G-Code Programming Tutorial

Brett Ian Balogh | bbalogh@saic.edu

Introduction: G-Code is a simple programming language that provides motion control instructions for a computer-numerically-controlled machine tool as well as control of other machine functions. G-code programs consist of blocks (or lines) that are made up of words. There are a number of valid words that can be used in G-Code programming, but for the purposes of the assignment, we will consider only a few. G-Code programs can be written in a simple text editor such as notepad.exe in windows or in textedit for mac and stored as a plain text file. The file extension is not important. Some examples are *.tap, *.gc, *.nc and *.txt.

G-Code Programming: G-Code programming is the process of creating instructions for a machine to move a tool along a path. The programmer uses the coordinates from a measured drawing of a part to determine the parameters/values to be given to appropriate words in the program. The path of the tool thus described follows along the lines and arcs that make up the geometry of the part to be produced.

Words: Words in a G-Code program begin with a letter address and are followed by a numeric parameter or value. There are many possible words in the G-Code specification, but we will be dealing with only a few:

Letter Address	Meaning	Value Range
F	Feed rate specification	F0.2 - F30.0
G	Preparatory function	G00 - G99
I	Arc X-Axis offset (G02/G03)	I±999.9999
J	Arc Y-Axis offset (G02/G03)	J±999.9999
M	Miscellaneous function	M00 - M99
N	Line number specification	N0 - N19999
S	Spindle speed specification	S60 - S4200
X	X-Axis specification	X±999.9999
Y	Y-Axis specification	Y±999.9999
Z	Z-Axis specification	Z±999.9999

Word Formatting: Spaces should not be inserted between the word and its parameters or values. For example, G00 is valid, and G 00 is not. For letter addresses G, M and N a leading zero is not required in the parameter. For example, G01 and G1 are both valid. I would suggest retaining leading zeroes for code readability. For letter addresses that take floating point values such as I, J, X, Y and Z, leading or trailing zeroes are not required. Examples of valid words are X10., X10.0, X.1 and X0.1 As a convention, however, always use a decimal point and at least one leading or trailing zero. For instance, X0.1 or X1.0.

Blocks: Blocks consist of one or more words terminated by the end of block (EOB) character. In our case, it is enter/return. Here is an example of a valid block:

Sequence Number	Preparatory Code	X-Axis Spec	Y-Axis Spec
N01	G00	X0.0	Y0.0

This block is line (or sequence) number 1. It instructs the machine to make a rapid linear move to the coordinate (0,0).

Block Formatting: Words in a block may be separated by space or tab characters, but this is not required. For example, N1G00X0.0Y0.0Z0.0 and N1 G00 X0.0 Y0.0 Z0.0 are both valid and equivalent. Use spaces to separate words, as some machines do not like tab characters, and not using spaces makes the code harder to read. Some words are modal, meaning that once they are called in a program, they remain in effect until cancelled. Consider the following example that defines the motion of a tool along the edges of a 1" x 1" square:

Verbose/Redundant Programming	Terse Programming
N01 G01 X0.0 Y0.0 F30.0	N01 G01 X0.0 Y0.0 F30.0
N02 G01 X1.0 Y0.0 F30.0	N02 X1.0
N03 G01 X1.0 Y1.0 F30.0	N03 Y1.0
N04 G01 X0.0 Y1.0 F30.0	N04 X0.0
N05 G01 X0.0 Y0.0 F30.0	N05 Y0.0

The terse programming example shows that G01, X, Y and F are modal. The function G01 prepares the machine to make a straight line to the point 0,0 at a feed rate of 30 IPM. All subsequent coordinates are considered end points for the next line and the motions are all done at the feed rate specified in block N01.

Descriptions of Words:

(F): F specifies the feed rate for machine motion in inches per minute (IPM). It is used by preparatory words such as G01, G02 and G03. The smallest increment in the feed rate is 0.1 IPM, and the feed rate should be specified as a decimal value. If the feed rate is a whole number such as 30, express it as F30.0. The feed rate may be changed at any point in the program by adding a F-word to the end of a block for which one of the preparatory words above is in effect. Example:

```
N01 G01 X0.0 Y0.0 F30.0
N02 X1.0 Y1.0 F10.0
```

(G00): G00 instructs the machine to move from its present position to the coordinates supplied in the block at a pre-set maximum rate for the machine. This word is usually used for positioning moves, or moves where the machine is not drawing/cutting.

