Auto Tool Zero Method for XYZ Axis Using Mach3 - Part 1

by **airnocker** » Sat Aug 21, 2010 12:44 am

Are you looking for a simple, sequential method to set the work piece XY axis to “0,0” and the work surface Z axis “0” location on your CNC mill? The method I devised makes it simple and easy and requires only one touch plate device, the existing Auto Tool Zero button on the Mach3 screen and my custom Visual Basic code for the button macro. In addition, the code includes the following execution user prompts: Confirm touch leads are connected to continue (y/n)? where "n" aborts entirely, Zero X & Y (y/n)? where "n" skips to Z confirmation, Zero Z (y/n)? where "n" aborts and ends macro.

Prerequisites
First, you should have your home position switches and travel limit switches setup and operational because the first requirement to utilize this method is that you Home your X,Y and Z axis.

Second, the tool should accurately move in all three axes without any positional errors or inaccuracies.

The All-In-One XYZ Touch Plate
The touch plate used consists of a 90 degree piece of 1/8” x 1 1/4” angle aluminum that measures 1 3/4” long. It is important that the end of the angle aluminum that will serve as the XY base be a clean, smooth perpendicular cross-cut.

The inside faces of the aluminum should be checked to ensure they are square with each other and flat. A hole is drilled and tapped for a short, 8-32 machine screw as shown in Figure 2. Pieces from a 1/16” copper clad circuit board are cut to match the dimensions of the inside faces of the aluminum angle. The circuit board material must make a butt joint with each other in the corner. The circuit board material is Super Glue'd with cyanoacrylate glue to the inside faces of the aluminum angle and held by compressions clamps for an hour. Using a digital micrometer, the thickness of the Z-axis touch plate side is then measured in multiple places to note any thickness variations and written on the face of the copper with a Sharpie pen. Mine is uniformly .180" thick.

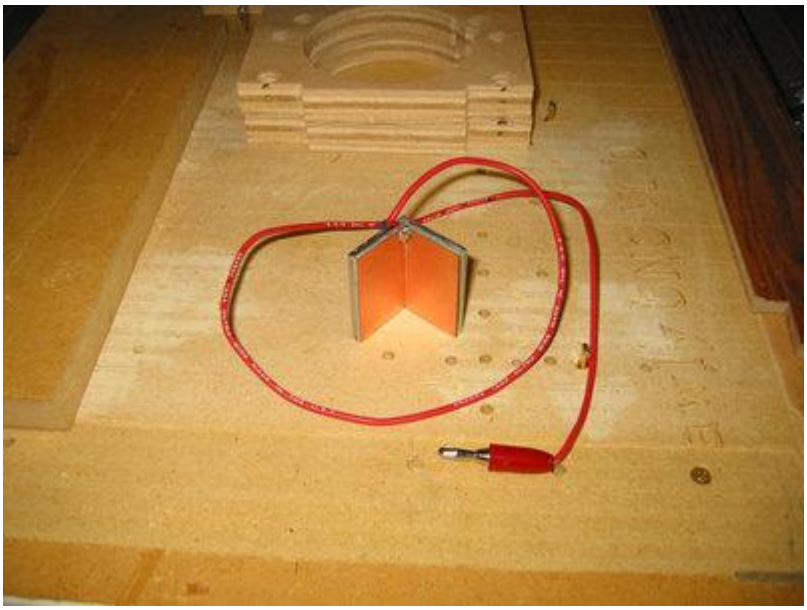


Figure 1

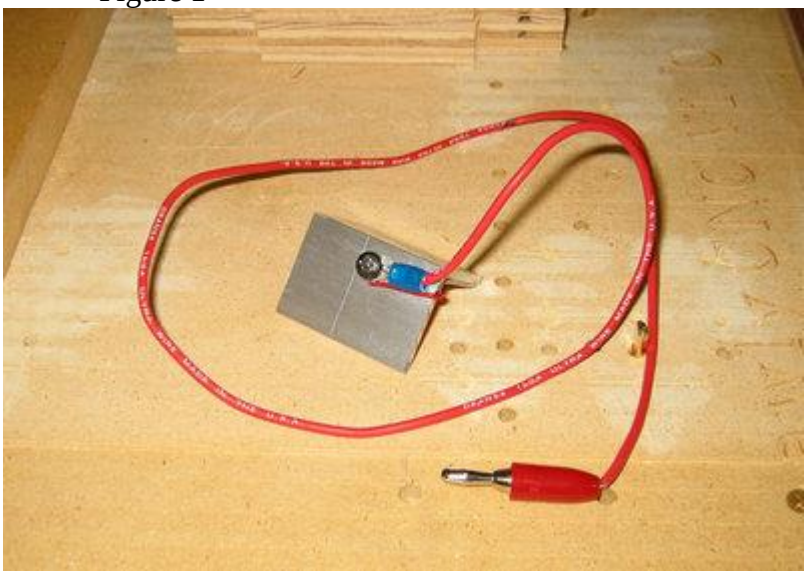


Figure 2

As Figure 1 above illustrates, the top, inside corners of the copper faces are soldered together. A small gauge wire is also soldered into the corner weld and bent around to the opposite side, stripped and tinned on the end. The Touch Plate signal lead is made from 16 Gauge silicon wire terminated with a Banana plug on one end and a lug crimp connector on the other.

The small wire is clamped between the lug crimp connector and the aluminum backside by the screw.
Figure 1 and 3 illustrates the “standing” position of the touch plate for XY zeroing.

