Mach3 Version 3.x
Macro Programmers Reference Manual

Draft Revision 0.23
August 22nd, 2010

For Mach3 v3 versions thru 3.43.19.
Table of Contents
Introduction .............................................................................................................. 1
Legacy Scripting Functions Grouped Alphabetically .............................................. 2
ActivateSignal ...................................................................................................... 2
AppendTeachFile ................................................................................................... 2
AskTextQuestion .................................................................................................. 3
CloseDigFile ......................................................................................................... 4
CloseTeachFile ..................................................................................................... 4
Code ..................................................................................................................... 5
CoupleSlave ........................................................................................................ 6
DeactivateSignal .................................................................................................. 7
DoMenu ............................................................................................................... 8
DoButton .............................................................................................................. 9
DoOEMButton ................................................................................................... 10
DoSpinCCW ...................................................................................................... 10
DoSpinCW ......................................................................................................... 11
DoSpinStop ........................................................................................................ 11
EndTHC ............................................................................................................. 12
FeedRate ............................................................................................................ 12
FileName ............................................................................................................ 13
GetABSPosition ................................................................................................. 13
GetACoor ........................................................................................................... 14
GetActiveProfileDir ........................................................................................... 15
GetActiveProfileName ....................................................................................... 15
GetActiveScreenSetName .................................................................................. 16
GetCoord ............................................................................................................ 16
GetCurrentTool .................................................................................................. 17
GetDRO ............................................................................................................. 18
GetDROString .................................................................................................... 19
GetIJMode .......................................................................................................... 20
GetLED .............................................................................................................. 20
GetLoadedGCodeDir ......................................................................................... 22
GetLoadedGCodeFileName ............................................................................... 22
GetMachVersion ................................................................................................ 23
GetMainFolder ................................................................................................... 23
GetMyWindowsHandle ....................................................................................... 24
GetOEMDRO .................................................................................................... 24
GetOEMLED ..................................................................................................... 25
getPage ............................................................................................................ 25
GetParam .......................................................................................................... 26
GetPortByte ..................................................................................................... 28
GetToolParam .................................................................................................. 28
GetRPM ............................................................................................................ 30
GetSafeZ ......................................................................................................... 30
OpenTeachFile ................................................................. 57
Param1 ............................................................................. 58
Param2 ............................................................................. 58
Param3 ............................................................................. 59
PlayWave ......................................................................... 60
ProgramSafetyLockout ..................................................... 60
PutPortByte ..................................................................... 60
Question .......................................................................... 61
QueueDepth ..................................................................... 62
Random ........................................................................... 62
RefCombination .................................................................. 63
ResetAxisSwap .................................................................. 64
ResetTHC .......................................................................... 64
RetractMode ..................................................................... 65
roun .................................................................................... 65
RunFile ............................................................................. 66
RunScript ......................................................................... 66
SaveWizard ....................................................................... 68
SetButtonText ..................................................................... 68
SetCurrentTool ................................................................... 69
SetDRO ............................................................................. 69
SetFeedRate ...................................................................... 70
SetFormula ......................................................................... 71
SetIJMode .......................................................................... 72
SetMachZero ...................................................................... 72
SetOEMDRO ....................................................................... 73
SetPage ............................................................................. 73
SetParam ........................................................................... 74
SetPulley .......................................................................... 76
SetSafeZ ........................................................................... 76
SetScale ............................................................................ 77
SetSpinSpeed ..................................................................... 77
SetTicker ........................................................................... 78
SetTimer ............................................................................ 78
SetToolDesc ....................................................................... 79
SetToolParam ..................................................................... 80
SetToolX ............................................................................ 81
SetToolZ ............................................................................ 81
SetTriggerMacro ................................................................... 82
SetUserDRO ....................................................................... 82
SetUserLabel ...................................................................... 83
SetUserLED ........................................................................ 84
SetVar .............................................................................. 84
SingleVerify ....................................................................... 85
<table>
<thead>
<tr>
<th>Legacy Functions Grouped By Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digitizing</td>
<td>97</td>
</tr>
<tr>
<td>G-Code &amp; G-code Files</td>
<td>97</td>
</tr>
<tr>
<td>Lathe-only Functions</td>
<td>97</td>
</tr>
<tr>
<td>Mach3 Configuration &amp; Status</td>
<td>97</td>
</tr>
<tr>
<td>Referencing, Verifying &amp; Zeroing Axes</td>
<td>98</td>
</tr>
<tr>
<td>SafeZ</td>
<td>98</td>
</tr>
<tr>
<td>Wizards &amp; Plugins</td>
<td>98</td>
</tr>
<tr>
<td>Machine Status &amp; Control</td>
<td>98</td>
</tr>
<tr>
<td>Motion Control</td>
<td>99</td>
</tr>
<tr>
<td>Spindle Control</td>
<td>99</td>
</tr>
<tr>
<td>Tool Parameters and Tool Changes</td>
<td>99</td>
</tr>
<tr>
<td>Torch Height Control</td>
<td>99</td>
</tr>
<tr>
<td>Screen sets</td>
<td>100</td>
</tr>
<tr>
<td>User Dialogs</td>
<td>100</td>
</tr>
<tr>
<td>Signals and Port I/O</td>
<td>101</td>
</tr>
<tr>
<td>Teach Files</td>
<td>101</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>101</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Modbus Functions Grouped Alphabetically</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInBit</td>
<td>103</td>
</tr>
<tr>
<td>GetInput</td>
<td>103</td>
</tr>
<tr>
<td>ResetOutBit</td>
<td>103</td>
</tr>
<tr>
<td>SetHomannString</td>
<td>104</td>
</tr>
<tr>
<td>SetModIOString</td>
<td>104</td>
</tr>
<tr>
<td>SetModOutput</td>
<td>105</td>
</tr>
<tr>
<td>SetOutBit</td>
<td>105</td>
</tr>
<tr>
<td>WaitForPoll – Unreliable</td>
<td>106</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Serial Output Functions Grouped Alphabetically</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>107</td>
</tr>
</tbody>
</table>
Introduction

This Programmers Reference Manual documents the commonly used Cypress Basic (CB) function calls available to macro programmers using Mach3 version 3. This information is being provided primarily to help Mach3 users understand existing macro code. While this interface will continue to be supported by future Mach3 versions for some period of time, Mach3 version 4 will provide a completely new, much more regular interface, much higher functionality interface for CB macro programming. It is strongly recommended that all new CB code use the new interface, as support for this old one will be discontinued at some point in the, possibly not too distant, future. In addition, it is unlikely there will be any further updates or bug fixes to this now obsolete interface after the release of Mach3 version 4, so any existing bugs and anomalies (and there are quite a few) will remain.

No attempt has been made to make this an exhaustive document covering all of the CB functionality. There are many functions which were never previously documented, or which were documented incompletely or incorrectly. In many cases, these functions are not included in this document. There are a number of functions which were partially documented, but found to either not function as documented, to have significant restriction in their operation, or, in some cases, were felt to be either of no real value, or even risky to use. These functions are generally not included in this document. The functionality described herein has been tested against Mach3 version 3.042.020. Some functions may behave differently in other versions. Some functions will be missing entirely in some earlier versions.
Legacy Scripting Functions Grouped Alphabetically

ActivateSignal

Sub ActivateSignal (SigNum As Integer)

This function causes the specified Mach output signal to be driven to its active state. If the signal is defined in Config->Ports&Pins as ActiveHigh, it will be driven to a logic High level, otherwise it will be driven to a logic Low level.

Arguments:
SignalID must be one of the pre-defined Mach3 CB output signal constants (see CB Constants), or other value or expression that evaluates to one of those values.

Return Value:
None

Example:
ActivateSignal(OUTPUT2)  ' Turn on Flux Capacitor
Sleep(1000)              ' Give it time to charge fully
DeactivateSignal(OUTPUT2) ' Turn it off

See also:
DeactivateSignal(), CB Constants

AppendTeachFile

Function AppendTeachFile(Filename As String) As Integer

This function re-opens an existing Teach file at Gcode\Filename in the Mach3 directory, and appends any commands subsequently executed via MDI or Code() to that file, until CloseTeachFile() is executed. The specified file must already exist.

Arguments:
Filename is the name of the Teach file to be re-opened. The file must reside in the Gcode subdirectory of the Mach3 install directory.

Return Value:
A non-zero value is returned if the operation was successful.

Example:
' Create a new Teach File in Mach3\Gcode
MyTeachFile = “TeachMe.nc”
Err = OpenTeachFile(MyTeachFile)
If Err <> 0 Then
    ' Teach file created successfully
    ' Write some G Code to it
    Code “G00 X0 Y0”
    Code “G02 X0 Y0 I-1 J0 F40”
    ' Close the Teach file
    CloseTeachFile()
    ' Now load the teach file for execution
    LoadTeachFile()
Else
    ' OpenTeachFile failed
    Message “Unable to open Teach File”
End If

See also:
OpenTeachFile(), CloseTeachFile()

AskTextQuestion

Function AskTextQuestion(Prompt As String) As String

This function displays a dialog box containing the specified prompt string, and waits for
the user to enter a text string in the dialogs text box. The user entered string is returned to
the caller.

Arguments:
Prompt string is the string that will be displayed above the text box when the
dialog is displayed.

Return Value:
The text string entered by the user.

Example:
Dim UsersName As String
UsersName = AskTextQuestion(“Please enter your name:”)
Message “Hello, “ & UsersName & “!”
CloseDigFile

Sub CloseDigFile()

This function closes an open digitizing file. If there is no open digitizing file, it does nothing.

Arguments:
None

Return Value:
None

Example:
OpenDigFile()  ' Open the digitizing file
ProbeOutline()  ' Call my custom probing function
CloseDigFile()  ' Close the digitizing file

See also:
OpenDigFile(), SetProbeActive(), IsProbing()
Return Value:
None

Example:
TeachFile = "MyTeachFile.nc"
Dim Err As Integer
Err = OpenTeachFile(TeachFile)
If Err = 0 Then
    MsgBox("Unable To Open: " & TeachFile)
Else
    ' MDI commands entered here are written to TeachFile
    Code("G55")
    Code("G00 X0.000 Y0.000 Z0.500")
End If
' Now close TeachFile
CloseTeachFile()
' Do something else here
' Now re-open TeachFile for append
Err = AppendTeachFile(TeachFile)
If Err = 0 Then
    MsgBox("Unable To Open: " & TeachFile)
Else
    ' MDI commands entered here are written to TeachFile
    Code("G56")
    Code("G00 X0.000 Y0.000 Z0.500")
End If
' Now close TeachFile
CloseTeachFile()

See also:
OpenTeachFile(), AppendTeachFile()

Code
Sub Code(Gcode As String)

This function executes the single line of G-code passed as the argument, exactly as if it had been entered on the MDI line, or encountered in a G-Code program. Note that the G-code line is placed in the execution queue, but, in general, the Code() function will return before the line has actually been executed. If it is important that your program know that the line has completed execution, a While loop must be used with the IsMoving() function, as shown in the example below.
Note also that if a Teach file is currently open, the G-code line will not actually be executed, but will simply be written out to the Teach file.

The following optional modes of the Code() function are documented here only for completeness. Their use is discouraged, and support for them may be discontinued at any time, without warning.

Sub Code(“LOAD:” & FilePath)

This mode loads a G-Code file from the specified FilePath. The LoadFile() function should be used instead.

Sub Code(“SAVE_XML”)

This method immediately saves the XML configuration file.

**Arguments:**
A single G-code line to be executed as a String

**Return Value:**
None

**Example:**

```
' Select our fixture
Code(“G55”)
' Move away from the vise
Code(“G00 X-4.000 Y1.000 Z1.000”)
' Wait for movement to complete
While (IsMoving())
  ' Sleep, so other threads can run while we’re waiting
  Sleep(100)
Wend
' Done
```

**See also:**

**CoupleSlave**

Sub Function CoupleSlave(State As Integer)

This function causes any slaved axis to be coupled to, or de-coupled from, its master axis during homing.

**Arguments:**
State is an Integer value of 0 or 1 that defines whether the slave axis should be coupled to its master axis during homing. If State is 0, the axes will be un-coupled. If State is 1, the axes will be coupled.

**Return Value:**
None

**Example:**
```
' Define the axes
Dim Xaxis As Integer
Xaxis = 0
Dim Yaxis As Integer
Yaxis = 0

' Couple the slaved A axis to its master X axis
CoupleSlave(1)
' Home the master and slave
SingleVerify(Xaxis)
' Un-couple the X and A axes
CoupleSlave(0)
```

**See also:**
None

---

**DeactivateSignal**

Sub DeactivateSignal (SigNum As Integer)

This function causes the specified Mach output signal to be driven to its inactive state. If the signal is defined in Config->Ports&Pins as ActiveHigh, it will be driven to a logic Low level, otherwise it will be driven to a logic High level.

**Arguments:**
SignalID must be one of the pre-defined Mach3 CB output signal constants (see CB Constants), or other value or expression that evaluates to one of those values.