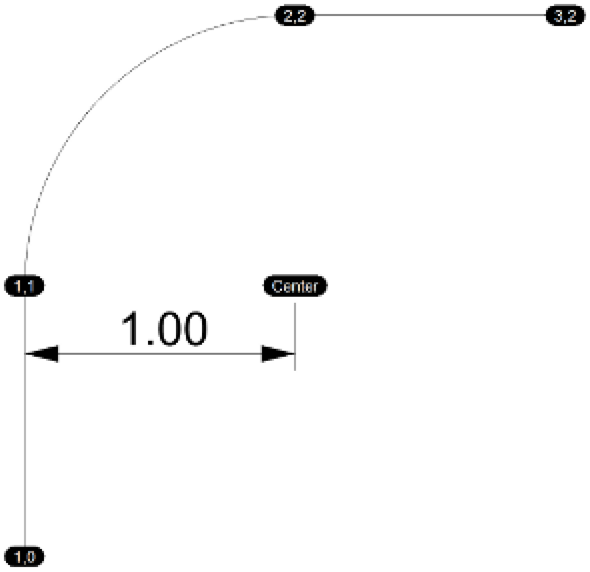
```
N01 G00 X0.0 Y0.0
```

(G01): G01 instructs the machine to move from its present position to the coordinates supplied in the block at a feed rate specified by the F-word at the end of the block. This word is used for linear drawing/cutting motions when the tool is engaged with the material. Consider the following blocks:

```
N01 G00 X0.0 Y0.0
N02 G01 Z0.0 F20.0
N03 X1.0 F10.0
```

These blocks describe a ‘rapid’ linear move to (0,0), a move at 20 IPM downward to (0,0,0) and then a 10 IPM move to (1,0)

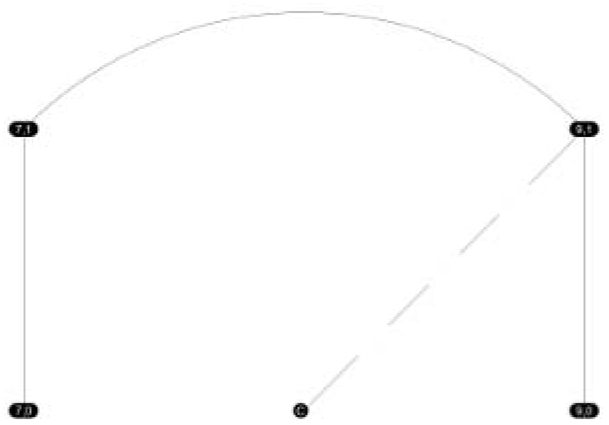
(G02): G02 instructs the machine to move at feed rate in a clockwise circular arc from its current position to an end point specified in the block. The radius of the arc is determined by words I and J that represent the X- and Y-coordinates of the center of the arc/circle relative to the start point of the arc. Since the coordinates of the center of the arc/circle are relative, I and J could be positive or negative depending on the direction to the center from the start point. The machine can interpolate a full circle in one block if the end point of the circle is the same as the start point. Look at the example below:



The illustration depicts a linear move from (1,0) to (1,1) and then a 1” radius arc move to (2,2) and then another linear move to (3,2). In this case, the arc center is at (2,1). The corresponding blocks would be:

```
N01 G01 X1.0 Y0.0 F30.0
N02 Y1.0 F10.0
N03 G02 X2.0 Y2.0 I1.0 J0.0
N04 G01 X3.0
```

(G03): G03 instructs the machine to move at feed rate in a counter-clockwise circular arc from its current position to an end point specified in the block. The radius of the arc is determined by words I and J that represent the X- and Y-coordinates of the center of the arc/circle relative to the start point of the arc. Since the coordinates of the center of the arc/circle are relative, I and J could be positive or negative depending on the direction to the center from the start point. The machine can interpolate a full circle in one block if the end point of the circle is the same as the start point. Look at the example below:



This illustration depicts a linear move from (9,0) to (9,1) then a counter-clockwise 1.414” radius arc move to (7,1) and then a linear move to (7,0). The center of the arc is at (8,0). The corresponding blocks would be:

```
N01 G01 X9.0 Y0.0 F30.0
N02 Y1.0
N03 G03 X7.0 Y1.0 I-1.0 J-1.0
N04 G01 Y0.0
```

(I): I is the X-position of the center of a circle or arc relative to the start point. It can be a positive or negative value. See the descriptions of G02 and G03 for examples.

(J): J is the Y-position of the center of a circle or arc relative to the start point. It can be a positive or negative value. See the descriptions of G02 and G03 for examples.

(N): N is the line or sequence number specification. It is not necessary to include in a block, but can aid the readability of a program. I use it here so I can refer to specific lines.

(S): S is the spindle speed specification. It instructs the machine to rotate its spindle (the part that holds the tool) at a specific speed in revolutions per minute, or RPM. We will not use this word in our assignment, but it will be important when cutting.