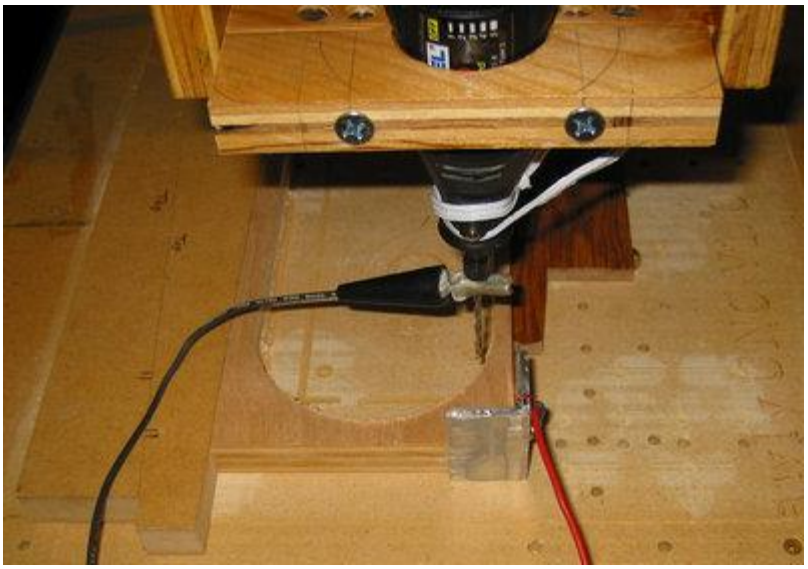


Figure 3

Figure 3 shows the XY touch plate against the intended work piece's XY 0,0 corner while Figure 4 shows the Z touch plate resting on the work piece.

Figure 4 illustrates the “sitting” position of the touch plate for Z-axis zeroing.

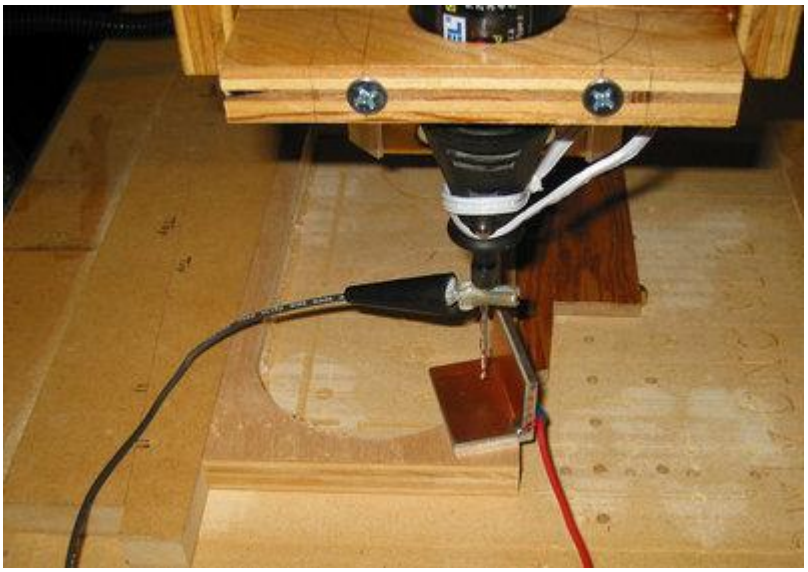


Figure 4

Tip: The copper surface will tarnish over time from handling it with you hands. You should periodically clean the surface with alcohol to remove oils then burnish the copper with a pencil eraser. This is especially critical for XY touching. When doing the XY “touch” press the angle aluminum against the work piece's corner if necessary.

My Mach3 Ports and Pins Settings

Figures 5 and 6 show my Mach3 Ports and Pins settings for Inputs. I am using the Probe input for the Touch Plate but any unused Input could be used.

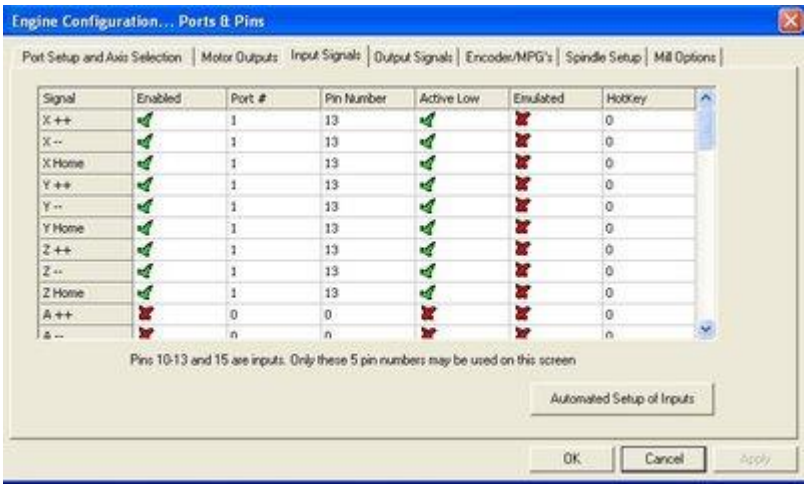


Figure 5

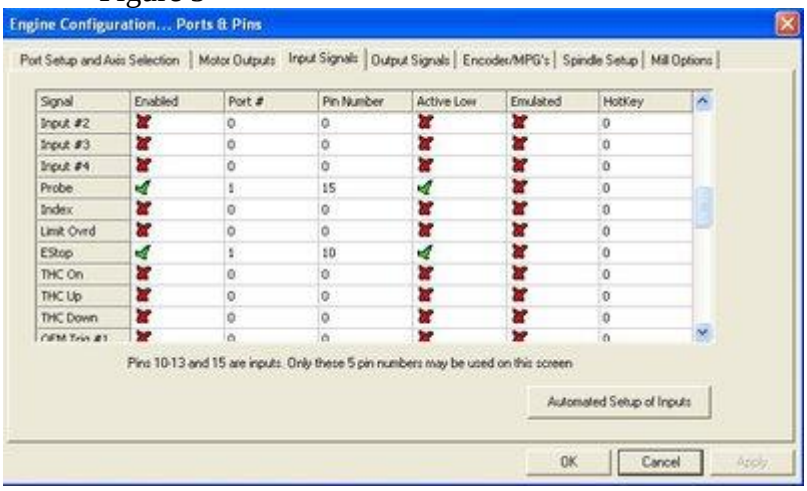


Figure 6

For the logic ground terminal I am using the Port 15 Common terminal on the Break-Out Board (BOB).