**Return Value:**
None

**Example:**
```
ActivateSignal (OUTPUT2) ' Turn on Flux Capacitor
Sleep(1000) ' Give it time to charge fully
DeactivateSignal (OUTPUT2) ' Turn it off
```

**See also:**
ActivateSignal (), CB Constants
**DoMenu**

Sub DoMenu (MenuIndex As Integer, MenuItem As Integer)

This function allows a macro script to invoke any function available through any of the Mach3 menus, exactly as if the user had clicked on the menu with the mouse. The specific menu item to be invoked is specified by the two arguments. The first indicates which menu the item to be invoked resides in. The second indicates which item within that menu is to be invoked.

**Arguments:**
- MenuIndex is the 0-based index of the menu to activate. The File menu is index 0, the Config menu is index 1, etc.
- MenuItem is the 0-based index of the menu item to activate. The first item on a given menu is index 0, the second is index 1, etc.

**Return Value:**
None

**Example:**
```
' Define menu indices, left-to-right, starting with 0
Dim FileMenu As Integer
FileMenu = 0
Dim ConfigMenu As Integer
ConfigMenu = 1
...
' Define File menu items, top-to-bottom, starting with 0
Dim FileMenuLoadGCodeMenuItem
FileMenuLoadGCodeMenuItem = 0
Dim FileMenuLazyCAMMenuItem As Integer
FileMenuLazyCAMMenuItem = 1
Dim FileMenuCloseFilesMenuItem As Integer
FileMenuCloseFilesMenuItem = 2
Dim FileMenuExitMenuItem As Integer
FileMenuExitMenuItem = 3
' Define Config menu items, top-to-bottom, starting with 0
Dim ConfigMenuDefineNativeUnitsMenuItem As Integer
ConfigMenuDefineNativeUnitsMenuItem = 0
Dim ConfigMenuPortsAndPinsMenuItem As Integer
ConfigMenuPortsAndPinsMenuItem = 1
...
' Pop-up the Config->Ports & Pins dialog
```
DoMenu(ConfigMenu, ConfigMenuPortsAndPinsMenuItem)

See also:
DoButton(), DoOEMButton()

DoButton
Sub DoButton(ButtonNum As Integer)

This legacy function allows a macro to execute an on-screen button function which has an assigned Button code. The specified function is invoked exactly as if the user had clicked the corresponding on-screen button with the mouse. Note that no actual on-screen button need exist for this function to work. This is simply an easy means to execute any of the “Button” functions through CB.

The use of DoButton is no longer recommended practice and this function exists only to support preexisting legacy scripts. This function is deprecated, and its use is strongly discouraged.

Legacy script note: Over time, there have been two different Button numbering schemes used with Mach; the “Button number” series and the “OEMButton number” series. This function uses the “Button number” series.

Within the “Button number” range, valid ButtonNums were from 0 to 31, which, at one time, corresponded to OEM LED numbers 1000 10 1031.

The numerical correspondence between the numbering series is not guaranteed for future releases of Mach.

Use the DoOEMButton function instead of this function.

Arguments:
ButtonNum must be one of the pre-defined Mach3 OEM Button Number constants (see CB Constants), or other value or expression that evaluates to one of those values.

Return Value:
None

Example:
' Define OEM codes for Mist On and Mist Off
Const RewindButton = 2

' rewind the gcode to start
DoButton(RewindButton)
See also:
DoButton(), DoOEMButton()

DoOEMButton
Sub DoOEMButton(OEMButtonCode As Integer)

This function allows a macro to execute any on-screen button function which has an
assigned OEM Button code. The specified function is invoked exactly as if the user had
clicked the corresponding on-screen button with the mouse. Note that no actual on-
screen button need exist for this function to work. This is simply an easy means to
execute any of the “OEM Button” functions through CB.

Arguments:
OEMButtonCode must be one of the pre-defined Mach3 OEM Button Code
constants (see CB Constants), or other value or expression that evaluates to one of
those values.

Return Value:
None

Example:
   ' Define OEM codes for Mist On and Mist Off
Const OEMButtonMistOn = 226
Const OEMButtonMistOff = 227

   ' Turn Mist coolant on for 3 seconds
DoOEMButton(OEMButtonMistOn)
Sleep(3000)
DoOEMButton(OEMButtonMistOff)

See also:
DoButton(), DoOEMButton(), DoMenu()

DoSpinCCW
Sub DoSpinCCW()

This function turns the spindle on, rotating counter-clockwise.

Arguments:
None
**Return Value:**
None

**Example:**
```
' Turn on the spindle, turning CCW
DoSpinCCW()
' Let it run 5 seconds
Sleep(5000)
' Now turn it off
DoSpinStop()
```

**See also:**
DoSpinCW(), DoSpinStop()

---

**DoSpinCW**
Sub DoSpinCW()

This function turns the spindle on, rotating clockwise.

**Arguments:**
None

**Return Value:**
None

**Example:**
```
' Turn on the spindle, turning CW
DoSpinCW()
' Let it run 5 seconds
Sleep(5000)
' Now turn it off
DoSpinStop()
```

**See also:**
DoSpinCCW(), DoSpinStop()

---

**DoSpinStop**
Sub DoSpinStop()

This function turns off the spindle.

**Arguments:**
None
Return Value:  
None

Example:  
' Turn on the spindle, turning CW  
DoSpinCW()  
' Let it run 5 seconds  
Sleep(5000)  
' Now turn it off  
DoSpinStop()

See also:  
DoSpinCW(), DoSpinCCW()

EndTHC
Sub EndTHC()

This function turns off torch height control. It is functionally identical to THCOff().

Arguments:  
None

Return Value:  
None

Example:  
StartTHC() ' Turn on torch height control  
... ' Do some cutting here  
EndTHC() ' Turn off torch height control

See also:  
StartTHC(), THCon(), EndTHC(), THCOFF(), ZeroTHC(), ResetTHC()

FeedRate
Sub FeedRate() As Double

This function gets the current feed rate. Note that Feed rate is specified in units per minute.

Arguments:  
Feed rate specified in units/minute, as a Double

Return Value:  
None

Example:
' Set the feed rate to 123.456 inches/minute
SetFeedRate(123.456 / 60)
' Get the current feed rate, in inches/minute, and
display it
CurrentFeedrate = FeedRate()
' Display it on the status line
Message “Current feed rate = “ & CurrentFeedrate

See also:
SetFeedrate()

FileName
Function FileName() As String

This function returns the filename and path of the currently loaded G-Code file, if any. If
no file is currently loaded, the string “No File Loaded.” is returned instead.

Arguments:
None

Return Value:
Current G-Code file name and path, or “No File Loaded.”

Example:
' Show user current G-Code file name and path
Message “Current file is: “ & FileName()

See also:
LoadFile(), LoadRun()

GetABSPosition
Function GetABSPosition(Axis As Integer) As Double

This function returns the machine position of the specified axis.

Arguments:
Axis is the axis whose machine position is being requested as follows:
0 = X Axis
1 = Y Axis
2 = Z Axis
3 = A Axis
4 = B Axis
5 = C Axis
**Return Value:**
Machine position as a Double

**Example:**
```
' Define the axes
Dim Xaxis As Integer
Xaxis = 0
Dim Yaxis As Integer
Yaxis = 1
Dim Zaxis As Integer
Zaxis = 2

' Get Y Axis Machine Position
Dim AxisPos As Double
AxisPos = GetABSPosition(Yaxis)
' Put it on the status line
Message "Y Axis Machine Pos = " & AxisPos
```

**See also:**
SetMachZero(), MinX(), MaxX(), MinY(), MaxY()

---

**GetACoor**

Function GetACoor() As Double

This function is used in conjunction with the GetCoord() function to get X, Y, Z and A axis coordinate values from the user. The GetACoor() function will return the A value entered by the user in the last GetCoord() function call.

**Arguments:**
None

**Return Value:**
A Axis coordinate value from last GetCoord() call

**Example:**
```
GetCoord("Enter target coordinates:")
Message "Coordinates are: " & GetXCoor() & " " & GetYCoor() & " " & GetZCoor() & " " & GetACoor()
```
GetActiveProfileDir
Function GetActiveProfileDir() As string

This function is used to retrieve the string with the full path of the active profile.

Arguments:
None

Return Value:
The path to the active profile.

Example:
MsgBox "The Running Profile path is " & GetActiveProfileDir()

See also:
GetActiveProfileDir

First Mach3 version with API:
This API was first implemented in Mach3 version 3.43.06.

GetActiveProfileName
Function GetActiveProfileName() As string

This function is used to retrieve the name of the currently running profile.

Arguments:
None

Return Value:
The name of the currently active profile.

**Example:**
```
MsgBox “The Running Profile is: “ & GetActiveProfileName()
```

**See also:**
GetActiveProfileDir

**First Mach3 version with API:**
This API was first implemented in Mach3 version 3.43.06.

### GetActiveScreenSetName

Function `GetActiveScreenSetName()` As string

This function is used to retrieve the name of the currently active Screen set.

**Arguments:**
None

**Return Value:**
The name of the currently active screen set.

**Example:**
```
MsgBox “The Running screen set is: “ & GetActiveScreenSetName()
```

**See also:**
n/a

**First Mach3 version with API:**
This API was first implemented in Mach3 version 3.43.06.

### GetCoord

Sub `GetCoord(Prompt As String)`

This function displays a dialog box containing the Prompt string, along with four textboxes, labeled X, Y, Z and A, into which the user can enter four coordinate values. The values are stored in variables within Mach3 which can be retrieved using the `GetXCoor()`, `GetYCoor()`, `GetZCoor()` and `GetACoor()` functions.

**Arguments:**
Prompt is a String that is displayed in the dialog box.
Return Value:
None

Example:
GetCoord("Enter target coordinates:"
Message "Coordinates are: " & GetXCoor() & " " & GetYCoor() & " " & GetZCoor() & " " & GetACoor()

See also:
GetXCoor(), GetYCoor(), GetZCoor(), GetACoor()
**GetDRO**

Function GetDRO(DRONum As Integer) As Double

This legacy function takes the DRO number passed as its argument, and returns the value of the Mach DRO of that number.

The use of GetDRO is no longer recommended practice and this function exists only to support preexisting legacy scripts. This function is deprecated, and its use is **strongly discouraged**.

**Legacy script note:** Over time, there have been two different DRO numbering schemes used with Mach; the “DRO number” series and the “OEMDRO number” series. This function uses the “DRO number” series.

The “DRO number” series was further subdivided into “User” and “OEM” ranges. Within the “OEM” range, valid DRONums were from 0 to 200, which, at one time, corresponded to OEM DRO numbers 800 to 1000.

The numerical correspondence between the numbering series is **not guaranteed** for future releases of Mach.

Use the GetOEMDRO and GetUserDRO functions instead of this function.

**Arguments:**

DRONum is the DRO number to read. The value has to be within the “DRO number series”.

**Return Value:**

Contents of DRO DRONum

**Example:**

```
' Define the axes
Const XaxisMultiFunctionDRONum = 0
Const YaxisMultiFunctionDRONum = 1
Const ZaxisMultiFunctionDRONum = 2

' Read the Z axis DRO
MsgBox "Using GetDRO() Z Axis DRO reads: " & GetDRO(ZaxisMultiFunctionDRONum)
```

**See also:**

SetOEMDRO(), GetOEMDRO(), SetUserDRO(), GetUserDRO()
**GetDROString**

Function GetDROString(DRONum As Integer) As String

This legacy function takes the DRO number passed as its argument and returns the value of the Mach DRO of that number, rounded to four decimal places, and formatted as a String.

The use of GetDROString is no longer recommended practice and this function exists only to support preexisting legacy scripts. This function is deprecated, and its use is **strongly discouraged**.

*Legacy script note:* Over time, there have been two different DRO numbering schemes used with Mach; the “DRO number” series and the “OEMDRO number” series. This function uses the “DRO number” series.

The “DRO number” series was further subdivided into “User” and “OEM” ranges. Within the “OEM” range, valid DRONums were from 0 to 200, which, at one time, corresponded to OEM DRO numbers 800 to 1000.

The numerical correspondence between the numbering series is **not guaranteed** for future releases of Mach.

Use of the GetOEMDRO and GetUserDRO functions instead; then use cStr to converts the numerical value to a string.

**Arguments:**

DRONum is the DRO number to read.

**Return Value:**

Contents of DRO DRONum

**Example:**

```
' Define the axes
Const XaxisMultiFunctionDRONum = 0
Const YaxisMultiFunctionDRONum = 1
Const ZaxisMultiFunctionDRONum = 2

Dim ZPositionString as String

' Read the Z axis DRO
ZPosition = cStr(GetDRO(ZaxisMultiFunctionDRONum))
```
MsgBox “Using GetDRO() Z Axis DRO reads: “ & ZPositionString

See also:
SetOEMDRO(), GetOEMDRO(), SetUserDRO(), GetUserDRO()

GetIJMode
Function GetIJMode() As Integer

This function returns the current IJ mode (absolute/incremental), as set in Config->GeneralConfig.

Arguments:
None

Return Value:
0 indicates absolute IJ mode is enabled
1 indicates incremental IJ mode is enabled

Example:
' Show user the current IJ mode
If GetIJMode() Then
   Message "IJ Mode is incremental"
Else
   Message "IJ Mode is absolute"
End If

See also:
SetIJMode()

GetLED
Function GetLED(LEDNum As Integer) As Integer

This legacy function takes the LED number passed as its argument, and returns the value of the Mach LED of that number.

