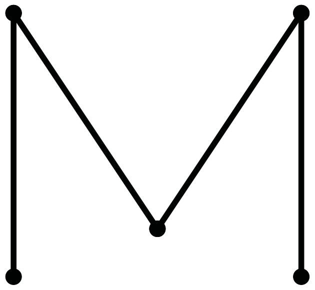
Fundamentals of G-Code

G-Code programming is a very simple programming language. When we were kids, we used to and may still do connect the dot puzzle games. G-code works ***exactly*** on the same principle. It takes two simple concepts to understand G-code programming, ***connect-the-dots*** and the ***number line***.

Connect-the-dots

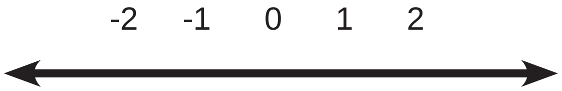


We can draw something very simply by connecting the dots.

We do not think about it, but we are creating something tangible in the 2-axis world.

The Number Line

The number line is simply the measurement of units. Let's look at this number line:



In a straight line numbers either get bigger or smaller from Zero. Machining, along with G-Code programming, uses both sides of Zero.

Signs +/ -

The Plus (+) sign and the Minus (-) sign are very important in machining. In machining we can also call them Positive or Negative respectively. We use these signs in two situations; location and direction.

Location

They signify what side of Zero a number is on, we can say this is a location indicator; Left for Minus and Right for Positive respectively from Zero.

Direction

These signs also are used as a tool to tell us which direction to move; Left for Minus and Right for Positive.

Addition

Let’s talk about simple addition for a second. When you add or subtract numbers you get an answer. The building block of programming is simple addition. If you can not think conceptually about which side of Zero you are on, or which direction you need to go because of a negative number, just add or subtract.

Ok, let’s look at some simple examples of positive and negative numbers with simple addition and subtraction.

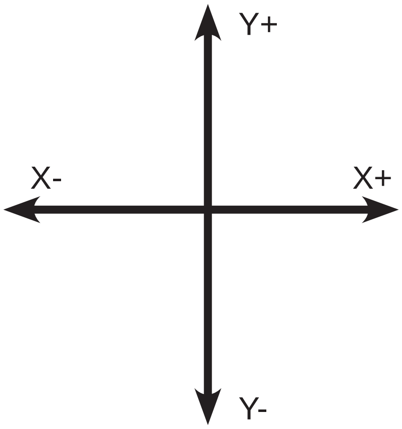
1+1=2	-1 + -1 = -2	-1 + 4 = 3
1+2=3	-1 + -2 = -3	-4 + -1 = -5
2+2=4	-2 + -2 = -4	-6 + 2 = -4

For simplicity, always add numbers. Sometimes we get very confused subtracting negative numbers.

First.

CNC Milling Machining Centers use three axis and sometimes more. Obviously this is because the world is 3-dimensional.

Don't be scared, we will start with two. We will start by thinking in terms of that old graphing we did in algebra classes. Just think of that same number line as before with another line running perpendicular.

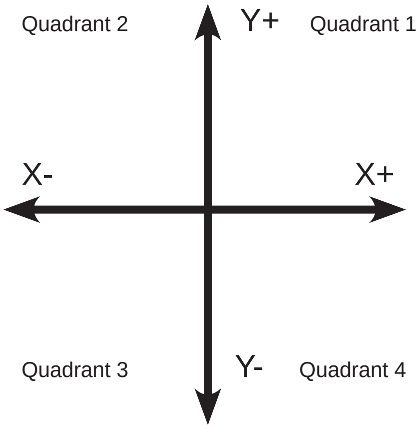


These lines are all you need to get started programming. But now we have to define them. The one that goes horizontally is "**X**" and the one vertically is "**Y**".

If you can understand this simple concept, and understand that Zero has two sides, then you are on your way. We use this in programming just simply for directional positioning in a graphed area.

