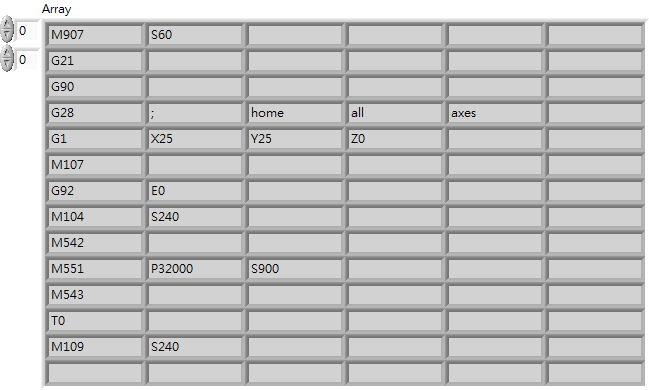
Overall Flow:

1. Read lines from text file. (Right now I read all lines at once, but this could be a problem for very large text files. In the future I might change the parser to line-by-line.)
2. Convert to array of strings.
3. Throw out the comment lines that start with “;”. Might add semicolon parser for middle of text as well.
4. Define the tokens to be searched for. A token is the letter that precedes the number, ie. G, M, X, Y, etc.
5. The “Scan String for Tokens” within the loop will separate the components of each line into the format as shown:

So for instance, line 1 will have 2 elements, line 2 will have 1 element, line 5 has 4 elements, etc.

1. For each element, write the parameter into the command cluster according to its letter.
2. For lines that have more than 1 code (i.e. G20 G40 G60 M3), an additional command cluster is inserted into the array (unless it is already the first code in the line, because there will be an insert function at the end of each line. See #8.) Right now it detects G/Ms, the case can be duplicated for other action codes.
3. Insert for each end of line, because there may be lines such as “X0 Y0” with no action code.

Right now, the following Gcodes are implemented for 4-axes:

G00: Rapid Positioning

G01: Linear Interpolation

G02: Circular interpolation, CW

G03: Circular interpolation, CCW

G04: Dwell

G90: Absolute Programming

G91: Incremental (Relative) Programming

G92: Reset Position