The use of GetLED is no longer recommended practice and this function exists only to support preexisting legacy scripts. This function is deprecated, and its use is strongly discouraged.
Legacy script note: Over time, there have been two different LED numbering schemes used with Mach; the “LED number” series and the “OEMLED number” series. This function uses the “LED number” series.

The “LED number” series was further subdivided into “User” and “OEM” ranges. Within the “OEM” range, valid LEDNums were from 0 to 55, which, at one time, corresponded to OEM LED numbers 800 to 855.

The numerical correspondence between the numbering series is not guaranteed for future releases of Mach.

Use the GetOEMLED and GetUserLED functions instead of this function.

**Arguments:**
LEDNum is the OEM LED number to read.

**Return Value:**
0 indicates LEDNum is currently turned off
1 indicates LEDNum is currently turned on

**Example:**
```
define the LEDs
const InchModeLED = 1
const mmModeLED = 2

' are we in inch or metric mode?
code "G20"
code "G21"

if getLED(inchModeLED) then
  ' we are in inch mode
  message "inch mode"
end if

if getLED(mmModeLED) then
  ' we are in mm mode
  message "mm mode"
end if
```

**See also:**
GetOEMLED(), SetUserLED(), GetUserLED()
**GetLoadedGCodeDir**

Function GetLoadedGCodeDir() As string

This function is used to retrieve the string with the full path to the loaded g-code file name.

*Arguments:*
None

*Return Value:*
The path to the loaded G-Code file. If no Gcode is loaded the function returns a null string ("").

*Example:*
```
MsgBox "The G-Code file path is " & GetLoadedGCodeDir()
```

*See also:*
GetLoadedGCodeFileName

*First Mach3 version with API:*
This API was first implemented in Mach3 version 3.43.06.

---

**GetLoadedGCodeFileName**

Function GetLoadedGCodeFileName() As string

This function is used to retrieve the name of the loaded g-code file.

*Arguments:*
None

*Return Value:*
The name of the loaded G-Code file. The extension is included in the string. If no Gcode is loaded the function returns a null string ("").

*Example:*
```
MsgBox "The G-Code file name is " & GetLoadedGCodeFileName()
```

*See also:*
GetLoadedGCodeDir

*First Mach3 version with API:*
The API was first implemented in Mach3 version 3.43.06.
**GetMachVersion**

Function GetMachVersion(ByRef Major as Integer, ByRef Minor as Integer, ByRef Build as Integer) As Boolean

This function returns the full path to the Mach3 installation folder.

**Arguments:**

*The functions arguments are passed by reference and used to return the Mach version numbers.*

- Major: Major Version number
- Minor: Minor version number
- Build: build Version number

You must declare (DIM) the variables you pass to the function as integers. Non declared variables are created as type Var which can’t be passed by reference.

**Return Value:**

- True: the returned version values are valid
- False: an error occurred processing the GetMachVersion call; the version values may not be valid.

**Example:**

```vbscript
If (GetMachVersion(Major, Minor, Build) <> true) Then
    MsgBox("Unable to get Version info")
Else
    MsgBox("Mach3 version = " & Major & "." & Minor & "." & Build )
End If
```

**See Also:**

n/a

**First Mach3 version with API:**

This API was first implemented in Mach3 version 3.42.30.

---

**GetMainFolder**

Function GetMainFolder() As String

This function returns the full path to the Mach3 installation folder.
GetMainFolder
Function GetMainFolder() as String

Arguments:
None

Return Value:
String full file system path to Mach3 installation folder

Example:
' Show the user where Mach3 is installed
Message “Mach3 is installed at: “ & GetMainFolder()

See also:
GetMyWindowsHandle

GetMyWindowsHandle
Function GetMyWindowsHandle() as Long

This function is used to retrieve the windows handle for the Mach3 window. This is
useful for passing to OS calls which require the callers windows handle as a parameter.

Arguments:
None

Return Value:
The Mach 3 windows handle

Example:
MsgBox “My Windows handle as number is “ &
GetMyWindowsHandle()

See also:
n/a

First Mach3 version with API:
This API was first implemented in Mach3 version 3.43.06.

GetOEMDRO
Function GetOEMDRO(DRONum As Integer) As Double

This function returns the value of OEM DRO DRONum.

Arguments:
DRONum must be a valid OEM DRO number.

Return Value:
Returns a Double value of the specified DRO

**Example:**
```
' Define the axes
Const XaxisMultiFunctionOEMDRONum = 800
Const YaxisMultiFunctionOEMDRONum = 801
Const ZaxisMultiFunctionOEMDRONum = 802

' Write 1.2345 to Z axis DRO using SetOEMDRO
SetOEMDRO(ZaxisMultiFunctionDRONum, 1.2345)

' Show the user the Z Axis DRO value, using
GetOEMDRO()
MsgBox "After using SetOEMDRO() the Z Axis DRO reads:
" & GetOEMDRO(ZaxisMultiFunctionDRONum)
```

See also:
SetOEMDRO(), SetUserDRO(), GetUserDRO()

**GetOEMLED**
Function GetOEMLED(LEDNum As Integer) As Integer

This function returns the value of OEM LED LEDNum.

**Arguments:**
LEDNum must be a valid OEM LED number.

**Return Value:**
Returns an Integer value representing the current state of the specified LED. 0 indicates the LED of off (unlit), 1 indicated the LED is on (lit).

**Example:**

See also:
SetOEMLED(), SetUserLED(), GetUserLED()

**GetPage**
Function GetPage() As Integer

This function returns the number of the currently active screen set page.
Arguments:
None

Return Value:
Current screen set page number, as Integer

Example:
' Make sure user is on Diagnostics page
If Not GetPage() = 5 Then
    MsgBox “Please switch to Diagnostics page…”
End If

See also:
SetPage()

GetParam
Function GetParam(ParamName As String) As Double

This function allows a number of Mach3 internal parameters (not to be confused with G-code parameters) to be read. Each Mach3 parameter is identified by name. The current value of the parameter whose name is given by ParamName is returned as a Double. Valid parameters are:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMachine</td>
<td>X axis machine position</td>
</tr>
<tr>
<td>YMachine</td>
<td>Y axis machine position</td>
</tr>
<tr>
<td>ZMachine</td>
<td>Z axis machine position</td>
</tr>
<tr>
<td>Encoder1</td>
<td>Encoder1 Count</td>
</tr>
<tr>
<td>Encoder2</td>
<td>Encoder2 Count</td>
</tr>
<tr>
<td>Encoder3</td>
<td>Encoder3 Count</td>
</tr>
<tr>
<td>Encoder4</td>
<td>Encoder4 Count</td>
</tr>
<tr>
<td>MPG1</td>
<td>MPG1 Count</td>
</tr>
<tr>
<td>MPG2</td>
<td>MPG2 Count</td>
</tr>
<tr>
<td>MPG3</td>
<td>MPG3 Count</td>
</tr>
<tr>
<td>XScale</td>
<td>X axis scale factor</td>
</tr>
<tr>
<td>YScale</td>
<td>Y axis scale factor</td>
</tr>
<tr>
<td>ZScale</td>
<td>Z axis scale factor</td>
</tr>
<tr>
<td>AScale</td>
<td>A axis scale factor</td>
</tr>
<tr>
<td>BScale</td>
<td>B axis scale factor</td>
</tr>
<tr>
<td>CScale</td>
<td>C axis scale factor</td>
</tr>
<tr>
<td>FeedRate</td>
<td>Feed rate</td>
</tr>
<tr>
<td>Units</td>
<td>Current units (inch/mm). 0 = mm, 1 = inch</td>
</tr>
<tr>
<td>StepsPerAxisX</td>
<td>X axis steps per unit</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>StepsPerAxisY</td>
<td>Y axis steps per unit</td>
</tr>
<tr>
<td>StepsPerAxisZ</td>
<td>Z axis steps per unit</td>
</tr>
<tr>
<td>StepsPerAxisA</td>
<td>A axis steps per unit</td>
</tr>
<tr>
<td>StepsPerAxisB</td>
<td>B axis steps per unit</td>
</tr>
<tr>
<td>StepsPerAxisC</td>
<td>C axis steps per unit</td>
</tr>
<tr>
<td>VelocitiesX</td>
<td>X axis maximum velocity, from motor tuning, in units/second</td>
</tr>
<tr>
<td>VelocitiesY</td>
<td>Y axis maximum velocity, from motor tuning, in units/second</td>
</tr>
<tr>
<td>VelocitiesZ</td>
<td>Z axis maximum velocity, from motor tuning, in units/second</td>
</tr>
<tr>
<td>VelocitiesA</td>
<td>A axis maximum velocity, from motor tuning, in units/second</td>
</tr>
<tr>
<td>VelocitiesB</td>
<td>B axis maximum velocity, from motor tuning, in units/second</td>
</tr>
<tr>
<td>VelocitiesC</td>
<td>C axis maximum velocity, from motor tuning, in units/second</td>
</tr>
<tr>
<td>AccelerationX</td>
<td>X axis maximum acceleration, from motor tuning</td>
</tr>
<tr>
<td>AccelerationY</td>
<td>Y axis maximum acceleration, from motor tuning</td>
</tr>
<tr>
<td>AccelerationZ</td>
<td>Z axis maximum acceleration, from motor tuning</td>
</tr>
<tr>
<td>AccelerationA</td>
<td>A axis maximum acceleration, from motor tuning</td>
</tr>
<tr>
<td>AccelerationB</td>
<td>B axis maximum acceleration, from motor tuning</td>
</tr>
<tr>
<td>AccelerationC</td>
<td>C axis maximum acceleration, from motor tuning</td>
</tr>
<tr>
<td>SpindleSpeed</td>
<td>Should modify Spindle Speed, but does not work in all versions. User SetSpinSpeed() instead.</td>
</tr>
<tr>
<td>ZInhibitOn</td>
<td>Z Inhibit Enable. 0=Z inhibit disabled, 1=Z inhibit enabled. When Z inhibit is enabled, the Z axis will not be allowed to move below the depth specified by the ZinhibitDepth parameter.</td>
</tr>
<tr>
<td>ZInhibitDepth</td>
<td>Z Inhibit Depth. When Z inhibit is enabled, the Z axis will not be allowed to move below the depth specified by the ZinhibitDepth parameter.</td>
</tr>
<tr>
<td>SafeZ</td>
<td>SafeZ height</td>
</tr>
<tr>
<td>XDRO</td>
<td>X axis DRO</td>
</tr>
<tr>
<td>YDRO</td>
<td>Y axis DRO</td>
</tr>
<tr>
<td>ZDRO</td>
<td>Z axis DRO</td>
</tr>
<tr>
<td>ADRO</td>
<td>A axis DRO</td>
</tr>
<tr>
<td>BDRO</td>
<td>B axis DRO</td>
</tr>
<tr>
<td>CDRO</td>
<td>C axis DRO</td>
</tr>
<tr>
<td>Boundry</td>
<td>Toolpath Boundaries display enable. 0=&gt;disable boundaries display, 1=&gt;enable boundaries display</td>
</tr>
<tr>
<td>XRefPer</td>
<td>X axis homing speed, as % of rapid speed</td>
</tr>
<tr>
<td>YRefPer</td>
<td>Y axis homing speed, as % of rapid speed</td>
</tr>
<tr>
<td>ZRefPer</td>
<td>Z axis homing speed, as % of rapid speed</td>
</tr>
<tr>
<td>ARefPer</td>
<td>A axis homing speed, as % of rapid speed</td>
</tr>
<tr>
<td>BRefPer</td>
<td>B axis homing speed, as % of rapid speed</td>
</tr>
<tr>
<td>CRefPer</td>
<td>C axis homing speed, as % of rapid speed</td>
</tr>
<tr>
<td>TotalHours</td>
<td>Running count of total Mach3 up-time</td>
</tr>
</tbody>
</table>

**Arguments:**
ParamName is the String name of the parameter to be returned. This must be one of the above names.

Return Value:
Current value of the requested parameter as a Double

Example:
' Get the new scale factor from the user
ScaleFactor = Question "Enter new scale factor:"
' Set the new scale factor for X/Y/Z
SetParam("Xscale", ScaleFactor)
SetParam("Yscale", ScaleFactor)
SetParam("Zscale", ScaleFactor)

See also:
SetParam()

GetPortByte
Function GetPortByte(PortAddr As Integer) As Integer

This function reads the 8-bit PC I/O port whose address is given by PortAddr, and returns the 8-bit data value read from the port as an unsigned integer value. This function can be used for reading hardware devices not directly supported by Mach3.

Note that this function is available only when the parallel port driver is loaded.

Arguments:
PortAddr is the Integer address of the port to be read

Return Value:
Unsigned 8-bit integer value read from the port.