Quadrants

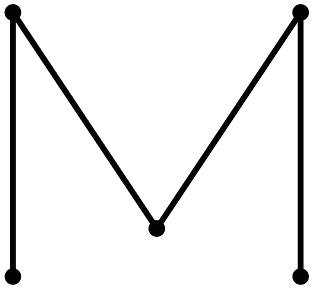
Quadrants are something to make note of simply on the basis of location. Because a machine tool uses lead screws as a means to move the table along a straight line, identifying the quadrants are important because we can identify which lead is being used to move the work piece.



Quadrants also help in identifying the signs of the respected axis in use.

Quadrant 1 = X+ Y+	Quadrant 3 = X- Y-
Quadrant 2 = X- Y+	Quadrant 4 = X+ Y-

2 Axis Fundamentals
Let us start by drawing that ***connect-the-dots*** of the Letter "**M**". If you right the letter on paper, you are actually writing it in some sort of units across a 2 dimensional surface.

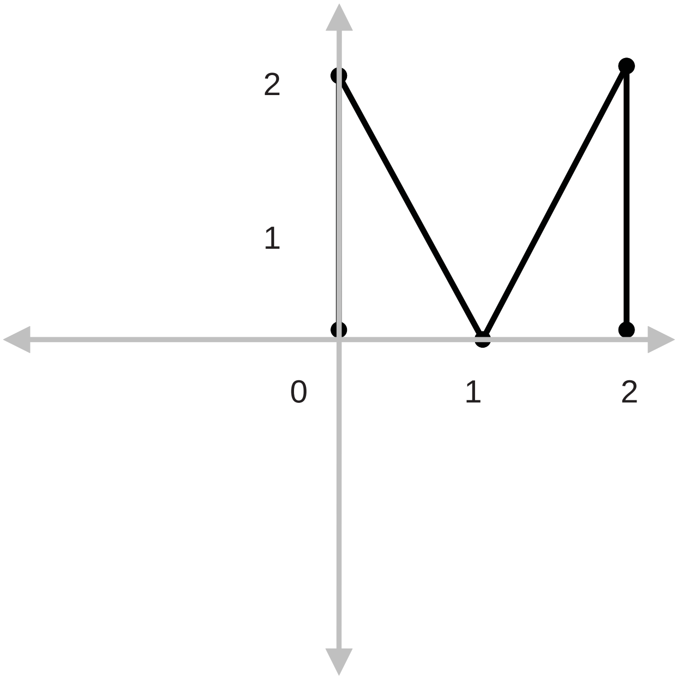


How would we describe this letter in Graphical terms applying it to our two axes?

- Move straight in the Y axis in a *positive* direction so many units.
- Then move in a *positive* X and a *negative* Y direction so many units.
- Then move in a *positive* X and a *positive* Y direction so many units.
- Then move in a *negative* Y direction so many units.

Second.

Let us take this letter "M" example and combine it with units. We will identify this scale of inches.



We will start at Zero.

- Move to Y2.
- Move to X1.Y0
- Move to X2.Y2.
- Move to Y0

This is the basics of programming. The combination of the right G-code in these simple lines would create the Letter "M" on a CNC Machining Center.

Absolute versus Incremental

Absolute programming is programming mode that is a reflection of each number read is from part or machine Zero.

Incremental programming is a programming mode that is a reflection from the current position.

In our Letter “M” example, the first example defined the letter from moving from a start position then on to the next position by so many units. This would be an example of *incremental* positioning. We said to move the next position from that current position.

In the second Letter “M” example adding units, we actually moved to locations with from a defined Zero position. This would be an example of *absolute* programming.

Summary

What we get from 2 Axis Fundamentals is that G-code is nothing more than numbers. What we do with G-code programming is define codes to let the machine tool know what the programmer wants to do with those numbers.

With the use of different codes we can do just about anything imaginable. These are such things like make a straight line or interpolate a circle or anything in between.

The following sections will discuss the different commonly used G-codes and M-codes along with the different complimentary aspects to assist those codes.