Auto Tool Zero Method for XYZ Axis Using Mach3 - Part 2

Before jumping into the Auto Tool Zero button macro, the very first thing you should do is to use a test script with your Mach3 to determine which version of the Auto Tool Zero macro code you should use and whether your XML config file has any form of corruption which may have gone undetected. I found it very beneficial to run a specific SetDRO/GetDRO or SetOEMDRO/GetOEMDRO command test script first. Some earlier versions of Mach3 do not support the Set/GetOEMDRO command but will support the older variant Set/GetDRO.

Using the Test Script

This script will help rule out a specific type of corruption in the XML file used when starting Mach3. In addition to this test script I will also provide screenshots of what you will see that indicates a corrupt XML file. Two versions of the test script are provided. One uses the legacy SetupDRO/GetDRO commands the other uses the newer version of these commands.

What the Test Script Does and Should Not Do

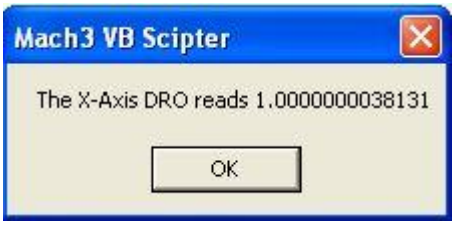
First, we will load the test script into the Mach3 VB Editor. This script only "sets" a value into the X DRO then "reads" the values it just set to a message window. Corruption in the XML file is indicated if the displayed values in the message window either have too many decimal places or incorrect values.

The script will write 6 different values to the X DRO in the following order 0.0000, 1.0000, 2.0000, 3.0000, 4.0000 and 5.0000 at two second intervals then "read back" and display each value in a message dialogue box. You will have to click the OK button in each message box to continue cycling through all six iterations.

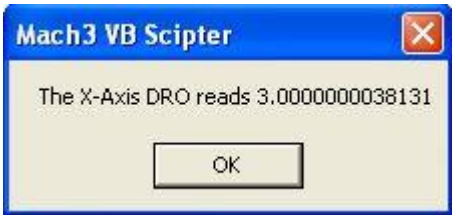
Although the values "set" are whole inches out to four decimal places, the expected "read" and displayed values should either be "0", "1", "2", "3", or 0.0000, 1.0000, 2.0000, 3.0000 etc.. If the first value displayed in the message box is 3.813065418.... instead of "0",

0 or 0.0000 is not zero

and the next values displayed in the message box are similar to these:



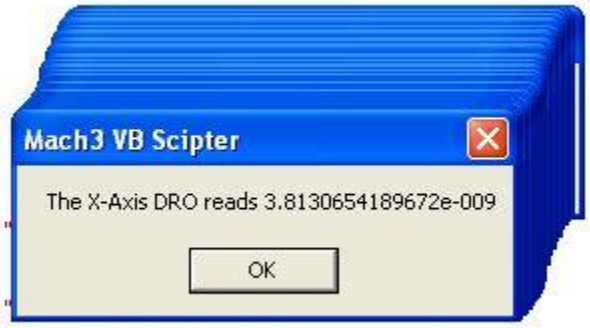
1 or 1.0000 error



3 or 3.0000 error

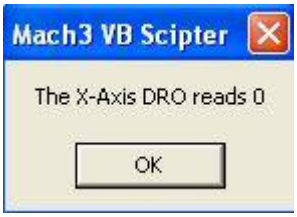
...then your XML file has some form of corruption and you need to revert to your original XML backup and not the daily Auto Save version.

Another indication of XML corruption is evident if you click and drag a message dialogue box and you get this:

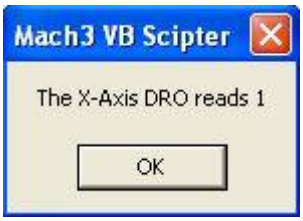


Window Echo Error

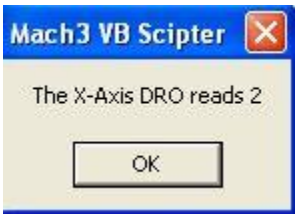
If your XML file is healthy dragging a message dialogue box will simply move it with no “echoes” and the values “read” and displayed in the message dialogues should look like these:



0 or 0.0000 is good



1 or 1.0000 is good



2 or 2.0000 is good

If the results of this test on your system were correct then you can proceed to test the XYZ Auto Tool Zero script. If you got any message displaying values with errors similar to those shown here, then you must revert to the COPY of the original CNC Mill profile XML file your created during installation. If you weren't diligent in maintaining periodic copies of your CNC Mill XML file then this may mean that you have to redo all your settings in the original copy. Also, you cannot use any of the daily XML profile backups automatically generated by Mach3 as they may also be corrupt.