Example:
' Our port address
PortAddr = 1016 ' 0x3f8
' Read data register of parallel port at 0x3f8
PortData = GetPortByte(PortAddr)

See also:
PutPortByte()

GetToolParam
Function GetToolParam(ToolNum As Integer, ParamNum As Integer)
This function allows any tool parameter, except the description text, for any tool to be read. ToolNum is the number of the tool whose parameters are being set, and can be from 1 to 255. ParamNum is a parameter number, defined as follows:

For Mach3Mill:
1 = Diameter
2 = Z Offset
3 = X Wear
4 = Z Wear

For Mach3Turn:
1 = Tip Type
2 = Tool Radius
3 = X Offset
4 = Z Offset
5 = X Wear
6 = Z Wear
7 = Turret Angle

**Arguments:**
ToolNum is an Integer tool number, and must be between 1 and 255.

**Return Value:**
Requested parameter value, as a Double

**Example:**
```
' Define some constants
DiameterParam = 1
ZoffsetParam = 2
XwearParam = 3
ZwearParam = 4

' Display tool #23 parameters
Diam = GetToolParam(23, DiameterParam)
Length = GetToolParam(23, ZoffsetParam)
Xwear = GetToolParam(23, XwearParam)
Zwear = GetToolParam(23, ZwearParam)
Desc = GetToolDesc(23)
```

**See also:**
SetToolParam(), GetToolDesc()
GetRPM
Function GetRPM() As Double

This function returns the currently commanded spindle speed (S-word) as a Double. Note that this returns the most recent S-word value, and not the actual spindle RPM.

Arguments:
None

Return Value:
Currently commanded spindle speed as a Double.

Example:
' Show current S-word
Message “S-Word = “ & GetRPM()

See also:
SetSpinSpeed(), DoSpinCW(), DoSpinCCW(), DoSpinStop()

GetSafeZ
Function GetSafeZ() As Double

This function returns the current SafeZ height.

Arguments:
None

Return Value:
Current SafeZ height as a Double.

Example:
' Get current SafeZ height
OldSafeZ = GetSafeZ()
' Set new SafeZ height
NewSafeZ = 1.5750
SetSafeZ(NewSafeZ)
...
' Restore old SafeZ height
SetSafeZ(OldSafeZ)

See also:
SetSafeZ()

GetScale
Function GetScale(Axis As Integer) As Double
This function returns the current scale factor for axis \textit{Axis}.

\textbf{Arguments:}
Axis is the Integer Axis. 0=X, 1=Y, 2=Z, 3=A, etc.

\textbf{Return Value:}
Current scale factor for specified axis, as a Double

\textbf{Example:}
\begin{verbatim}
' Define some constants
Xaxis = 0
Yaxis = 1
Zaxis = 2
' Get the current axis scale factors
OldXScaleFactor = GetScale(Xaxis)
OldYScaleFactor = GetScale(Yaxis)
OldZScaleFactor = GetScale(Zaxis)
' Get the new scale factor from the user
ScaleFactor = Question("Enter new scale factor:"
' Set new scale factors for X/Y/Z
SetScale(Xaxis, ScaleFactor)
SetScale(Yaxis, ScaleFactor)
SetScale(Zaxis, ScaleFactor)
\end{verbatim}

\textbf{See also:}
SetScale()

\textbf{GetSelectedTool}
Function GetSelectedTool() As Integer

This function returns the tool specified by the most recent tool change (M6) command. This function is typically used in the M6Start macro to make the selected tool the current tool.

\textbf{Arguments:}
None

\textbf{Return Value:}
Selected tool as an Integer

\textbf{Example:}
\begin{verbatim}
' Sample M6Start macro
' Get selected tool
NewTool = GetSelectedTool()
\end{verbatim}
Make it the current tool
SetCurrentTool(NewTool)

See also:
GetCurrentTool(), SetCurrentTool()

GetSetupUnits

Function GetSetupUnits() As Integer

This function returns the native setup units of the machine. The native setup units are a
characteristic of machine and the return value of this API does not change with the use of
G20/G21.

Arguments:
None

ReturnValue:
0 = mm units
1 = inch units

Example:
' display the machine’s setup units
Case Select GetSetupUnits()
Case 0:
    MsgBox “Setup units are millimeters”
Case 1:
    MsgBox “Setup units are inches”
Else:
    MsgBox “Unknown setup units value”
End case

See also:
n/a

First Mach3 version with API:
This API was first implemented in Mach3 version 3.43.06.

GetTimer

Function GetTimer(TimerNum As Integer) As Double

This function returns the current count for the specified timer. Mach3 provides 25 timers,
numbered 0 to 24, which can be used for timing in CB scripts. To time an event, first
clear the timer using SetTimer(), then use GetTimer() to read the timer. Note that this function works only with the parallel port driver, and support for this function may be removed without notice in a future release.

**Arguments:**
TimerNum is an Integer timer number, which must be between 0 and 24.

**Return Value:**
Double value of timer TimerNum.

**Example:**
```
' Clear timer 15
SetTimer(15)
' Wait for OEM Trigger 10 to go active
While IsActive(OEMTRIG10) = False Then
    Sleep 10
Wend
' See how long it took
Message “OEMTRIG10 active after “ & GetTimer(15) & “ seconds”
```

**See also:**
SetTimer()

**GetToolChangeStart**
Function GetToolChangeStart(Axis As Integer) As Double

This function returns the position of the specified axis at the time a tool change started. This is typically used in an M6End macro to restore the axis positions to the positions they were in before the tool change.

**Arguments:**
Axis is the Integer Axis. 0=X, 1=Y, 2=Z, 3=A, etc.

**Return Value:**
Double position of the specified axis at the start of the last tool change.

**Example:**
```
' Example M6End macro
' Define some constants
Xaxis = 0
Yaxis = 1
Zaxis = 2
' Move all axes back to where they were before the tool change
Xpos = GetToolChangeStart(Xaxis)
Ypos = GetToolChangeStart(Yaxis)
```
Zpos = GetToolChangeStart(Zaxis)
Code “G00 X” & Xpos & “ Y” & Ypos & “ Z” & Zpos
' Wait for move to complete
While IsMoving()
    Sleep 100
Wend

See also:
GetSelectedTool(), GetCurrentTool(), SetCurrentTool()

**GetToolDesc**
Function GetToolDesc(ToolNum As Integer) As String

This function returns the tool descriptor text for the tool specified by ToolNum.

*Arguments:
ToolNum is an Integer tool number, from 1 to 255

*Return Value:
Tool table description text for specified tool, as a String

*Example:
' Show user the current tool description
Message “Tool “ & GetCurrentTool() & “: “ & _
GetToolDesc(GetCurrentTool())

See also:
GetToolParam(), SetToolParam(), SetToolDesc()

**GetTurretAng**
Function GetTurretAng() As Double

This function returns the current lathe tool turret angle.

*Arguments:
None

*Return Value:
Current lathe tool turret angle, as a Double

*Example:
' Display the current tool turret angle
Message “Tool turret position = “ & GetTurretAng()
See also:

**GetUserDRO**
Function GetUserDRO(DRONum As Integer) As Double

This function returns the value of User DRO DRONum as a Double.

**Arguments:**
DRONum is the Integer User DRO number to be set. Valid User DRO numbers range from 1000-2254.

**Return Value:**
Current value of UserDRO DRONum as a Double

**Example:**

```
' Define some constants
MyWidgetDRO = 1125
' Set MyWidgetDRO to 1.234
SetUserDRO(MyWidgetDRO, 1.234)
...
' Get current value of MyWidgetDRO
MyDROVal = GetUserDRO(MyWidgetDRO)
```

See also:
GetUserDRO(), SetOEMDRO(), GetOEMDRO()

**GetUserLabel**
Function GetUserLabel(LabelNum As Integer) As String

This function allows the user to retrieve the value of an on-screen User label. “User” labels are those that are created in the screen designer with the default text containing the String “UserLabel” followed by one or more digits.

**Arguments:**
LabelNum is the numeric portion of the user label default text. LabelNum must be between 0 and 255.

**Return Value:**
None

**Example:**

```
' Change the text in UserLabel25
SetUserLabel(25, “This is Label 25”)
```
See also:
SetUserLabel()

GetUserLED
Function GetUserLED(LEDNum As Integer) As Integer

This function allows the current state of a User LED to be retrieved.

Arguments:
LEDNum is the User LED whose state is to be retrieved, which must be in the range of 1000 to 2254

Return Value:
Current state of the specified User LED. 0 indicates the LED is off (unlit), 1 indicates the LED is on (lit).

Example:
' Define some constants
FluxCapacitorControl = OUTPUT1  ' Output that controls the flux capacitor
FluxCapacitorLED = 1234  ' LED that indicates flux capacitor is active
' Turn on the Flux capacitor
ActivateSignal(FluxCapacitorControl)
' Turn on the Flux Capacitor LED for the operator
SetUserLED(FluxCapacitorLED, 1)
...
' Is the Flux Capacitor on?
FluxCapacitorOn = GetUserLED(FluxCapacitorLED)

See also:
GetUserLED(), SetOEMLED(), GetOEMLED()

GetVar
Function GetVar(VarNum As Integer) As Double
This function returns the current value of the Mach variable specified by VarNum as a Double. Mach variables are accessible both to CB scripts, using the SetVar() and GetVar() functions, as well as G-code programs, using the #nnnn syntax.

**Arguments:**
VarNum is Integer the number of the Mach variable to be retrieved.

**Return Value:**
Current value of the specified variable, as a Double.

**Example:**
' Set a variable 1234 to our target position of 2.3456
SetVar(1234, 2.3456)
' Now move X to our target position
Code “G0 X #1234”
' Or, another way...
Code “G0 X “ & GetVar(1234)

**See also:**
SetVar()

---

**GetXCoor**

Function GetXCoor() As Double

This function is used in conjunction with the GetCoord() function to get X, Y, Z and A axis coordinate values from the user. The GetXCoor() function will return the X value entered by the user in the last GetCoord() function call.

**Arguments:**
None

**Return Value:**
X Axis coordinate value from last GetCoord() call

**Example:**
GetCoord(“Enter target coordinates:”)
Message “Coordinates are: “ & GetXCoor() & “ “ & GetYCoor() & “ “ & GetZCoor() & “ “ & GetACoor()
See also:
GetCoord(), GetYCoor(), GetZCoor(), GetACoor()

GetYCoor
Function GetYCoor() As Double

This function is used in conjunction with the GetCoord() function to get X, Y, Z and A axis coordinate values from the user. The GetYCoor() function will return the Y value entered by the user in the last GetCoord() function call.

Arguments:
None

Return Value:
Y Axis coordinate value from last GetCoord() call

Example:
GetCoord(“Enter target coordinates:”)
Message “Coordinates are:” & GetXCoor() & “ “ & GetYCoor() & “ “ & GetZCoor() & “ “ & GetACoor()
GetCoord(), GetXCoor(), GetZCoor(), GetACoor()

**GetZCoor**
Function GetZCoor() As Double

This function is used in conjunction with the GetCoord() function to get X, Y, Z and A axis coordinate values from the user. The GetZCoor() function will return the Z value entered by the user in the last GetCoord() function call.

**Arguments:**
None

**Return Value:**
Z Axis coordinate value from last GetCoord() call

**Example:**
GetCoord(“Enter target coordinates:”)
Message “Coordinates are: “ & GetXCoor() & “ “ & GetYCoor() & “ “ & GetZCoor() & “ “ & GetACoor()

See also:
GetCoord(), GetXCoor(), GetYCoor(), GetACoor()

**GotoSafeZ**
Sub GotoSafeZ()

This function will move the Z axis to the Safe_Z position, if Safe_Z is enabled in Config->Safe_Z Setup. If Safe_Z is not enabled, an error message will be displayed on the status line, and no move takes place.
Arguments:
None

Return Value:
None

Example:
```
' Move Z axis to Safe_Z position
GotoSafeZ()
```

See also:
GetSafeZ(), SetSafeZ()

HelpAbout
Sub HelpAbout()
This function displays a dialog box showing the current version of the scripting engine.

Arguments:
None

Return Value:
None

Example:
```
' Show the user the scripting engine version
HelpAbout()
```

See also:

IncludeTLOinZFromG31
Function IncludeTLOinZFromG31() As Boolean

This function is used to retrieve the state of the Mach menu config/general->config
dialog option which determines if Mach includes the calculation of the current tool’s
TLO value in the Z coordinate value returned by a G31 probing operation.

Arguments:
None

Return Value:
True: The option is checked
False: the option is not checked
**Example:**

MsgBox “TLO in G31 option is “ & IncludeTLOinZFromG31()

**See also:**

n/a

**First Mach3 version with API:**

This API was first implemented in Mach3 version 3.43.06.

**IsActive**

Function IsActive(Signal As Integer) As Boolean

This function returns a Boolean True if the current state of the specified input signal is its active state. Note that in terms of actual signal level, the term “active” depends on how the signal is defined. If the signal is defined in Config->Ports&Pins->InputSignals as ActiveLow, then IsActive() will return True when that signal is being driven to a logic low. If the signal is defined in Config->Ports&Pins->InputSignals as ActiveHigh, then IsActive() will return True when that signal is being driven to a logic high.

**Arguments:**

Signal is an integer value corresponding to one of pre-defined input signal constants.

**Return Value:**

False indicates the signal is currently in its inactive state

True indicates the signal is currently in its active state

**Example:**

`Show the user the state of the INPUT #1 input
If IsActive(INPUT1) Then
    Message “INPUT #1 input is active”
Else
    Message “INPUT #1 input is inactive”
End If`

**See also:**

**IsDiameter**

Function IsDiameter() As Integer

This lathe-only function returns 1 if Mach3 is currently operating in Diameter mode.