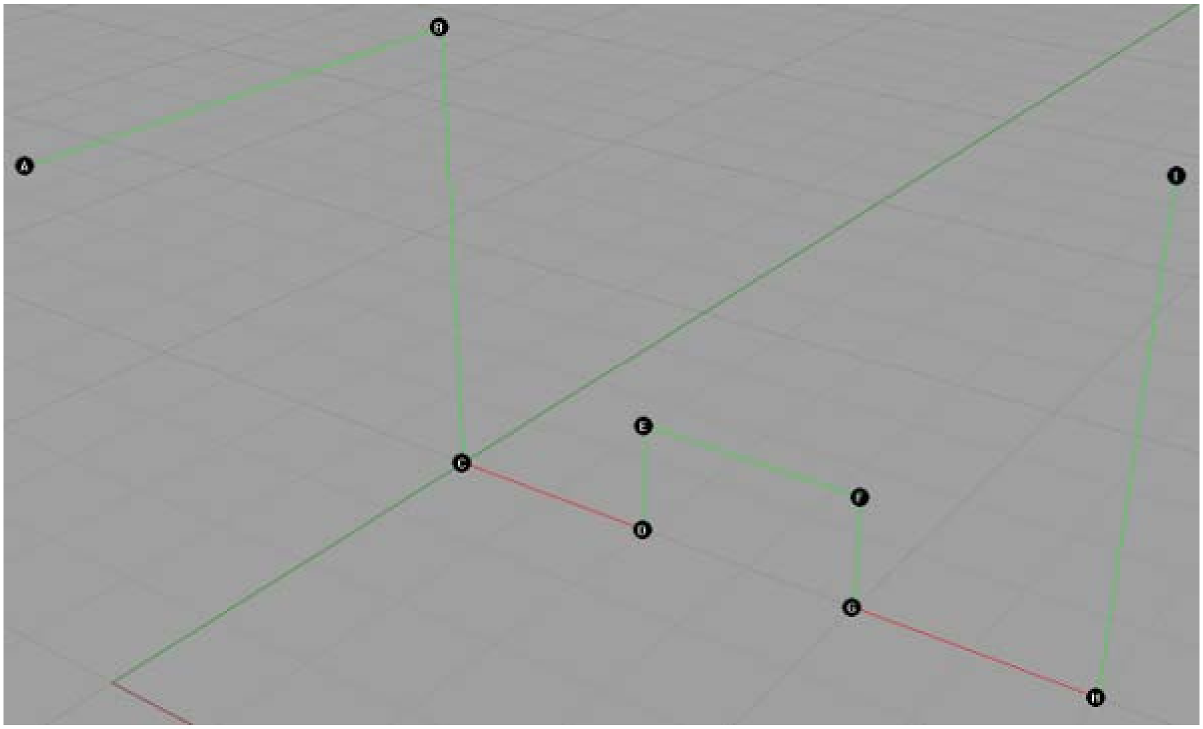
(X): X specifies the X-axis component of a position used with words such as G00, G01, G02 and G03

(Y): Y specifies the Y-axis component of a position used with words such as G00, G01, G02 and G03

(Z): Z specifies the Z-axis component of a position used with words such as G00, G01, G02 and G03

Motion Logic: Most of the examples above do not take into consideration the difference between positioning moves (G00) and cutting/drawing motions (G01, G02 and G03 at a specified feed rate). If you are programming to draw/cut something, there are motions you want to do in contact with the material and others above it. If you're drawing a picture, you must lift the pen and move to another position to make a line or arc somewhere else if you don't want to see the line or arc that connects them. Pick a positive Z-position above your material to make all moves that don't involve engaging the material. Z=0.25 is a good value.

For example, let's draw/cut two 0.5" lines separated by a 0.5" space. The pen motions are depicted below:



The red lines are cut/draw moves and the green lines are positioning moves. The points in the path are as follows:

Point Tag	X	Y	Z
A	0.0	0.0	1.0
B	0.0	1.0	1.0
C	0.0	1.0	0.0
D	0.5	1.0	0.0
E	0.5	1.0	0.25
F	1.0	1.0	0.25
G	1.0	1.0	0.0
H	1.5	1.0	0.0
I	1.5	1.0	1.0