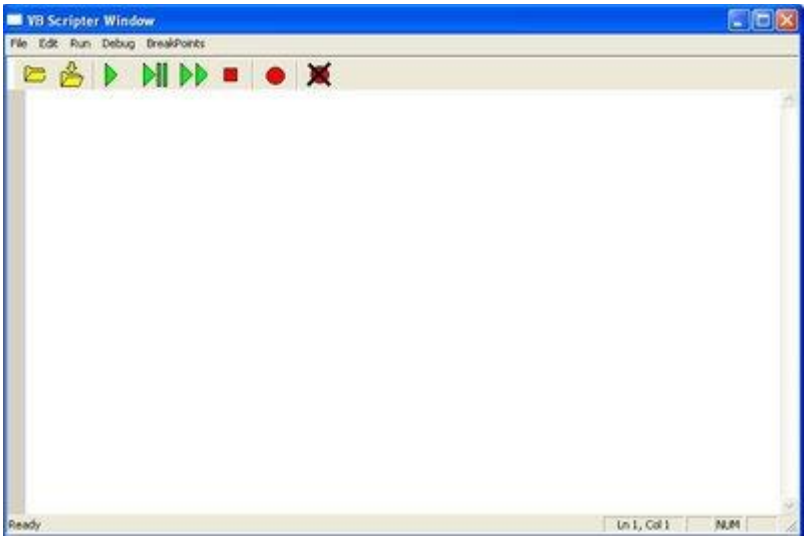
Be sure to HOME your X-axis after you have finished running the test script and before you do any normal CNC work.

[b]Loading the Test Script[/b]

1. With Mach3 running click “Operator > VB Script Editor.

To open VB Script Editor

2. A blank editor window will open.



VB Editor opens a blank script

3. Copy and paste the legacy version of the Test script first into the script window
4. Click the “Run” button
5. Watch the X DRO and click through the six message box OK buttons noting the values set and read back.

If you got the correct results, you can then try the newer version.

1. Click anywhere on blank space in the script window
2. Type Ctrl+A or Edit > Select All
3. Copy and paste the newer test version over the selected text
4. Click the “Run” button
5. Watch the X DRO and click through the six message box OK buttons noting the values set and read back.

If you get the same results then your version of Mach3 supports the newer code commands. Any other results may indicate your version is too old, so stick with the version that works. If only the legacy test script works, then you will need to use the copy of the legacy auto tool zero macro I provide otherwise you should use the newer version that uses the "SetOEMDRO()" commands.

The Legacy Version of the Test Script: (copy and paste from "Sub Main()" down through the first "End Sub" into the VB Editor)

```
Sub Main()

Dim ConfirmReady As String
Dim XaxisDRO As Integer
Dim YaxisDRO As Integer
Dim ZaxisDRO As Integer
XaxisDRO=0
YaxisDRO=1
ZaxisDRO=2

MsgBox( "Starting X DRO Set and Get..." )
SetDRO(XaxisDRO, 0.0000)
Sleep 1000
MsgBox "The X-Axis DRO reads " & GetDRO(XaxisDRO)
SetDRO(XaxisDRO, 1.0000)
Sleep 1000
MsgBox "The X-Axis DRO reads " & GetDRO(XaxisDRO)
SetDRO(XaxisDRO, 2.0000)
Sleep 1000
MsgBox "The X-Axis DRO reads " & GetDRO(XaxisDRO)
```

```
SetDRO(XaxisDRO, 3.0000)
Sleep 1000
MsgBox "The X-Axis DRO reads " & GetDRO(XaxisDRO)
SetDRO(XaxisDRO, 4.0000)
Sleep 1000
MsgBox "The X-Axis DRO reads " & GetDRO(XaxisDRO)
SetDRO(XaxisDRO, 5.0000)
Sleep 1000
MsgBox "The X-Axis DRO reads " & GetDRO(XaxisDRO)
Sleep 1000

End Sub
```

The Newer Command Version of the Test Script:

```
Sub Main()

Dim ConfirmReady As String
Dim XaxisDRO As Integer
Dim YaxisDRO As Integer
Dim ZaxisDRO As Integer
XaxisDRO=800
YaxisDRO=801
ZaxisDRO=802

MsgBox( "Auto Zeroing X..." )
SetOEMDRO(XaxisDRO, 0.0000)
Sleep 1000
MsgBox "The X-Axis DRO reads " & GetOEMDRO(XaxisDRO)
SetOEMDRO(XaxisDRO, 1.0000)
Sleep 1000
MsgBox "The X-Axis DRO reads " & GetOEMDRO(XaxisDRO)
SetOEMDRO(XaxisDRO, 2.0000)
Sleep 1000
MsgBox "The X-Axis DRO reads " & GetOEMDRO(XaxisDRO)
SetOEMDRO(XaxisDRO, 3.0000)
Sleep 1000
MsgBox "The X-Axis DRO reads " & GetOEMDRO(XaxisDRO)
SetOEMDRO(XaxisDRO, 4.0000)
Sleep 1000
MsgBox "The X-Axis DRO reads " & GetOEMDRO(XaxisDRO)
```

```
SetOEMDRO(XaxisDRO, 5.0000)
Sleep 1000
MsgBox "The X-Axis DRO reads " & GetOEMDRO(XaxisDRO)
Sleep 1000

End Sub
```

In Part 3 of this series we will get to the real meat of Auto Tool Macro code.

Auto Tool Zero Method for XYZ Axis Using Mach3 - Part 3

The Visual Basic Code – The Main Event
That Mach3 provides for customizable, user-defined button macros on some of the existing screen buttons is what makes this possible without having to do Mach3 screen designs to add new buttons. The Auto Tool Zero button on the Programs Run screen is the one used for this purpose.