**Arguments:**
Return Value:

0 indicates Mach3 is currently operating in Radius mode
1 indicates Mach3 is currently operating in Diameter mode

Example:

' Tell the user what mode we’re in
If IsDiameter() Then
    Message “Currently in Diameter mode”
Else
    Message “Currently in Radius mode”
End If

See also:

IsEStop

Function IsEStop() As Integer

This function returns 1 if Mach3 is currently in E-Stop mode.

Arguments:
None

Return Value:

0 indicates Mach3 is currently not in E-Stop mode
1 indicates Mach3 is currently in E-Stop mode

Example:

Dim FluxCapacitorControl As Integer
FluxCapacitorControl = OUTPUT3

' Are we in E-Stop?
If IsEStop() Then
    ' Yes, so turn off the flux capacitor
    DeactivateOutput(FluxCapacitorControl)
Else
    ' No, so turn on the flux capacitor
    ActivateOutput(FluxCapacitorControl)
End If

See also:
**IsLoading**

Function IsLoading() As Integer

This function returns 1 if a G-code file is currently being loaded.

**Arguments:**
None

**Return Value:**
- 0 indicates a G-code file is not currently being loaded
- 1 indicates a G-code file is currently being loaded

**Example:**

```vba
Dim FluxCapacitorControl As Integer
FluxCapacitorControl = OUTPUT3

' Are we in loading a new G-code file?
If IsLoading() Then
    ' Yes, so turn off the flux capacitor
    DeactivateOutput(FluxCapacitorControl)
Else
    ' No, so turn on the flux capacitor
    ActivateOutput(FluxCapacitorControl)
End If
```

**See also:**

**IsMoving**

Function IsMoving() As Integer

This function returns 1 if any axis is currently moving. This is most often used when commanding motion within a macro, to pause macro execution until the motion is complete. This is a complement to IsStopped().

**Arguments:**
None

**Return Value:**
- 0 if all axes currently stopped
- 1 if any axis is currently moving

**Example:**

```vba
' Move Z axis to Safe_Z position
GotoSafeZ()
' Wait for SafeZ move to complete
```
Sleep(100)
While IsMoving()
    Sleep(100)
Wend

See also:
IsStopped()

**IsOutputActive**
Function IsOutputActive(Signal As Integer) As Boolean

This function returns Boolean True if the current state of the specified output signal is its active state. Note that in terms of actual signal level, the term “active” depends on how the signal is defined. If the signal is defined in Config->Ports&Pins->OutputSignals as ActiveLow, then IsOutputActive() will return a True value when that signal is being driven to a logic low. If the signal is defined in Config->Ports&Pins->InputSignals as ActiveHigh, then IsOutputActive() will return a True value when that signal is being driven to a logic high.

**Arguments:**
Signal is an integer value corresponding to one of pre-defined output signal constants.

**Return Value:**
A Boolean value indicating the state of the specified output signal.

**Example:**
```
' Set OUTPUT1 to its active state
ActivateSignal(OUTPUT1)
' Show the user the state of the OUTPUT #1 input
If IsActive(OUTPUT1) Then
    MsgBox “OUTPUT #1 output is active”
Else
    MsgBox “OUTPUT #1 output is inactive”
End If
' Set OUTPUT1 to its inactive state
DeactivateSignal(OUTPUT1)
' Show the user the state of the OUTPUT #1 input
If IsActive(OUTPUT1) Then
    MsgBox “OUTPUT #1 output is active”
Else
    MsgBox “OUTPUT #1 output is inactive”
End If
```

See also:
**IsPeriodicScriptRunning**
Function IsPeriodicScriptRunning(ByVal ScriptQFN as String) as Boolean

This function is used to determine if a periodic script has been started.

**Arguments:**
- ScriptQFN: the string of the Qualified File Name (QFN) for the script to check. The QFN is relative to the Mach install directory. The QFN passed must be identical to the QFN used to start the periodic script.

**Return Value:**
- True = the Script is running.
- False = the script is not running.

**Example:**
```
' check if the oiler has been started
If IsPeriodicScriptRunning("OilerScript") then
    MsgBox "Oiler script is running."
Else
    MsgBox "Oiler script is not running."
End If
```

**See also:**
StartPeriodicScript, StopPeriodicScript,

**First Mach3 version with API:**
This API was first implemented in Mach3 version 3.43.06.

**IsSafeZ**
Function IsSafeZ() As Integer

This function returns 1 if Safe_Z is enabled in Config->Safe_Z Setup.

**Arguments:**
- None

**Return Value:**
0 indicates Safe_Z is not enabled in Config->Safe_Z Setup
1 indicates Safe_Z is enabled in Config->Safe_Z Setup

Example:
`' Show user IsSafeZ()
If IsSafeZ() Then
  Message "SafeZ is enabled"
Else
  Message "SafeZ is disabled"
End If

See also:

**IsStopped**
Function IsStopped() As Integer

This function returns 1 if all axes are currently stopped. This is most often used when commanding motion within a macro, to pause macro execution until the motion is complete. This is a complement to IsMoving().

**Arguments**:
None

**Return Value**:
0 if any axis is currently moving
1 if all axes are currently stopped

**Example**:
`' Move Z axis to Safe_Z position
GotoSafeZ()
' Wait for SafeZ move to complete
Sleep(100)
While Not IsStopped()
  Sleep(100)
Wend

See also:
IsMoving()

**IsSuchSignal**
Function IsSuchSignal(SignalID As Integer) As Integer
This function returns an Integer value indicating whether the specified signal is defined in Config->Ports&Pins. A 0 return value indicates the signal is not defined, while a non-zero return value indicates the signal is defined. This can be used, for example, to ensure a PROBE input is properly defined before trying to do probing.

**Arguments:**
SignalID must be one of the pre-defined Mach3 CB output signal constants (see CB Constants), or other value or expression that evaluates to one of those values.

**Return Value:**
Signal definition state, as an Integer. 0 => signal is not defined, 1 => signal is defined

**Example:**
```
' Is a YHOME input signal properly defined?
If IsSuchSignal(YHOME) = 0 Then
    Message “Error! No YHOME input is defined”
End If
```

**See also:**

**JogOff**
Function JogOff(Axis As Integer)

This function is used to stop jogging of the specified axis. It is typically used in conjunction with JogOn() to jog an axis under control of a script.

**Arguments:**
Axis is an Integer, specifying the axis for which to disable jogging. 0=X, 1=Y, 2=Z, 3=A, etc.

**Return Value:**
None

**Example:**
```
' Define some constants
Xaxis = 0
Yaxis = 1
Zaxis = 2
Plus = 0
Minus = 1
' Jog the Y axis in the minus direction for one second
JogOn(Yaxis, Minus)
Sleep(1000)
' Now stop it
JogOff(Yaxis)

See also:
JogOn()

JogOn
Function JogOn(Axis As Integer, Dir as Integer)

This function is used to start an axis jogging in a specified direction at the current default jog speed. Once the axis starts jogging, it will continue until stopped by means if the JogOff() function, execution of a Stop button command, E-stop, or hitting a limit.

Arguments:
Axis is an Integer, specifying the axis for which to disable jogging. 0=X, 1=Y, 2=Z, 3=A, etc.
Dir is the direction in which to jog. 0=+ direction, 1=– direction

Return Value:
None

Example:
' Define some constants
Xaxis = 0
Yaxis = 1
Zaxis = 2
Plus = 0
Minus = 1
' Jog the Y axis in the minus direction for one second
JogOn(Yaxis, Minus)
Sleep(1000)
' Now stop it
JogOff(Yaxis)

See also:
JogOff()

LoadFile
Sub LoadFile(FilePath As String)

This function loads the G-code file specified by FilePath.

Arguments:
FilePath is the full file system path to the G-code file to be loaded
**Return Value:**
None

**Example:**
```
' Load the roadrunner demo file
LoadFile("C:\Mach3\Gcode\roadrunner.tap")
' Now run it
RunFile()
```

**See also:**
RunFile()

---

**LoadRun**

Sub LoadRun(Filepath As String)

This function loads the G-code file specified by Filepath, then immediately begins execution. This is exactly equivalent to a LoadFile() followed by a RunFile().

**Arguments:**
Filepath is the String path to the G-code file to be run.

**Return Value:**
None

**Example:**
```
' Load and run the roadrunner demo file
LoadRun("C:\Mach3\Gcode\roadrunner.tap")
```

**See also:**
LoadFile(), RunFile(), Filename(), IsLoading()

---

**LoadStandardLayout**

Sub LoadStandardLayout()

This function re-loads the current default screen set. In most cases, this will be the currently loaded screen set. This is used primarily to re-load the default screen set when exiting a Wizard.

**Arguments:**
None

**Return Value:**
None
Example:
`' Load default screen set
LoadStandardLayout()
``

See also:
ToggleScreens()

LoadTeachFile
Sub LoadTeachFile()

This function loads the most recent Teach file into Mach3 for execution. The file must first have been opened or created using OpenTeachFile() or AppendTeachFile() during the current Mach3 session.

Arguments:
None

Return Value:
None

Example:
`' Create a new Teach File in Mach3\Gcode
MyTeachFile = “TeachMe.nc”
Err = OpenTeachFile(MyTeachFile)
If Err <> 0 Then
   ' Teach file created successfully
   ' Write some G Code to it
   Code “G00 X0 Y0”
   Code “G02 X0 Y0 I-1 J0 F40”
   ' Close the Teach file
   CloseTeachFile()
   ' Now load the teach file for execution
   LoadTeachFile()
Else
   ' OpenTeachFile failed
   Message “Unable to open Teach File”
End If
``

See also:
OpenTeachFile(), AppendTeachFile(), CloseTeachFile()
**LoadWizard**

Sub LoadWizard(WizardName As String)

This function loads and runs the specified wizard. Wizards reside in the Mach3\Addons directory, and each wizard consists of a number of files with a subdirectory of the Addons directory. The WizardName String passed as the argument to LoadWizard must be just the name of the top-level subdirectory for the wizard.

**Arguments:**
WizardName is the name of the Wizard to be run, as a String

**Return Value:**
None

**Example:**
`' Load and run the “Circular Pocket” Wizard
LoadWizard(“Circular Pocket”)`

**See also:**

**MachMsg**

Function MachMsg(Prompt As String, Title As String, DialogType As Integer) As Integer

This function display a dialog box with one of several different combinations of buttons, and waits for the user to click one of the buttons. The Title argument String is displayed in the title bar of the dialog. The Message argument String is displayed in the client portion of the dialog, above the buttons. Clicking on any button closes the dialog, and the return value of the function indicates which button the user clicked.

**Arguments:**
Message is the String to be displayed in the client area of the dialog, above the button(s)
Title is the String to be displayed in the title bar of the dialog
DialogType is an Integer value which defines which buttons will be displayed on the dialog as follows:

0 = OK button
1 = OK, Cancel buttons
2 = Abort, Retry, Ignore buttons
3 = Yes, No, Cancel buttons
4 = Yes, No buttons
5 = Retry, Cancel buttons
6 = Cancel, Try Again, Continue buttons

Return Value:
Integer value indicating which button the user clicked, as follows:

<table>
<thead>
<tr>
<th>Button</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>1</td>
</tr>
<tr>
<td>Cancel</td>
<td>2</td>
</tr>
<tr>
<td>Abort</td>
<td>3</td>
</tr>
<tr>
<td>Retry</td>
<td>4</td>
</tr>
<tr>
<td>Ignore</td>
<td>5</td>
</tr>
<tr>
<td>Yes</td>
<td>6</td>
</tr>
<tr>
<td>No</td>
<td>7</td>
</tr>
<tr>
<td>Try Again</td>
<td>10</td>
</tr>
<tr>
<td>Continue</td>
<td>11</td>
</tr>
</tbody>
</table>

Example:
```
' Define some constants for MachMsg dialog types
MachMsgTypeOK = 0
MachMsgTypeOKCancel = 1
MachMsgTypeAbortRetryIgnore = 2
MachMsgTypeYesNoCancel = 3
MachMsgTypeYesNo = 4
MachMsgTypeRetryCancel = 5
MachMsgTypeCancelTryAgainContinue = 6
' Define some constants for MachMsg return codes
MachMsgReturnOK = 1
MachMsgReturnCancel = 2
MachMsgReturnAbort = 3
MachMsgReturnRetry = 4
MachMsgReturnIgnore = 5
MachMsgReturnYes = 6
MachMsgReturnNo = 7
MachMsgReturnTryAgain = 10
MachMsgReturnContinue = 11
' Display an Abort/Retry/Ignore dialog
Ret = MachMsg("Flux Capacitor Failed to Start", 
             "Flux Capacitor Warning", 
             MachMsgTypeAbortRetryIgnore)
If Ret = MachMsgReturnAbort Then
    ' Handle Abort here
ElseIf Ret = MachMsgReturnRetry Then
    ' Handle Retry here
ElseIf Ret = MachMsgReturnIgnore Then
    ' Handle Ignore here
End If
```
See also:
Message(), Question(), AskTextQuestion(), GetCoord(),

MaxX
Function MaxX() As Double

This function returns the maximum X extent of the currently open G-code file as a Double

Arguments:
None

Return Value:
Maximum X extent of currently loaded G-code file as a Double

Example:
' Show user the current program extents
Message "MinX=" & MinX() & " MaxX=" & MaxX() & _
" MinY=" & MinY() & " MaxY=" & MaxY()

See also:
MinX(), MinY, MaxY()

MaxY
Function MaxY() As Double

This function returns the maximum Y extent of the currently open G-code file as a Double

Arguments:
None

Return Value:
Maximum Y extent of currently loaded G-code file as a Double

Example:
' Show user the current program extents
Message "MinX=" & MinX() & " MaxX=" & MaxX() & _
" MinY=" & MinY() & " MaxY=" & MaxY()

See also:
MaxX(), MinY, MaxY()

Message
Sub Message(MessageText As String)

This function displays MessageText on the status line.

Arguments:
MessageText is the String text to display on the status line.

Return Value:
None

Example:
   ' Display the current tool number on the status line
   Message "Current tool is " & GetCurrentTool()

See also:
Message(), Question(), Ask TextQuestion(), MachMsg(), GetCoord()

MinX
Function MinX() As Double

This function returns the minimum X extent of the currently open G-code file as a Double

Arguments:
None

Return Value:
Minimum X extent of currently loaded G-code file as a Double

Example:
   ' Show user the current program extents
   Message "MinX=" & MinX() & " MaxX=" & MaxX() & _
   " MinY=" & MinY() & " MaxY=" & MaxY()

See also:
MinX(), MinY, MaxY()
**MinY**
Function MinY() As Double

This function returns the minimum Y extent of the currently open G-code file as a Double

*Arguments:*
  None

*Return Value:*
  Minimum Y extent of currently loaded G-code file as a Double

*Example:*
  `Show user the current program extents`
  Message "MinX=" & MinX() & " MaxX=" & MaxX() & _
  " MinY=" & MinY() & " MaxY=" & MaxY()

*See also:*
  MinX(), MinY, MaxY()

**nFmt**
Function nFmt(Val As Double, Digits As Integer) As Double

This function rounds a Double value to the specified number of decimal places

*Arguments:*
  Val is the Double value to be rounded
  Digits is the number of digits right of the decimal place to round to

*Return Value:*
  Rounded value, as a Double

*Example:*
  `Round 1.23456789 to 4 decimal places`
  `Display it – will display as 1.2345`
  Message nFmt(1.23456789, 4)

*See also:*

**NotifyPlugins**
Sub NotifyPlugins(Event As Integer)
This function invokes the MyNotify method of all currently loaded Plug-ins, passing Event as the argument.

**Arguments:**
Event is an Integer value that will be passed as the argument to the MyNotify methods of all loaded plug-ins.

**Return Value:**
None

**Example:**

See also:

**NumberPad**
Function NumberPad (ByVal PadTitle As string) As Double

This function displays a number pad for data entry. The pad is large to allow easy use with touch screens. The number pad size can also be changed via dragging a border.

**Arguments:**
PadTitle is the string to display as the title of the number pad.

**Return Value:**
The value entered into the numberpad.

**Example:**
Option explicit

Dim d As Double
d = NumberPad("test num pad")
MsgBox "value entered = " & d

**First Mach3 version with API:**
This API was first implemented in Mach3 version 3.43.06.

**OpenDigFile**
Sub OpenDigFile()

This function opens a digitization log file. When OpenDigFile is executed a FileOpen dialog will be displayed, and the user can select an existing file, or enter a new file name,
to which digitization (probing) data points will be written. Once opened, G31 commands will cause the probe trigger position to be written to the digitization file. When digitization is complete, CloseDigFile() can be used to close the file.

**Arguments:**
None

**Return Value:**
None

**Example:**

**See also:**

---

**OpenTeachFile**

Function OpenTeachFile (Filename As String) As Integer

This function re-opens an existing Teach file at Gcode\Filename in the Mach3 directory, and appends any commands subsequently executed via MDI or Code() to that file, until CloseTeachFile() is executed.

**Arguments:**
F

**Return Value:**
A

**Example:**

