WinCNC Controller
PC Based Motion Control

Microsystems World CNC LLC
PO Box 2547
14 East Lincoln Street
Buckhannon WV 26201

Phone: (304) 472 7206 Fax: (304) 472 9638

# Creating a macro in WinCNC

## Macro File Configuration

First and foremost, the user needs to determine what the macro is going to do. Is the macro going to simply call a file? Or is the macro going to call a series of commands and then pause the CNC machine waiting for user input? There are a number of commands already build into WinCNC. These pre-set "M" codes can be referenced starting on page 99 of the user's manual. The CNC.MAC file allows the user to create his or her own custom commands not already built into WinCNC. If macros are not used, a blank text file should be created in the installation directory, and named CNC.MAC. This may already have been done.

The simplest example is a macro that calls a specific file every time. Let's assign M3 to call testfile.tap. In the CNC.MAC file, add the following line:

| | |
|---|---|
| M3 = m98 testfile.tap | [CALLS "testfile.tap" FILE] |

Now save the CNC.MAC file. "M98" is the command you use to "call" a file. If WinCNC is already open, close it and re-start it. If it is not open yet, open WinCNC, and type m3 in the command line, then hit enter. If the following error is received: "file not found: testfile.tap", then the exercise was done correctly. By typing m3, the program is attempting to call the "testfile.tap". Any working .tap file may be put in place of "testfile'tap" if further functionality would like to be tested.

If the user wanted to create shortcut commands to turn auxiliary output channel 2, which controls the spindle on and off power, the entries to the CNC.MAC file might be as follows:

| | |
|---|---|
| M3 = M11C2 | [SPINDLE ON] |
| M5=M12C2 | [SPINDLE OFF] |

**NOTE**: Macro commands defined in the CNC.MAC file can only contain a single command.

If multiple commands are needed within a macro, you can define a macro that calls a subroutine. Using a subroutine, you can call multiple commands. The following example shows how to set a macro to call a subroutine and an example of what that subroutine would contain. This example demonstrates a macro that calls a subroutine that will turn on a spindle and wait for it to get up to speed before moving.

Macro Define:
(The following would be added to the CNC.MAC file)

| | |
|---|---|
| M3=M98 SPIN.MAC | [SPINDLE ON] |

Subroutine:
(The following would be named spin.mac and saved as a .mac file inside of the WinCNC folder)

| | |
|---|---|
| M11C1 | [SPINDLE ON] |
| M17C10 | [WAIT FOR SPEED UP] |

Another useful feature of the CNC.MAC file is to disable unwanted M-Codes output by some CAD/CAM packages. WinCNC will ignore disabled codes. Following is an example of how to disable an unwanted M2 code.

M2=[
(The square bracket "[" makes anything after it no more than a simple comment)

Also the CNC.MAC file allows the user to override command names which have already been built into WinCNC. Once you override a built in command you will have to use a tilde (~) any time you want to use the built in function. One example of this would be if a user wanted to use the command G28 to return to 0 on axis XYZ in that order. The CNC.MAC file entry to accomplish this would be:

Macro Define: (in the CNC.MAC file)

| | |
|---|---|
| G28=M98 HOME.MAC | |

This sets the WinCNC G28 command to call the macro instead of its built in function.

In the HOME.MAC file the tilde (~) tells the controller that the built in command has been renamed, and must override the renamed assignment and run its built in function. Keep in mind that by declaring this macro the built in G28 command will be disabled and cannot use its built in function without calling it with a tilde (~). The HOME.MAC file would look similar to this.

Subroutine:
~G28X [Return to 0 on X axis]
~G28Y [Return to 0 on Y axis]
~G28Z [Return to 0 on Z axis]

Note: The .MAC extension of the subroutine file names is NOT required. That extension can be used if desired, but was used here only as an example.