Using this method requires the utmost in Operator attention and focus.

I first learned about how to auto zero the Z-axis from Patrick's BYCNC web site. Chris Cockrum's Control showed how to customize the Auto Tool Zero button macro to use a touch plate to zero the Z-axis here http://buildyourcnc.com/creationsgallery.aspx. I simply took Chris' insight a few steps further, posted my thoughts and ideas on the Mach3 Forum and got some great feedback from forum users to arrive at my current script.

The Visual Basic (VB) script utilizes commands and statements from Cypress Visual Basic, found in the VB-Script-Commands.pdf document and the Mach3 custom Macro commands found in the Mach3 V3.x Macro Programmer Reference Draft V0.22.pdf document on the Mach3 Wiki site here:
http://www.machsupport.com/MachCustomizeWiki/index.php?title=Main_Page , (note: this is now a bad web link) or you can find in on the Mach Support Forum. Make sure you get V0.22 as it has corrections that earlier versions do not have.

Please note that the Programmer Reference is still a draft. Also refer to related documentation on the Wiki site.

Here are the functional criteria I wanted:
1. To confirm my touch plate leads were connected and I was ready to proceed or exit.
2. To be able to set the work piece's XY & Z zero position as one integrated action
3. The option to zero X and Y only, just the Z-axis or the XY and Z axis.
4. To be able to pause and change the tool bit after zeroing the XY axis so that the tool bit mounted would be used to set the work piece Z-axis “zero”. This gives me the option of using a standard .125” steel reference rod when zeroing the X and Y axis.
5. To be able to skip zeroing the Z-axis.

There are important changes forth coming in version 4 of Mach3 relating to Mach3's macro commands. The scripts provide here incorporate the new Mach3 command variants that replace legacy versions of Set/GetDRO commands. These old command versions will not work in Mach3 version 4. The SetDRO and GetDRO subroutine commands are superseded by SetOEMDRO and GetOEMDRO commands. These commands write new values to the Mach3 screen “Digital Read Outs” (DRO) to set the current X, Y or Z tool locations.

Since the Visual Basic code for X, Y and Z is similar, I only provided program comments for some of the code.

Program Line Labels
Mach3's present implementation of Cypress Visual Basic only supports VB labels if they are numeric text. Most VB code compilers and interpreters support alpha-numeric labels. Labels are used for re-direction or jump points. “GoTo 3” means jump from "here" immediately to the program line labeled as “3:”. An example of a text label would be “BeginXY:” and "GoTo BeginXY".

The Script Explained
It would be a good time for you to check and see which version of Mach3 you have. In Mach3 click Help > About Mach3 CNC Controller... . I am using Version R3.042.038.

In a VB script, comments are proceeded by the single quote character, "'" and are have no effect on script execution. User responses to input prompts in my code are expected to be lower-case characters in the form of “y” or “n”. The green text are comments, black text are the Visual Basic code lines.

Due to expected changes coming in Mach3 I have provided two different versions of the VB script, one that uses "SetOEMDRO()" and one that uses "SetDRO()". The first example has comments to help explain what the code lines do or mean. Knowing about these two command variants will help you determine which one will work best for your version of Mach3.

The legacy SetDRO(a, v) command will work with almost any prior version of Mach3 but remember, Mach3 Version4 will not support it. The “a” must be a number 0 through 5 where the x-axis=0, y-axis=1, z-axis=2, a-axis=3, etc. The “v” must be a number that represents the new value to write to that axis' DRO.

The new preferred SetOEMDRO(a, v) command works exactly the same way, but requires different values to represent the XYZABC axis. For this command version, x-axis=800, y-axis=801, z-axis=802, etc. A test script is also provided here to see if your version can use this newer command. Versions after Version R3.042.020 I know can use the newer command version.

When the code runs and you are prompted with a question you must click your cursor in the input line to type your response before you can click the OK button. This code example is set for a tool diameter of .125" and a Z-touch plate thickness of .180". These values can be changed to suit your needs. I have several versions of the code for different tool diameters and load the one I need at the time. **NOTE: The tool must be pre-positioned in relation to the touch plate, as shown in Figure 3 in part 1 of this series. The X an Y-axis should be approximately centered on the touch plate face it will move toward and about 3/4" away. The Z-axis tool tip should be approximately 1/2 to 3/4" above the work surface and the Z-axis DRO manually set to 2.000".**