```
' Create a new Teach File in Mach3\Gcode
MyTeachFile = “TeachMe.nc”
Err = OpenTeachFile(MyTeachFile)
If Err <> 0 Then
  ' Teach file created successfully
  ' Write some G Code to it
  Code “G00 X0 Y0”
  Code “G02 X0 Y0 I-1 J0 F40”
  ' Close the Teach file
  CloseTeachFile()
  ' Now load the teach file for execution
  LoadTeachFile()
Else
  ' OpenTeachFile failed
  Message “Unable to open Teach File”
End If
```
See also:
AppendTeachFile(), CloseTeachFile()

**Param1**

Function Param1() As Double

This function returns the value of the P parameter passed to an M-macro.

**Arguments:**
None

**Return Value:**
Double value passed a P parameter to M-macro

**Example:**
' This macro expects three arguments: P, Q, & R
' If put into an M-macro, and invoked via MDI, it will
display the argument values on the status line
' For example, if using M1200:
'   M1200 P1.234 Q2.345 R3.456
' Executing the above line to MDI will display:
'   P=1.234 Q=2.345 R=3.456
Parg = Param1()
Qarg = Param2()
Rarg = Param3()
Message “P=” & Parg & “P=” & Qarg & “R=” & Rarg

See also:
Param2(), Param3()

**Param2**

Function Param2() As Double

This function returns the value of the Q parameter passed to an M-macro.

**Arguments:**
None

**Return Value:**
Double value passed a Q parameter to M-macro

**Example:**
' This macro expects three arguments: P, Q, & R
'If put into an M-macro, and invoked via MDI, it will
display the argument values on the status line
For example, if using M1200:
  M1200 P1.234 Q2.345 R3.456
Executing the above line to MDI will display:
  P=1.234 Q=2.345 R=3.456
Parg = Param1()
Qarg = Param2()
Rarg = Param3()

See also:
Param1(), Param3()

**Param3**

Function Param3() As Double

This function returns the value of the R parameter passed to an M-macro.

Arguments:
None

Return Value:
Double value passed a R parameter to M-macro

Example:
  This macro expects three arguments: P, Q, & R
  If put into an M-macro, and invoked via MDI, it will
display the argument values on the status line
  For example, if using M1200:
    M1200 P1.234 Q2.345 R3.456
  Executing the above line to MDI will display:
    P=1.234 Q=2.345 R=3.456
  Parg = Param1()
  Qarg = Param2()
  Rarg = Param3()

See also:
Param1(), Param2()
**PlayWave**

Sub PlayWave(Filename As String)

This function plays a .WAV file through the PCs audio system. Filename gives the name of the WAV file to be played.

**Arguments:**
Filepath is the String file path to the WAV file to be played

**Return Value:**
None

**Example:**
PlayWave(“C:\WINDOWS\Media\Windows XP Startup.wav”)

**See also:**

**ProgramSafetyLockout**

Function ProgramSafetyLockout() As Boolean

This function is used to retrieve the state of the Mach menu config/general-config dialog option which determines if input 1 is being used to control the Mach G-Code inhibit feature.

**Arguments:**
None

**Return Value:**
True: The option is checked
False: the option is not checked

**Example:**
MsgBox “Program Safety Lockout “ & ProgramSafetyLockout()

**See also:**
n/a

**First Mach3 version with API:**
This API was first implemented in Mach3 version 3.43.06.

**PutPortByte**

Sub PutPortByte(PortAddr As Integer, Value As Integer)
This function writes the 8-bit data given by Value to the 8-bit PC I/O port whose address is given by PortAddr. This function can be used for writing hardware devices not directly supported by Mach3.

**Arguments:**
- PortAddr is the Integer address of the port to be written
- Value is the Integer value to be written. Only the 8 least significant bits (LSBs) are written.

**Return Value:**
None

**Example:**
```
' Our port address
PortAddr = 1016 ' 0x3f8
PortData = 154 ' 0xa5
' Write 0xa5 to data register of parallel port at 0x3f8
PortData = PutPortByte(PortAddr, PortData)
```

**See also:**
GetPortByte()

**Question**
Function Question(Prompt As String) As Double

This function displays a dialog box with an OK button, and a text box into which the user can enter a numeric value. The Prompt string is displayed above the test box. This value is returned as a Double. This can be used for getting a single numeric value from the user.

**Arguments:**
- Prompt is the prompt string displayed in the Question dialog.

**Return Value:**
Double value entered into text box by user

**Example:**
```
' Get next X position from user
NextPos = Question(“Enter next X position:“)
' Go there
Code “G00 X“ & NextPos
```
QueueDepth
Function QueueDepth() As Integer

This function returns the current number of entries in the trajectory planning queue.

Arguments:
None

Return Value:
Number of trajectory planner queue entries as an Integer

Example:
' Show current queue depth
Message "Queue depth = " & QueueDepth()

See also:

Random
Function Random() As Double

This function returns a pseudo-random number between 0 and 1. This can be used in place of the CB rnd() function.

Arguments:
None

Return Value:
Next pseudo-random number in sequence, as a Double

Example:
' Get next random number
NextRand = Random()

See also:

RefCombination
Sub RefCombination(Axes As Integer)

This function allows any combination of axes to be simultaneously referenced (homed). Which axes will be referenced is determined by the Axes argument, which is a bit-mapped variable, with the bits mapped as defined below.

Arguments:
Axes is a bit-mapped value the defines which axes are to be referenced. The value of Axes can be calculated by adding the values corresponding to the individual axes to be referenced. The axis values are:

- X = 1
- Y = 2
- Z = 4
- A = 8
- B = 16
- C = 32

So, for example, to reference the X, Z and B axes, Axes = 1 + 4 + 16 = 21.

Return Value:
None

Example:
' Define some constants
RefX = 1
RefY = 2
RefZ = 4
RefZ = 8
RefB = 16
RefC = 32
' Reference Y, Z and C axes
RefCombination(RefY + RefZ + RefC)

See also:
VerifyAxis(), SingleVerify(), SingleVerifyReport()
**ResetAxisSwap**

Sub ResetAxisSwap()

This function un-does the effect of a preceding SwapAxis(). Note that this is the only way to undo a Swap, and it is illegal to perform two SwapAxis() calls without an intervening ResetAxisSwap().

**Arguments:**
None

**Return Value:**
None

**Example:**

```vbscript
' Define some constants
Xaxis = 0
Yaxis = 1
Zaxis = 2
' Swap X and Y axes
SwapAxis(Xaxis, Yaxis)
' Do something with swapped axes here...
' Undo the SwapAxis
ResetAxisSwap()
' Now Swap Y and Z axes
SwapAxis(Yaxis, Zaxis)
```

**See also:**
SwapAxis()

---

**ResetTHC**

Sub ResetTHC()

This function resets the Torch Height. This function is identical to ZeroTHC(), except if a program is running, it does nothing.

**Arguments:**
None

**Return Value:**
None

**Example:**

```vbscript
' Reset THC
ResetTHC()
```
See also:
THCOn(), THCOff()

RetractMode
Function RetractMode() As Integer

This function returns the current Peck Drilling retract mode. If a G98 was last run, it will return a 0. If a G99 was most recently run, it will return a 1.

Arguments:
None

Return Value:
Integer value indicating which peck cycle was last run. 0=G98, 1=G99.

Example:
' Run a G98
Code “G98 Z-0.5 R0.1 F10”
' Wait for it to complete
While IsMoving()
   Sleep 100
Wend
' Now show retract mode - should show 0
MsgBox(“RetractMode = “ & RetractMode())
Code “Hit CycleStart to continue…”
' Run a G99
Code “G99 Z-0.5 R0.1 F10”
' Wait for it to complete
While IsMoving()
   Sleep 100
Wend
' Now show retract mode - should show 1
MsgBox(“RetractMode = “ & RetractMode())
Code “Hit CycleStart to continue…”

See also:

roun
Function roun(Val As Double) As Double

This function rounds the Double value specified by Val to four decimal placed.
Arguments:
Val is the Double value to be rounded

Return Value:
Double value of Val rounded to four decimal places

Example:
' Round 1.23456789
Message “1.23456789 rounds to “ & roun(1.23456789)

See also:
Round()

RunFile
Sub RunFile()

This function begins execution of the currently loaded G-code program, if any. This is functionally identical to DoOEMButton(CYCLESTART)

Arguments:
None

Return Value:
None

Example:
' Load our G-code file
LoadFile(“MyGCode.tap”)  
' Run it
RunFile()

See also:
FileName(), LoadFile(), LoadRun(), IsLoading(), DoOEMButtonDown()

RunScript
Function RunScript(ByVal QFN as string) as Integer

This function can be used to run the script in the specified filename. 
The qualified file name (QFN) parameter is relative to the Mach install directory. 
The function will load the specified script file, compile it, execute it and return when the script has completed.

Prior to the addition of this call, it was common practice to put script code into a Mxxx.m1s macro, and use the Code call to execute the Mxxx macro. The execution of an Mxxx macro involves the use of the Mach GCode interpreter (as what you are really
doing is executing a GCode M word block) and can result in the programmer having to invent and handle semaphores to coordinate the asynchronous execution of the Mxxx macro.

It is recommended that the use of the Code “as a subroutine call method” be avoided and that, when possible, the RunScript call should be used instead.

**Arguments:**
- **QFN:** The Qualified File Name (without extension) of the script file to run. The extension is not included as Mach3 will look for either an .m1s or .mcc extension for the named macro.

**Return Value:**
- 0 = Script was found and invoked.
- < 0 = error condition, script was not run.

Error Return values:

<TBD>

**Example:**

