

Application Note #54: Infinite Looping

Functional Description

Skeleton Code Name: Infinite Looping Program

Software: Q Programmer

Hardware Used: ST10-Q Stepper Drive

Hardware Compatibility: Q Program Capable Drives Only

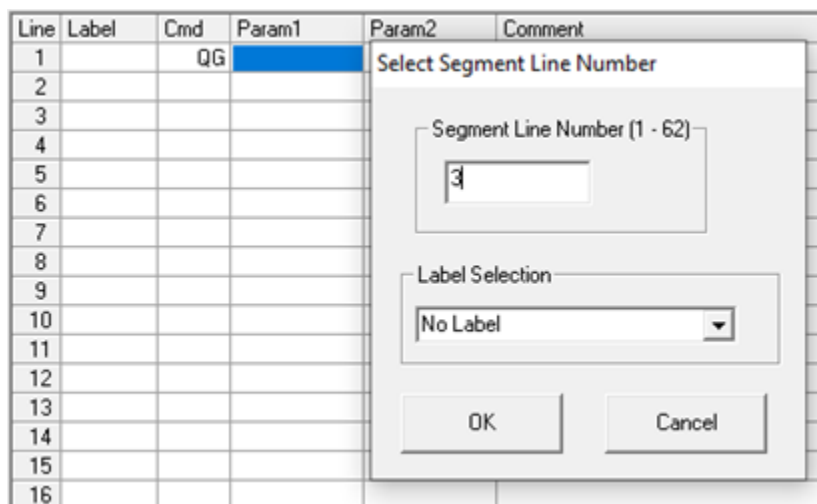
Reference: [Host Command Reference Manual](#)

- Commands
 - o QG – Queue Goto
 - o NO – No Operation

Goals

Goal 1: Familiarize oneself with Queue Go (QG) command

- a. For its Param1, it takes the # of the line we wish the program to jump to within our current program segment. This is NOT a conditional statement. The program will automatically jump to the desired line after command execution.
 - i. Upon filling in Param1 for QG, Q Programmer will automatically create a label that is assigned to the line we wish to jump to. Param1 will then match the Label that was assigned. Q Programmer does this so that line jumps are tied to a label and not a line #, because sometimes when we edit a program, we end up shifting out code up or down. Allow Q Programmer to assign its own labels. See the following images:



Application Note #54: Infinite Looping



Line	Label	Cmd	Param1	Param2	Comment
1		QG	#LABEL1 ...		
2					
3	LABEL1				
4					
5					
6					
7					
8					
9					

- ii. The above program will jump from line 1 to line 3 (Label 1) as soon as the program runs. The program will then execute from line 3 (Label 1) and continue executing from there. It will skip line 2 entirely.
- iii. NOTE: Q Programmer does not accept empty command lines. In this case, line 2 would be invalid. It was left empty for show only. If this were to execute properly, we would have to use a NO (No Operation) command or any other command on line 2.

Goal 2: Execute a simple program that loops indefinitely

- a. Based on Goal 1, we know that a QG command will allow us to jump to any line in our current program segment and continue executing from there. That means we can jump to lines before and after our QG command. Then, the program will execute from that line. Knowing that, what would happen if we jumped to line before our QG command? We'll call this line Label1 for now. The answer is that the program would jump to Label 1 and start executing again from that Label 1, until it reached QG again and jumps back to Label 1 again. This then creates our infinite loop!

Line	Label	Cmd	Param1	Param2
1	LABEL1	WT	3.00	
2		WT	3.00	
3		WT	3.00	
4		WT	3.00	
5		QG	#LABEL1 ...	
6				
7				

- b. The above program will execute four timers sequentially, each of three seconds, at the end of which it will loop back up and restart all over again, indefinitely!
- c. With this concept in mind, you could potentially start looping more complex programs that tests inputs, set outputs, and execute motion within loops.

Application Note #54: Infinite Looping



[Click here for the Sample Code.](#)

Try it out today!

If you have questions about this or any other application, please reach out to our Apps Engineering Group for any assistance at 1-800-525-1609 or support@applied-motion.com.