwise.
>
> Set the Z offset of the layer. This setting just offsets the value of all Z values by the number specified. So if the drawing was made on the XY plane located at Z=0, the user could change this plane to be any Z value.
>
> Set the maximum depth per pass. This setting allows the user to determine if more than one pass needs to be made during the machining process. So for example, if your tool path is drawn at Z=-0.8 and the Max Z/Pass=0.25, then there would need be 4 passes generated in order the get to the proper Z depth of -0.8.

Determine the character to be used in place of the Z character in the DXF file. This can be very useful in 4 axis operations. So for example, if the user's fourth axis is named W, this character can be set to W and this layer would be generated only for the fourth axis control. This effectively allows the user to integrate his or hers fourth axis into tool path code yet not lose optimization by doing so.

Determine which Priority the layer belongs to. Each layer belongs to a priority. The priority properties define the layer properties further.

The Priority Dialog Box

The priority dialog box is accessible by double clicking on any one of the priorities listed in the priority column of the ACEconverter window. This means, of course, that a DXF file must be loaded into ACEconverter before accessing the Layer dialog box.
This dialog allows the user to:

Insert users code both before and after the priority is generated.

Optimize the priority. If checked, all the entities that belong to the priority will get optimized and coded in a systematic manner to make code most efficient.

Set the release plane. This value is used when generating the code for pen up, or non-cutting moves. It is the height at which the tool will release to in order to make the move.

Set the close enough value. This value determines whether the machine needs to release to the release plane to make a jump from one entity to another. A rule of thumb would be to set this value to half of the tool diameter in order to optimize the code further.

Multi-pass Milling

One of the less obvious features about the Ace converter that users frequently ask is "How can I do multi-pass milling ?" This is where the tool path repeats itself, but each time the cutter goes in deeper on the Z-axis. To generate the proper G code for this you need to use the "Z offset" feature of the converter. From the Ace Converter window select the dxf file that you want to convert to G code then double-click in the "Layer…Priority" column on the layer of interest. This brings up a window showing the "Z offset" text box. For multi-pass cutting you need to set the Z offset to a negative value. For example if you want to make 4 cuts, each 0.10" deep, then you set Z offset to -0.4 and set the Max Z/Pass to 0.10.

You also will need to set the release plane. Release plane is the Z height at which all rapid non-cutting moves are made. The default value of release plane is 3.00. To set this to a different value, from the G code converter window you need to double-click on the layer of interest in the "Priority" column. This brings up a window where you can enter the Z Release Plane value. When all these variables are set then hit the "Convert" to generate the G code steps. I have included an example of a four-pass G code file called Qsdemo1MP.txt for your viewing.

Someone asked me what the maximum number of G code steps are that the converter can handle. I don't know if there is a limit. I have been able to convert very complex dxf files that generated over 50,0000 steps of G code. I can't imagine anything much more intense than that.

Quickstep 4 Notes

This page is intentionally left blank for you to add your own personal notes. To add your notes to this page you can open it up with any word processor and then save it in its current location on your hardrive. To open it with your word processor locate the file named “qsnotes.rtf”. You will find the file in the Program Files/quickstep4 directory. It is in a format called rtf, that stands for “rich text format”. It is a universal text format that any word processor can open and save. Using this format allows you to add pictures and any font you choose. Just make sure you save the file in the same directory and with the exact same file name and it will always be available to view from the Quickstep help menu.

You can print or edit any of the Help files by opening them up with your word processor. They are all “.rtf” (rich text format) type files that allow you to paste bmp type images within the text for illustrations. The help files are all in the Program Files\Quickstep4 directory and are named: QSIntroV4.rtf , qscomm.rtf, qssetupV4.rtf, excellon.rtf, aboutace.rtf, qsnotes.rtf