```
  ' Run script example
  If RunScript(“MsgBoxScript”) < 0 then
      MsgBox “RunScript returned an error”
  Else
      MsgBox “Script ran”
  End If

  Where the file MsgBoxScript.m1s contains
  MsgBox “Message from script file”

  Will result show a dialog box with the content “Message from script file”. Followed by a dialog that says “Script Ran”
```

**See also:**

n/a

**First Mach3 version with API:**
This API was first implemented in Mach3 version 3.43.06. On 3.43.6 the API was defined as a Boolean function

**Return Value:**
- True = Script was found and invoked.
- False = Script was not found.

This API was revised in Mach3 version 3.43.19 to be an integer function.
**SaveWizard**

Sub SaveWizard()

This function can be used within a wizard to save the user-entered parameter value settings for the wizard, so the same values will be loaded the next time the wizard is invoked.

**Arguments:**
None

**Return Value:**
None

**Example:**
```
' Save current Wizard parameter values
SaveWizard()
```

See also:

**SetButtonText**

Sub SetButtonText(Text As String)

This function allows the text of an on-screen button to be changed by that buttons associated button script. This can be used to change the text on a button based on the state of a variable or mode.

**Arguments:**
Text is a String that specifies the new text to be displayed on the button.

**Return Value:**
None

**Example:**
```
' Example Spindle On/Off Toggle Button
SpindleCWLED = 11
If GetOEMLED(SpindleCWLED) Then
   ' Spindle is on, so turn it off
   DoSpinStop()
   SetButtonText(“Spindle On”)
Else
   ' Spindle is off, so turn it on
   DoSpinCW()
```
SetButtonText(“Spindle Off”)
End If

See also:

**SetCurrentTool**

Sub SetCurrentTool(ToolNum As Integer)

This function sets the current tool number to ToolNum. This is typically used in the M6Start script to make the selected tool the current tool.

*Arguments:*

ToolNum is an Integer tool number, from 1 to 255.

*Return Value:*

None

*Example:*

' Typical M6Start script
' Get selected tool
NewTool = GetSelectedTool()
' Make it the current tool
SetCurrentTool(NewTool)

See also:

GetCurrentTool(), GetSelectedTool()

**SetDRO**

Function SetDRO(DRONum As Integer, DROVal As Double)

This legacy function sets the DRO number specified by DRONum to the value specified by DROVal.

The use of SetDRO is no longer recommended practice and this function exists only to support preexisting legacy scripts. This function is deprecated, and its use is strongly discouraged.

Legacy script note: Over time, there have been two different DRO numbering schemes used with Mach; the “DRO number” series and the “OEMDRO number” series. This function uses the “DRO number” series.
The “DRO number” series was further subdivided into “User” and “OEM” ranges. Within the “OEM” range, valid DRONums were from 0 to 200, which, at one time, corresponded to OEM DRO numbers 800 to 1000.

The numerical correspondence between the numbering series is not guaranteed for future releases of Mach.

Use the SetOEMDRO and SetUserDRO functions instead of this function.

**Arguments:**
- DRONum is the Integer DRO number to read. The value has to be within the “DRO number series”.
- DROVal is the Double value to which the specified DRO will be set.

**Return Value:**
None

**Example:**
```vbscript
' Define the axes
Const XaxisMultiFunctionDRONum = 0
Const YaxisMultiFunctionDRONum = 1
Const ZaxisMultiFunctionDRONum = 2

' Write 1.2345 to Z axis DRO using SetDRO
SetDRO(ZaxisMultiFunctionDRONum, 1.2345)

' Show the user the Z Axis DRO value, using GetDRO()
MsgBox “After using SetDRO() the Z Axis DRO reads: “ & GetDRO(ZaxisMultiFunctionDRONum)
```

**See also:**
SetOEMDRO(), GetOEMDRO(), SetUserDRO(), GetUserDRO(), GetDRO()

**SetFeedRate**
Sub SetFeedRate(FeedRate As Double)

This function sets the feed rate. Note that FeedRate is specified in units per second, rather than units per minute.

**Arguments:**
- FeedRate specified in units/second, as a Double

**Return Value:**
None
Example:

‘ Set the feed rate to 123.456 inches/minute
SetFeedRate(123.456 / 60)
‘ Get the current feed rate, in inches/minute, and display it
CurrentFeedrate = FeedRate()
‘ Display it on the status line
Message “Current feed rate = “ & CurrentFeedrate

See also:
FeedRate()

SetFormula

Sub SetFormula(Formula As String, Axis As Integer)

This function sets one of the axis formulas, accessible by FunctionCfg’s->Formulas. These allow motion for axes to be defined by formulas involving the positions of other axes, using algebraic and trigonometric functions. Note that for Formulas to take effect, you must check the “Formulas enabled” checkbox in FunctionCfg’s->Formulas.

Arguments:
Formula is a String that defines the algebraic/trigonometric function to be used to calculate axis position.
Axis is an Integer value that defines which axis the Formula is to be applied to

Return Value:
None

Example:

‘ Define some constants
Xaxis = 0
Yaxis = 1
Zaxis = 2
‘ Define the axis formulas such that a programmed linear move
‘ Y causes a causes a sinusoidal move in X
‘ Z axis should move normally
SetFormula(“Z", Zaxis)
SetFormula(“Y", Yaxis)
SetFormula(“Sin(y)”, Xaxis)

See also:
**SetIJMode**
Sub SetIJMode(IJMode As Integer)

This function sets the IJ mode. IJMode = 0 sets absolute mode, while IJMode = 1 sets incremental mode.

*Arguments:*
  IJMode is the Integer mode to select. 0=Absolute, 1=Incremental

*Return Value:*
  None

*Example:*
  ' Define some constants
  IJAbsolute = 0
  IJIncremental = 1
  ' Set IJMode to incremental
  SetIJMode(IJIncremental)

*See also:*
  GetIJMode()

**SetMachZero**
Sub SetMachZero(Axis As Integer)

This function zeroes the machine position of the specified axis to the current position.

*Arguments:*
  Axis is an Integer value identifying the axis to be zeroed. 0=X, 1=Y, 2=Z, 3=Z, etc.

*Return Value:*
  None

*Example:*
  ' Define some constants
  Xaxis = 0
  Yaxis = 1
  Zaxis = 2
  ' Set machine zero for Y axis to current position
  SetMachZero(Yaxis)

*See also:*
**SetOEMDRO**
Sub SetOEMDRO(DRONum As Integer, DROVal as Double)

This function sets the OEM DRO specified by DRONum to the value specified by DROVal.

**Arguments:**
DRONum is an Integer value or expression that evaluates to one of the OEM DRO numbers.

**Return Value:**
None

**Example:**
```
' Define the axes
Const XaxisMultiFunctionOEMDRONum = 800
Const YaxisMultiFunctionOEMDRONum = 801
Const ZaxisMultiFunctionOEMDRONum = 802

' Write 1.2345 to Z axis DRO using SetOEMDRO
SetOEMDRO(ZaxisMultiFunctionDRONum, 1.2345)

' Show the user the Z Axis DRO value, using GetOEMDRO()
MsgBox "After using SetOEMDRO() the Z Axis DRO reads:"
   " & GetOEMDRO(ZaxisMultiFunctionDRONum)
```

**See also:**
GetOEMDRO(), SetUserDRO(), GetUserDRO()

**SetPage**
Sub SetPage(PageNum As Integer)

This function switches the current display page to the one specified by PageNum.

**Arguments:**
PageNum is the Integer number of the display page to switch to.

**Return Value:**
None

**Example:**
' For 1024.set, change to Diagnostics page
SetPage(5)

See also:
GetPage()

SetParam
Sub SetParam(ParamName As String, ParamVal As Double)

This function allows a number of Mach3 internal parameters (not to be confused with G-code parameters) to be set. Each Mach3 parameter is identified by name. The parameter whose name is given by ParamName is set to the value given by ParamVal. Valid parameters are:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPMOverRide</td>
<td>When this parameter is set to 1, the True Spindle Speed DRO (OEM DRO 39) is made writeable, over-riding Mach3s normal calculation of True Spindle Speed.</td>
</tr>
<tr>
<td>Xscale</td>
<td>X axis scale factor</td>
</tr>
<tr>
<td>YScale</td>
<td>Y axis scale factor</td>
</tr>
<tr>
<td>ZScale</td>
<td>Z axis scale factor</td>
</tr>
<tr>
<td>AScale</td>
<td>A axis scale factor</td>
</tr>
<tr>
<td>BScale</td>
<td>B axis scale factor</td>
</tr>
<tr>
<td>CScale</td>
<td>C axis scale factor</td>
</tr>
<tr>
<td>FeedRate</td>
<td>Feed rate</td>
</tr>
<tr>
<td>Units</td>
<td>Current units (inch/mm). 0 = mm, 1 = inch</td>
</tr>
<tr>
<td>StepsPerAxisX</td>
<td>X axis steps per unit</td>
</tr>
<tr>
<td>StepsPerAxisY</td>
<td>Y axis steps per unit</td>
</tr>
<tr>
<td>StepsPerAxisZ</td>
<td>Z axis steps per unit</td>
</tr>
<tr>
<td>StepsPerAxisA</td>
<td>A axis steps per unit</td>
</tr>
<tr>
<td>StepsPerAxisB</td>
<td>B axis steps per unit</td>
</tr>
<tr>
<td>StepsPerAxisC</td>
<td>C axis steps per unit</td>
</tr>
<tr>
<td>VelocitiesX</td>
<td>X axis maximum velocity, from motor tuning, in units/second</td>
</tr>
<tr>
<td>VelocitiesY</td>
<td>Y axis maximum velocity, from motor tuning, in units/second</td>
</tr>
<tr>
<td>VelocitiesZ</td>
<td>Z axis maximum velocity, from motor tuning, in units/second</td>
</tr>
<tr>
<td>VelocitiesA</td>
<td>A axis maximum velocity, from motor tuning, in units/second</td>
</tr>
<tr>
<td>VelocitiesB</td>
<td>B axis maximum velocity, from motor tuning, in units/second</td>
</tr>
<tr>
<td>VelocitiesC</td>
<td>C axis maximum velocity, from motor tuning, in units/second</td>
</tr>
<tr>
<td>AccelerationX</td>
<td>X axis maximum acceleration, from motor tuning</td>
</tr>
<tr>
<td>AccelerationY</td>
<td>Y axis maximum acceleration, from motor tuning</td>
</tr>
<tr>
<td>AccelerationZ</td>
<td>Z axis maximum acceleration, from motor tuning</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AccelerationA</td>
<td>A axis maximum acceleration, from motor tuning</td>
</tr>
<tr>
<td>AccelerationB</td>
<td>B axis maximum acceleration, from motor tuning</td>
</tr>
<tr>
<td>AccelerationC</td>
<td>C axis maximum acceleration, from motor tuning</td>
</tr>
<tr>
<td>SpindleSpeed</td>
<td>Should modify Spindle Speed, but does not work on all versions. Use SetSpinSpeed() instead.</td>
</tr>
<tr>
<td>ZInhibitOn</td>
<td>Z Inhibit Enable. 0=Z inhibit disabled, 1=Z inhibit enabled. When Z inhibit is enabled, the Z axis will not be allowed to move below the depth specified by the ZinhibitDepth parameter.</td>
</tr>
<tr>
<td>ZInhibitDepth</td>
<td>Z Inhibit Depth. When Z inhibit is enabled, the Z axis will not be allowed to move below the depth specified by the ZinhibitDepth parameter.</td>
</tr>
<tr>
<td>SafeZ</td>
<td>SafeZ height</td>
</tr>
<tr>
<td>Boundry</td>
<td>Toolpath Boundaries display enable. 0=&gt;disable boundaries display, 1=&gt;enable boundaries display</td>
</tr>
<tr>
<td>XRefPer</td>
<td>X axis homing speed, as % of rapid speed</td>
</tr>
<tr>
<td>YRefPer</td>
<td>Y axis homing speed, as % of rapid speed</td>
</tr>
<tr>
<td>ZRefPer</td>
<td>Z axis homing speed, as % of rapid speed</td>
</tr>
<tr>
<td>ARefPer</td>
<td>A axis homing speed, as % of rapid speed</td>
</tr>
<tr>
<td>BRefPer</td>
<td>B axis homing speed, as % of rapid speed</td>
</tr>
<tr>
<td>CRefPer</td>
<td>C axis homing speed, as % of rapid speed</td>
</tr>
<tr>
<td>ToolPathLock</td>
<td>Toolpath Lock enable. 0=&gt;Toolpath unlocked, 1=&gt;Toolpath locked. When the tool path is locked, it cannot be moved, scaled or rotated.</td>
</tr>
<tr>
<td>PrepMove</td>
<td>Preparatory Move Dialog Inhibit. 0=&gt;Allow showing of Preparatory Move dialogs, 1=&gt;Inhibit showing of Preparatory Move dialogs</td>
</tr>
<tr>
<td>AutoToolChange</td>
<td>??? – Indicates auto tool changer in use?</td>
</tr>
<tr>
<td>ADirActive</td>
<td>??? – Reverse direction of A axis?</td>
</tr>
</tbody>
</table>

**Arguments:**
ParamName is the String name of the parameter to be set. This must be one of the above names.
ParamVal is the Double value to which the specified parameter will be set.

**Return Value:**
None

**Example:**
```
' Get the new scale factor from the user
ScaleFactor = Question "Enter new scale factor:"
' Set the new scale factor for X/Y/Z
SetParam("Xscale", ScaleFactor)
SetParam("Yscale", ScaleFactor)
SetParam("Zscale", ScaleFactor)
```

**See also:**
GetParam()
SetPulley
Sub SetPulley(Pulley As Integer)

This function sets the current spindle pulley number. This allows Mach3 to properly scale the spindle speed output based on the current pulley and commanded spindle speed. Pulley ratios and allowable speed ranges are configured in Config->SpindlePulleys

Arguments:
Pulley is an Integer pulley number, from 1 to 15

Return Value:
None

Example:
' Prompt the user for new pulley setting
NewPulley = Question "Enter new pulley number:"
' Tell Mach3
SetPulley(NewPulley)

See also:

SetSafeZ
Sub SetSafeZ(SafeZ As Double)

This function sets a new value for SafeZ. Note that SafeZ must be enabled in Config->SafeZConfig

Arguments:
SafeZ is the Double value to set SafeZ to

Return Value:
None

Example:
' Change SafeZ to +2.000
SetSafeZ(2.000)

See also:
**SetScale**
Sub SetScale(Axis As Integer, Scale As Double)

This function sets the scale factor for axis Axis to the value given by Scale.

**Arguments:**
- Axis is the Integer Axis.  0=X, 1=Y, 2=Z, 3=A, etc.
- Scale is the Double Scale factor

**Return Value:**
None

**Example:**
```
' Define some constants
Xaxis = 0
Yaxis = 1
Zaxis = 2
' Get the current axis scale factors
OldXScaleFactor = GetScale(Xaxis)
OldYScaleFactor = GetScale(Yaxis)
OldZScaleFactor = GetScale(Zaxis)
' Get the new scale factor from the user
ScaleFactor = Question(“Enter new scale factor:”)
' Set new scale factors for X/Y/Z
SetScale(Xaxis, ScaleFactor)
SetScale(Yaxis, ScaleFactor)
SetScale(Zaxis, ScaleFactor)
```

**See also:**
GetScale()

---

**SetSpinSpeed**
Sub SetSpinSpeed(RPM As integer)

This function sets the spindle speed, in RPM, exactly as the “S” word in G-code does.

**Arguments:**
- RPM is the Integer spindle speed desired.

**Return Value:**
None

**Example:**
```
' Turn spindle on at 2500 RPM
SetSpinSpeed(2500)
```
DoSpinCW()

See also:
DoSpinCW(), DoSpinCCW(), DoSpinStop()

SetTicker
Sub SetTicker(TickerNum As Integer, TickerText As String)

This function loads the text given by TickerText into the ticker specified by TickerNum. A Ticker is a scrolling text label, with its default text set to “Tickernn”, where “nn” is a number between 0 and 255. A long message can be put in a Ticker, and it will scroll continuously so the entire message is visible, even if the label is shorter than the text.

Arguments:
TickerNum is the Integer number of the ticker to be written
TickerText is the String text to be written to ticker TickerNum

Return Value:
None

Example:
' Write the current file path to Ticker 25
SetTicker(25, FileName())

See also:
SetUserLabel(), Message()

SetTimer
Function SetTimer(TimerNum As Integer)

This function clears the specified timer. Mach3 provides 25 timers, numbered 0 to 24, which can be used for timing in CB scripts. To time an event, first clear the timer using SetTimer(), then use GetTimer() to read the timer. Note that this function works only with the parallel port driver, and support for this function may be removed without notice in a future release.

Arguments:
TimerNum is an Integer timer number, which must be between 0 and 24.

Return Value:
None

Example:
' Clear timer 15
SetTimer(15)
' Wait for OEM Trigger 10 to go active
While IsActive(OEMTRIG10) = False Then
    Sleep 10
Wend
' See how long it took
Message “OEMTRIG10 active after “ & GetTimer(15) & “ seconds”

See also:
GetTimer()

SetToolDesc
Function SetToolDesc(ToolNum As Integer, TDesc As String)

This function set the text description of a tool in internal Mach the tool table.

Arguments:
  ToolNum is an Integer tool number, from 0 to 254.
  TDesc is a string with the description of the tool.

Return Value:
  True: description was set for the tool
  False: An error occurred attempting to set the tool description.

Example:
  ' Show user the current tool description
  Dim TNum as Integer
  Dim TDesc as string

  TNum = 1 ' want to set description for tool # 1
  TDesc = “1/4 135 degree split point drill”

  If SetToolDescription(TNum, TDesc) then
      Message “Tool description was set”
  Else
      Message “Error setting Tool description”
  End if

See also:
GetToolDesc(), GetToolParam(), SetToolParam(),

First Mach3 version with API:
This API was first implemented in Mach3 version 3.42.30.

**SetToolParam**

Sub SetToolParam(ToolNum As Integer, ParamNum As Integer, ParamVal As Double)

This function allows any tool parameter for any tool to be set. ToolNum is the number of the tool whose parameters are being set, and can be from 1 to 255. ParamNum is a parameter number, defined as follows:

For Mach3Mill:
1 = Diameter  
2 = Z Offset  
3 = X Wear  
4 = Z Wear

For Mach3Turn:
1 = Tip Type  
2 = Tool Radius  
3 = X Offset  
4 = Z Offset  
5 = X Wear  
6 = Z Wear  
7 = Turret Angle

**Arguments:**
ToolNum is an Integer tool number, and must be between 1 and 255. ParamNum is an Integer parameter number, defined as described above.

**Return Value:**
None

**Example:**