XYZ Auto Tool Zero VB Macro Version 3.2 – using the newer SetOEMDRO() command

```
Sub Main()

' declare character and numeric string variables and their variable types before using them, this is done with the Dimension statement
Dim ConfirmReady As String
Dim DoXY As String
Dim DoZ As String
Dim XaxisDRO As Integer
Dim YaxisDRO As Integer
Dim ZaxisDRO As Integer

' define variable values used by the SetOEMDRO() command
XaxisDRO=800
YaxisDRO=801
ZaxisDRO=802

' the next line makes sure the operator has the touch plate leads connected and is ready, otherwise exit completely
ConfirmReady = AskTextQuestion("Confirm Touch plate leads are connected and ready. (y/n)")

' if the response is yes then jump to 1: otherwise jump to 5:
If ConfirmReady = "y" Then GoTo 1 Else GoTo 5

1:
' present the option to zero X & Y or bypass and go straight to zeroing Z
DoXY = AskTextQuestion("Zero X and Y also? (y/n)")

' if the response is yes then jump to 2: and begin XY “touching” otherwise jump to 3: and begin with Z “touching
If DoXY = "y" Then GoTo 2 Else GoTo 3

2:
' write message to Mach3's status line
Message( "Auto Zeroing X..." )

' zero the x-axis where ever it is to ensure it moves toward the touch plate
SetOEMDRO(XaxisDRO, 0.0000)

' give the command time to “zero” the DRO and let the CPU do other things for 1000 milliseconds
Sleep 1000

If IsSuchSignal (22) Then ' signal 22 equates to the Input 15 port of the BOB

' if the touch plate leads have not touched yet move the X-axis toward the value of -2 with a feed rate of 10
```

```
code "G31 X-2 F10"

' while the x-axis is moving let the CPU do other things for 100 milliseconds until the touch plate input goes low
While IsMoving()
Sleep 100

' this is the end to the While IsMoving statement
Wend

' at the point the touch plate leads touch, set the X-axis DRO to half the diameter of the tool bit.
' if you are using a 1/4” shank bit change the .0625 to .125, make the same change in the section for the Y-axis
SetOEMDRO(XaxisDRO, .0625)

' let the CPU do other things while this is happening
Sleep 1000

' then move the tool in the X-axis away from the touch plate to +.5”
code "G1 X.5"

' we are done with zeroing the X-axis proceed immediately to do Y
End If

Message( "Auto Zeroing Y..." )
SetOEMDRO(YaxisDRO, 0.0000)
Sleep 1000
If IsSuchSignal (22) Then
code "G31 Y-1 F10"
While IsMoving()
Sleep 100
Wend
SetOEMDRO(YaxisDRO, .0625)
Sleep 1000
code "G1 Y.5"
End If

3:
' get a confirmation that the user wants to proceed with the Z zeroing
DoZ = AskTextQuestion("Position the touch plate to zero Z. y to continue or n to skip. (y/n)")
If DoZ = "y" Then GoTo 4 Else GoTo 6

4:
Message( "Auto Zeroing Z..." )
SetOEMDRO(ZaxisDRO, 0.0000)
Sleep 1000
```

```
If IsSuchSignal (22) Then
code "G31 Z-2 F10"
While IsMoving()
Sleep 100
Wend
' the thickness of the z touch plate described in this document is .180 inches
Call SetOEMDRO(ZaxisDRO, .180)
Sleep 100
code "G1 Z1"
End If
GoTo 6

5:
Message ("Tool zeroing aborted. Try again when ready.")
GoTo 7

6:
Message "Tool zeroing complete. Check the results on the DROs."

7:
End Sub

' VB syntax defines a subroutine starting with “Sub 'name'()” line, e.g. Sub Main() as the beginning point of a VB
' subroutine and defines the ending of a subroutine with “End Sub”

' In a similar fashion an “If...Then” statement must have a closing “End If” statement unless the “If...Then
' performs an immediate “GoTo” jump command.
```

Here is the code with all comments removed.

```
Sub Main()

Dim ConfirmReady As String
Dim DoXY As String
Dim DoZ As String
Dim XaxisDRO As Integer
Dim YaxisDRO As Integer
Dim ZaxisDRO As Integer

XaxisDRO=800
YaxisDRO=801
ZaxisDRO=802
```

```
ConfirmReady = AskTextQuestion("Confirm Touch plate leads are connected and ready. (y/n)")
If ConfirmReady = "y" Then GoTo 1 Else GoTo 5

1:
DoXY = AskTextQuestion("Zero X and Y also? (y/n)")
If DoXY = "y" Then GoTo 2 Else GoTo 3

2:
Message( "Auto Zeroing X..." )
SetOEMDRO(XaxisDRO, 0.0000)
Sleep 1000
If IsSuchSignal (22) Then
code "G31 X-2 F10"
While IsMoving()
Sleep 100
Wend
SetOEMDRO(XaxisDRO, .0625)
Sleep 1000
code "G1 X.5"
End If

Message( "Auto Zeroing Y..." )
SetOEMDRO(YaxisDRO, 0.0000)
Sleep 1000
If IsSuchSignal (22) Then
code "G31 Y-1 F10"
While IsMoving()
Sleep 100
Wend
SetOEMDRO(YaxisDRO, .0625)
Sleep 1000
code "G1 Y.5"
End If

3:
DoZ = AskTextQuestion("Position the touch plate to zero Z. y to continue or n to skip. (y/n)")
If DoZ = "y" Then GoTo 4 Else GoTo 6

4:
Message( "Auto Zeroing Z..." )
SetOEMDRO(ZaxisDRO, 0.0000)
Sleep 1000
If IsSuchSignal (22) Then
code "G31 Z-2 F10"
While IsMoving()
```

```
Sleep 100
Wend
SetOEMDRO(ZaxisDRO, .180)
Sleep 1000
code "G1 Z1"
End If
GoTo 6

5:
Message ("Tool zeroing aborted. Try again when ready.")
GoTo 7

6:
Message "Tool zeroing complete. Check the results on the DROs."

7:
End Sub
```

The Legacy SetDRO() Command Version is here:

```
Sub Main()

Dim ConfirmReady As String
Dim DoXY As String
Dim DoZ As String
Dim XaxisDRO As Integer
Dim YaxisDRO As Integer
Dim ZaxisDRO As Integer

' notice the legacy SetDRO() command uses single digit values for xyzabc
XaxisDRO=0
YaxisDRO=1
ZaxisDRO=2

ConfirmReady = AskTextQuestion("Confirm Touch plate leads are connected and ready. (y/n)")
If ConfirmReady = "y" Then GoTo 1 Else GoTo 5

1:
DoXY = AskTextQuestion("Zero X and Y also? (y/n)")
If DoXY = "y" Then GoTo 2 Else GoTo 3

2:
Message( "Auto Zeroing X..." )
SetDRO(XaxisDRO, 0.0000)
Sleep 100
```

