

# Programming In QC5000

---

The QC5000 is programmable, it can store a series of steps, and then prompt a user to perform the same steps at a later date. Programs are useful if you will be measuring the same type of part often. Every program that you will create with the QC5000 is tied to a specific part; there is one program per part, and a program is never independent of a part. The QC5000 allows you to record programs, save programs, and run saved programs.

**Recommended:** This section contains a sample program. After you have completed this section, you should perform the sample program straight through at least once on your QC5000.

---

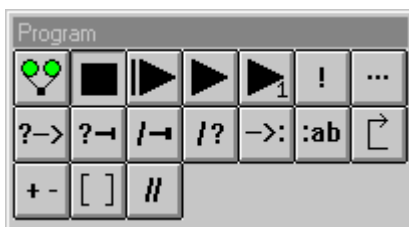
## Recording A Program

You will tell the QC5000 exactly what you want it to do at each step, and it will play the program back in the same manner. Here we will create a small program, and we will save the program (this section will use the QC5000 demonstration part, but any multi-feature part may be substituted). **Datum Magic does not work during programming.**

## The Program Toolbar

Before we begin programming; we should review the Program toolbar. If this toolbar is not displayed, select VIEW > TOOLBARS... from the main menu. Select "Program" then select the "Show" button. Click "OK" to continue.

The Program toolbar:



Select an icon in the toolbar above to learn about its functionality.



**Record/Edit Program** — Places the QC5000 into Record/Edit Mode. All actions performed from this point will be applied to the program. To exit this mode, select the Pause Program icon (see below)



**Pause Program** — Stops a currently running program. The Program window will highlight the last executed step performed in the program before it was paused.



**New Run** — Begins running the program from the first step, regardless of which step was selected in the Program window.



**Run Program From "Current" Step** — Begins running the program from the "currently" selected step.



**Run Just "Current" Step** — Executes only the "currently" selected step, then pause the program after execution.



**Toggle Break Point** — Attaches a marker to the currently selected step. This marker will instruct the program to stop when this step is reached in the program. To remove this marker, simply select this step again, then select the "Toggle Break Point" icon to clear it. Break Points can be placed on more than one step within a program.



**Edit Steps** — Allows you to edit the currently selected step. The options you will be presented with will depend on which functionality is associated with the currently selected step (e.g., manually change point coordinates, toggle the display of a position indicator, modify probe path data, etc..)



**If-Goto Statement** — Allows you to create a conditional argument for the currently selected step. If the selected step is executed during the program, and the conditions you have placed on this step are met, then the program will "goto" a different step within the program.

**Example**, if the value of the X coordinate exceeds a defined value, goto a specified step within the program.



**If-Then Statement** — Allows you to create a conditional argument for the currently selected step. If the selected step is executed during the program, and the conditions you have placed on this step are met, then perform the function you have specified. If the conditions have not been met, then continue running without change.

**Example**, if the value of the X coordinate exceeds a defined value, apply an offset to the value to X before proceeding to the next step.



**Else Statement** — Used in conjunction with either an If-Then statement or an Else-If statement. If the conditions set by either the If-Then statement or the Else-If statement have not been met, then skip the steps that the If-Then or Else-If statement would have executed and execute the steps after the Else statement.

**Example**, an **If-Then** statement within a program instructs the QC5000 to execute a second inspection if an inspected distance is less than 5mm. The next line in the program sets an **Else** statement that instructs the QC5000 to stop executing the program if the same inspected distance is greater than 6mm. If the distance returned during the programs execution is 4.7mm, then the QC5000 will execute a second inspection. If the distance returned during the programs execution is 6.2mm, then the QC5000 will stop the program. If the distance returned during the programs execution is 5.4mm, then the QC5000 continue the program skipping both the **Then** condition steps (second inspection) *and* the **Else** condition steps (stop program).



**Else-If Statement** — This statement is the same as the **Else** statement, except that an additional condition needs to be met to see if the statements after the **Else-If** are performed.



**Goto Label** — Allows you to direct the program to proceed to a "labeled" section within the program. When selected, you will need to enter the name of the label you want to go to. You must enter the name exactly as it appears in the destination label (e.g., if the label step appears as Label "Find-Diameter-1"..., then you would enter "Find-Diameter-1" without the quotes). For more information.



**Label** — Allows you to place a labeled marker within the program. This marker can divide your program into sections. The name you enter must not contain any spaces or special characters. These sections can then be switched to during the program, when used in conjunction with the "Goto lable" option.

**Example**, You have a program that has the following:

- Steps 10 through 18 — find the center of Circle 1
- Steps 19 through 27 — find the center of Circle 2
- Steps 28 through 36 — find the center of Circle 3

You insert Labels at the beginning of each process as follows:

- Step 10 - Label "Find-Diameter-1"...
- Steps 11 through 19 — find the center of Circle 1
- Step 20 - Label "Find-Diameter-2"...
- Steps 21 through 29 — find the center of Circle 2
- Step 30 - Label "Find-Diameter-3"...
- Steps 31 through 39 — find the center of Circle 2

Elsewhere in the program, you can insert a step that uses the "Goto Label" option to send the program to one of these locations (Find Diameter 1, Find Diameter 2, or Find Diameter 3).



**Loop** — Allows you to *repeat a selection of steps* (loop) a selected number of times (e.g., repeat program steps 4 through 10 for 5 loops then continue on to step 11.)



**Offset Positions** — Allows you to manually enter a coordinate that will offset the current coordinate.

**Example**, suppose the X coordinate in Step 3 is 1.03621. In Step 4 the X coordinate moves to 1.13621. In Step 5 you want the X coordinate to be the same value it was in Step 3. You would then: select Step 4, select the Offset Positions icon, then input the value you wish to offset by (in this example, X = - 0.10000). A program step will be added after Step 4 that will offset the coordinate. Step 5 (now Step 6) will begin with the same X coordinate that Step 3 began with.



**SuperStep** — Allows you to group a selection of steps into a *collapsible group* (SuperStep). This can assist programmers by organizing and shortening the display of large programs in the Program window.

**Example**, a program uses Steps 12 through 24 to plot a series of points that are required to determine a specific coordinate. To reduce the space that all of these steps will take up in the Program window, select Steps 12 through 24 with the mouse, then select the SuperStep icon. You will be prompted for a name to identify this SuperStep. After entering a name and selecting OK, the Program windows will update to show two new steps. Step 12 is now the SuperStep with the title you specified. You can collapse this group by selecting the minus (-) symbol in the third column (default), then select the plus (+) symbol to expand the group. Steps 13 through 25 (originally 12 through 24) are the program steps you selected to group. Step 26 indicates the end of the SuperStep.

**To delete a SuperStep**, select it in the Program window, then either, select Edit > Delete from the main menu or select the delete key on your keyboard.



**Program Comment** — Allows programmers to enter a note into the program for future reference. This note will not perform a function.

**Example**, at Step 12, you want to enter a note that explains the programs current functionality (e.g., finding the radius of the first circle in a program), you would select Step 12, select the Program Comment icon, then enter the comment "Finding Radius #1" (without quotes) into the Enter Program Comment dialog box, then select OK. Step 12 will now display the comment "Finding Radius #1" and the // symbol will