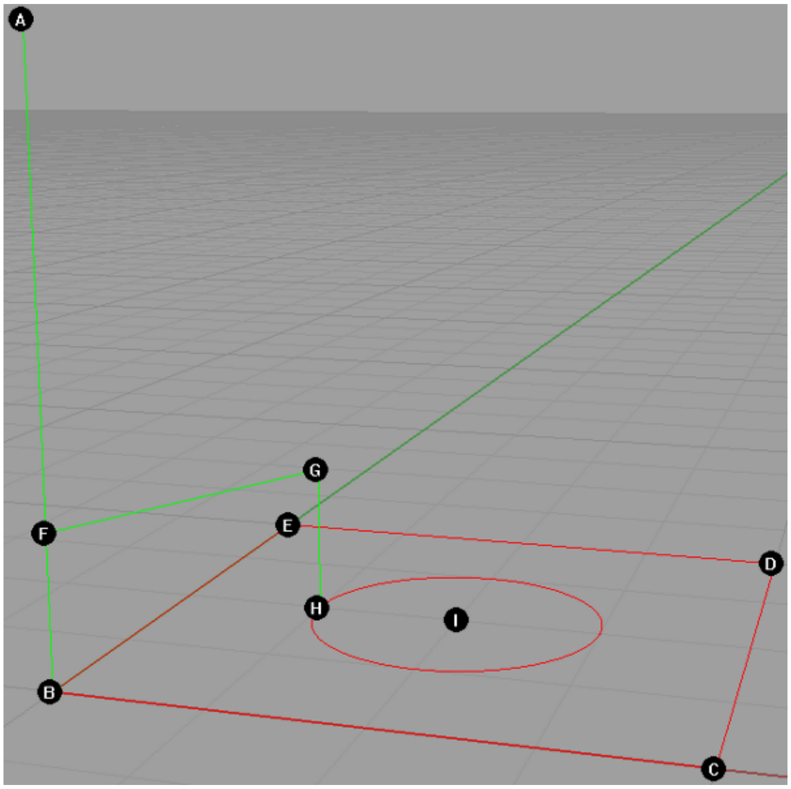
The corresponding blocks would be:

```
N01 G01 X0.0 Y1.0 F30.0
N02 Z0.0 F10.0
N03 X0.5
N04 Z0.25 F30.0
N05 X1.0
N06 Z0.0 F10.0
N07 X1.5
N08 Z1.0 F30.0
```

These blocks program a linear move from the initial position of the tool (currently at (0,0,1)) to (0,1,1) at a rapid feed rate of 30.0 IPM. The tool then moves to engage the material with a linear move to (0,1,0) at a feed rate of 10.0 IPM. The tool, now drawing/cutting, moves to (0.5,1,0). The tool lifts at 30.0 IPM to a height of 0.25 inches and then over another 0.5 inches to (1.0,1.0,0.25). The tool drops to (1.0,1.0,0.0) at a feed rate of 10.0 IPM and begins another line to (1.5,1.0,0.0). This is the end of drawing/cutting, so lift the tool at 30.0 IPM to a convenient height of (1.5,1.0,1.0)

Example:

I am going to draw a square with a full circle at its center. The image below shows the lines/arcs to be drawn in red. The green lines are positioning moves.



Point Tag	X	Y	Z
A	0.0	0.0	1.0

B	0.0	0.0	0.0
C	1.0	0.0	0.0
D	1.0	1.0	0.0
E	0.0	1.0	0.0
F	0.0	0.0	0.25
G	0.25	0.5	0.25
H	0.25	0.5	0.0
I	0.5	0.5	0.0

A pseudocode description of the steps necessary:

```
Linear positioning move from A(0,0,1) to B(0,0,0) at a feed rate of 30 IPM
Linear drawing/cutting move from B(0,0,0) to C(1,0,0) at a feed rate of 10 IPM
Linear drawing/cutting move from C(1,0,0) to D(1,1,0)
Linear drawing/cutting move from D(1,1,0) to E(0,1,0)
Linear drawing/cutting move from E(0,1,0) back to B(0,0,0)
Linear positioning move from B(0,0,0) to safe Z at F(0,0,0.25) at a feed rate of 30 IPM
Linear positioning move from F(0,0,0.25) to G(0.25,0.5,0.25)
Linear positioning move from G(0.25,0.5,0.25) to H(0.25,0.5,0.0)
Full circle drawing/cutting move from H(0.25,0.5,0.0) back to H with a center at I(0.5,0.5,0.0) at a feed of 10
Linear positioning move from H(0.25,0.5,0.0) back to G(0.25,0.5,0.25) at a feed of 30
```

The resulting G-Code:

```
G01 Z0.0 F30.0
X1.0 F10.0
Y1.0
X0.0
Y0.0
Z0.25 F30.0
X0.25 Y0.5
Z0.0
G03 X0.25 Y0.5 I0.25 J0.0 F10.0
G01 Z.025 F30.0
```