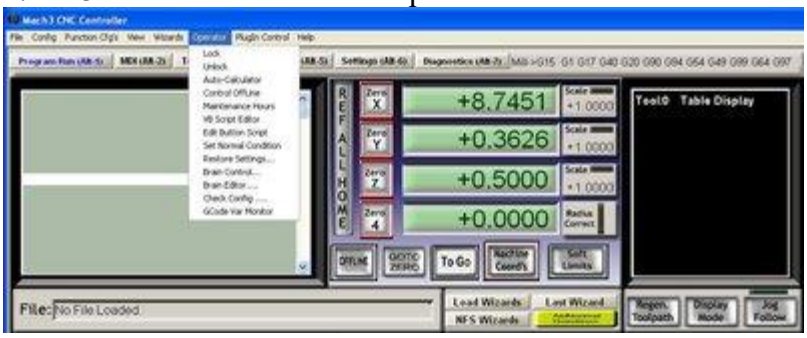
```
If IsSuchSignal (22) Then
code "G31 X-2 F10"
While IsMoving()
Sleep 100
Wend
SetDRO(XaxisDRO, .0625)
Sleep 100
code "G1 X.5"
End If

Message( "Auto Zeroing Y..." )
SetDRO(YaxisDRO, 0.0000)
Sleep 100
If IsSuchSignal (22) Then
code "G31 Y-1 F10"
While IsMoving()
Sleep 100
Wend
SetDRO(YaxisDRO, .0625)
Sleep 100
code "G1 Y.5"
End If

3:
DoZ = AskTextQuestion("Position the touch plate to zero Z. y to continue or n to skip. (y/n)")
If DoZ = "y" Then GoTo 4 Else GoTo 6

4:
Message( "Auto Zeroing Z..." )
SetDRO(ZaxisDRO, 0.0000)
Sleep 100
If IsSuchSignal (22) Then
code "G31 Z-2 F10"
While IsMoving()
Sleep 100
Wend
SetDRO(ZaxisDRO, .180)
Sleep 100
code "G1 Z1"
End If
GoTo 6

5:
Message ("Tool zeroing aborted. Try again when ready.")
GoTo 7
```

6:
Message "Tool zeroing complete. Check the results on the DROs."

7:
End Sub

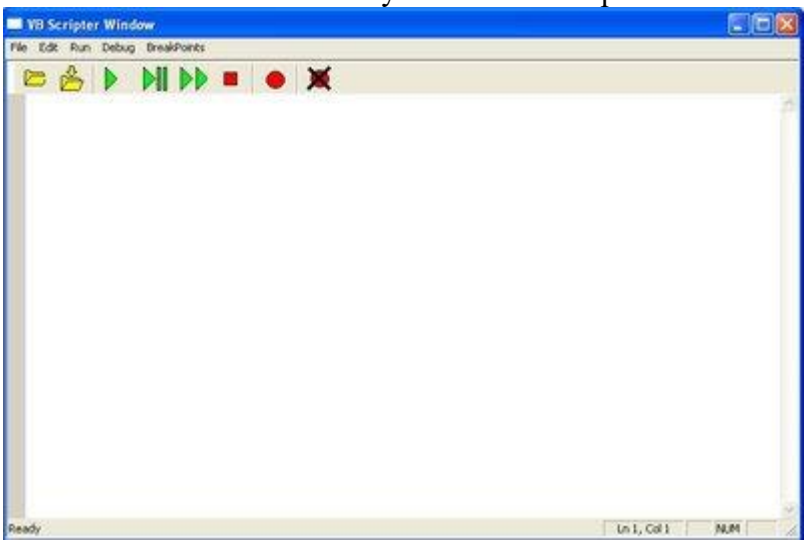
Loading the VB Script to the Auto Tool Zero Button
From the Mach3 Program Run screen:

1. Click “Operator” on the Menu bar
2. Then click “Edit Button Script”. The buttons that are editable will start flashing.



Accessing the Edit Button Script Window

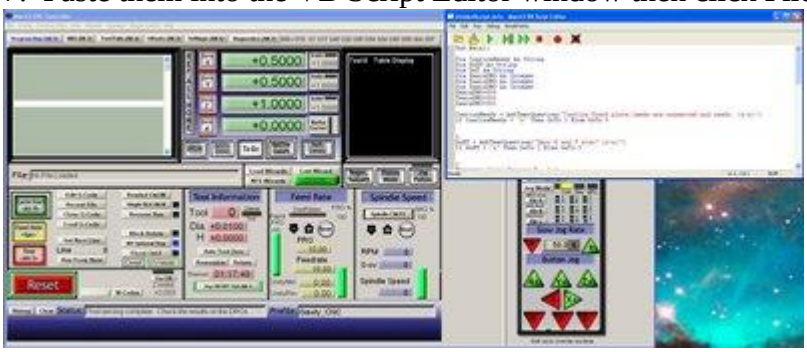
3. Click the flashing Auto Tool Zero button. The Mach3 VB Script Editor window will open. By default this file will always be named "HiddenScript.m1s and at first there is one line of code in the edit window that may have a “Not Implemented” message in it.



VB Editor Window Opens

4. Click anywhere in the edit window's white space
5. Highlight any lines by typing Ctrl+A then press the Delete key or click Edit > Select All > press Delete key.

6. Copy all of the lines in the script from this document from the comments omitted section, from “Sub Main()” to the last “End Sub”
7. Paste them into the VB Script Editor window then click File > Save.



Pasted VB macro

7. Close the VB Script Editor window. If you are asked to save changes click Yes and exit out of Mach3.

Exiting from Mach3 at this point is very important.

8. Re-open Mach3

9. Click the flashing Reset button.

10. Check to make sure the VB script is in fact loaded to the Auto Tool Zero button by clicking “Operator > Edit Button Script”
11. Then click the flashing Auto Tool Zero button again to open the VB Script Editor.

If the script is showing, then the next thing to do is test the script. If the script is not showing repeat steps 1 through 11.

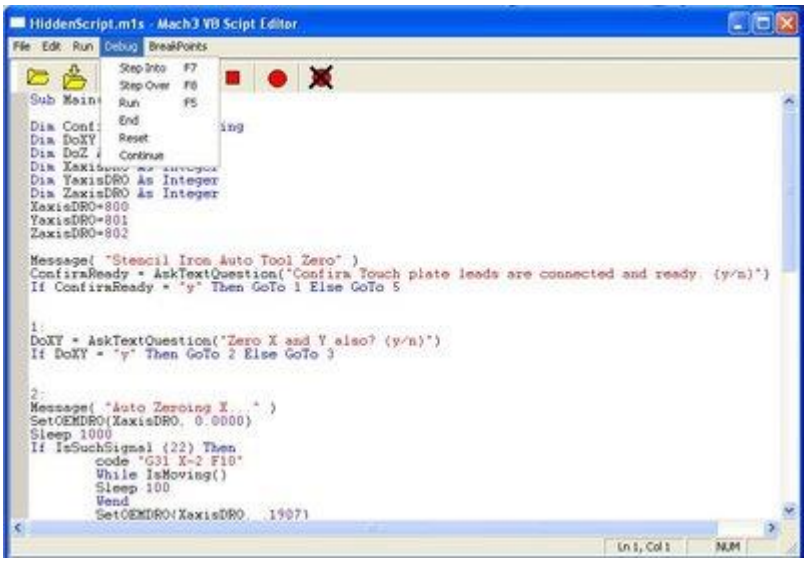
Testing the XYZ Auto Tool Zero Script

IMPORTANT: Whenever you expect to test or use the Auto Tool Zero button script it is imperative that you first HOME all of your axes, then pre-position your tool's XY axes locations relative to the XY touch plate “standing” position as shown in Figure 3. The tool starting position must be less than 2 inches from the touch plate X face, less than 1 inch from the touch plate Y face and less than 2 inches from the touch plate Z face.

Referring again to Figure 3 or 4 in Part 1, the tool on my table will move in the X axis away from the camera as a positive direction, will move Y in a positive direction toward the left and move Z downward in a negative direction. If any of your axis move opposite to this you need to change the "move" value in the code to either plus or minus for the respective axis.

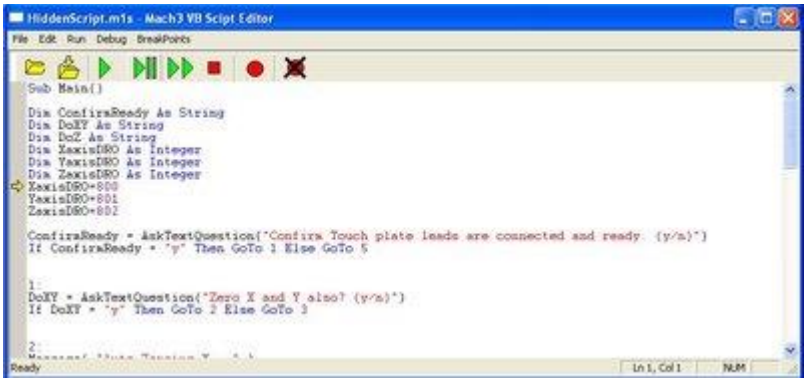
Before any VB script is used it should be tested in the most conservative manner. The ideal way to test VB scripts is to open the Mach3 Visual Basic Editor with the script loaded and single-step

through the lines of code. This is also referred to as “Stepping Into Program” a line at a time. By testing with the single-step method you have the ability to stop executing the next line at any time. There are three ways to step through the program one line at a time, click the green, right-facing triangle with two parallel lines tool bar button, click Debug > Step Into but the easiest way is to press F7 for each line step.



Debug Menu

When stepping through a program a yellow arrow will appear to the left of the next line to be executed.



Step Mode Shows Yellow Arrow

Single-stepping through the program can be stopped (aborted) at any time by clicking the round, Red tool button. But when single-stepping through any program where aborting is not required, it is important to make sure you have stepped through the entire program all the way through the "End Sub" command line. The script window will reflect you have successfully stepped through all the lines by displaying the program at the beginning.

The respective axis motors will begin their movement when their “code G31...” line is executed. You should position the VB Script Editor window as shown above so you can see your DRO's and have ready access to the Jog buttons and “Reset” button or your eStop button.

As long as a VB script is in the execution process, Mach3 functions and buttons, e.g. like the Jog controls will not be functional until the code execution has ended.

Side Note: I had thought about adding user input in the code to ask for the tool diameter but I chose not to because I didn't want to have to answer too many questions each time I ran the Auto Tool Zero function. That is why I opted to have several versions of the code for the few tool diameters that I use, e.g. XYZ-zero eight.m1s, XYZ-sero-quarter.m1s, XYZ-zero hotstencil.m1s. **So, which every tool I'm using initially, I pre-load the appropriate macro and save it in the Auto Tool Zero button editor as HiddenScript.m1s.**

I hope you find this as useful as I have. My thanks to the dedicated, experienced programmers on the Mach3 forum that helped me realize these end results.

I cannot guarantee this will work with your setup and will be happy to answer any questions.