```vba
' Define some constants
DiameterParam = 1
ZoffsetParam = 2
XwearParam = 3
ZwearParam = 4
' Define tool #23
SetToolParam(23, DiameterParam, 0.2500)  ' Set tool diameter = 1/4"
SetToolParam(23, ZoffsetParam, 1.2500)  ' Set length offset = 1.25"
SetToolParam(23, XwearParam, 0.0005)  ' Set X wear = 0.0005"
```
SetToolParam(23, ZwearParam, 0.0013)   ' Set Z wear = 0.0013"

See also:
GetToolParam(), GetToolDesc()

SetToolX
Function SetToolX(Pos As Double)

This function sets the X axis DRO to the value given by Pos, then displays the message “Work Offset Shifted” on the status line. Note that with the exception of this message, SetToolX() is functionally identical to using SetOEMDRO to modify the X axis DRO value directly.

Arguments:
Pos is the Double position to which the X axis DRO will be set.

Return Value:
None

Example:
' Prompt the user to zero the X axis
Message “Move X axis to zero position.  Press OK when ready…”
' Zero the X axis DRO
SetToolX(0.0000)

See also:
SetToolZ()

SetToolZ
Function SetToolZ(Pos As Double)

This function sets the Z axis DRO to the value given by Pos, then displays the message “Work Offset Shifted” on the status line. Note that with the exception of this message, SetToolZ() is functionally identical to using SetOEMDRO to modify the Z axis DRO value directly.

Arguments:
Pos is the Double position to which the Z axis DRO will be set.

Return Value:
None

Example:
' Prompt the user to zero the Z axis
Message “Move Z axis to zero position. Press OK when ready...”
' Zero the Z axis DRO
SetToolZ(0.0000)

See also:
SetToolX()

**SetTriggerMacro**

Sub SetTriggerMacro(MacroNum As Integer)

This function allows an M-macro to be associated with OEM code 301. This can be used to cause one of the OEM Trigger signals to automatically run a macro when asserted. To do this, the OEM Trigger signal must first be configured in Config->Ports&Pins->InputSignals. Then the OEM Trigger must be associated with OEM code 301 in Config->SystemHotKeys. Finally, SetTriggerMacro must be used to define which M-macro will be executed when OEM button code 301 is executed.

**Arguments:**
MacroNum is the number of the M-macro to be run when OEM button code 301 is executed.

**Return Value:**
None

**Example:**
' Assume OEMTRIGGER5 is assigned to OEM code 301
' Assign the M1025 macro to OEM Code 301
SetTriggerMacro(1025)
' Now, when OEMTRIGGER5 is driven to its active level, M1025.m1s will be executed

See also:

**SetUserDRO**

Sub SetUserDRO(DRONum As Integer, DROVal As Double)

This function sets the value of User DRO DRONum to DROVal.

**Arguments:**
DRONum is the Integer User DRO number to be set. Valid User DRO numbers range from 1000-2254.

DROVal is the Double value to which the User DRO will be set

Return Value:
None

Example:
`Define some constants
MyWidgetDRO = 1125
'Set MyWidgetDRO to 1.234
SetUserDRO(MyWidgetDRO, 1.234)
...
'Get current value of MyWidgetDRO
MyDROVal = GetUserDRO(MyWidgetDRO)

See also:
GetUserDRO(), SetOEMDRO(), GetOEMDRO()

SetUserLabel
Sub SetUserLabel(LabelNum As Integer, LabelText As String)

This function allows the user to change an on-screen “User” labels from CB, rather than by using a screen set editor. “User” labels are those that are created in the screen designer with the default text containing the String “UserLabel” followed by one or more digits.

Arguments:
LabelNum is the numeric portion of the user label default text. LabelNum must be between 0 and 255.
LabelText is the text to be placed into the label.

Return Value:
None

Example:
`Change the text in UserLabel25
SetUserLabel(25, “This is Label 25”)
...
'Retrieve the text from UserLabel25
LabelText = GetUserLabel(25)

See also:
**SetUserLED**

Sub SetUserLED(LEDNum As Integer, State As Integer)

This function allows the state of a User LED to be set or cleared. User LEDs are numbered from

**Arguments:**

- LEDNum is the User LED to be set, and must be in the range of 1000 to 2254
- State is the new state of the User LED. 0 indicates the LED is off (unlit), 1 indicates the LED is on (lit).

**Return Value:**

None

**Example:**

```
' Define some constants
FluxCapacitorControl = OUTPUT1 ' Output that controls the flux capacitor
FluxCapacitorLED = 1234 ' LED that indicates flux capacitor is active
' Turn on the Flux capacitor
ActivateSignal(FluxCapacitorControl)
' Turn on the Flux Capacitor LED for the operator
SetUserLED(FluxCapacitorLED, 1)
...
' Is the Flux Capacitor on?
FluxCapacitorOn = GetUserLED(FluxCapacitorLED)
```

**See also:**

GetUserLED(), SetOEMLED(), GetOEMLED()

---

**SetVar**

Sub SetVar(VarNum As Integer, Val As Double)

This function sets the Mach variable specified by VarNum to the value given by Val. Mach variables are accessible both to CB scripts, using the SetVar() and GetVar() functions, as well as G-code programs, using the #nnnn syntax.

**Arguments:**

- VarNum is Integer the number of the Mach variable to be set.
- Val is the Double value to which the variable will be set

**Return Value:**

None
Example:

```
  ' Set a variable 1234 to our target position of 2.3456
  SetVar(1234, 2.3456)
  ' Now move X to our target position
  Code “GO X #1234”
```

See also:

GetVar()

---

**SingleVerify**

Sub SingleVerify(Axis As Integer)

This function performs a position verify on a single axis by homing that axis, and, once homed, zeroing the work offset, then returning to the initial offset. This is commonly used at the beginning of a program to ensure the fixture offset is set properly, or after a crash to restore the correct work offset.

Arguments:

Axis in an Integer value that specifies the first axis to verify. 0=X, 1=Y, 2=Z, 3=A, etc.

Return Value:

None

Example:

```
  Xaxis = 0
  Yaxis = 1
  Zaxis = 2
  ' Verify Z axis work offset is properly set
  SingleVerify(Zaxis)
```

See also:

SingleVerifyReport(), VerifyAxis(), RefCombination()

---

**SingleVerifyReport**

Function SingleVerifyReport(Axis As Integer)

This function performs a position verify on a single axis by homing that axis, and, once homed, zeroing the work offset, then returning to the initial offset. When complete, a message is displayed showing how far each axis was offset from its correct home position. This function is identical to the SingleVerify function except for this message
display. This is commonly used at the beginning of a program to ensure the fixture offset is set properly, or after a crash to restore the correct work offset.

Arguments:
Axis in an Integer value that specifies the first axis to verify. 0=X, 1=Y, 2=Z, 3=A, etc.

Return Value:
None

Example:
Xaxis = 0
Yaxis = 1
Zaxis = 2
' Verify Z axis work offset is properly set, and report error, if any
SingleVerifyReport(Zaxis)

See also:
SingleVerifyReport(), VerifyAxis(), RefCombination()

Sleep
Sub Sleep(Time As Integer)

This function causes the CB program to pause for the specified period of time, in mSec. During this time, other processes can have access to the CPU. A Sleep() call should always be inserted in any loop which might remain active for any period of time, for example the While loop used to wait for a move command to complete.

Arguments:
Time, an Integer value expressing the sleep time in milliseconds.

Return Value:
None

Example:
' Move to zero position
Code "G0 X0.000 Y0.000"
' Wait for move to complete
While IsMoving()
  ' Let other processes have CPU while we wait
  Sleep 100
Wend

See also:
**Speak**
Sub Speak(TextToSpeak As String)

This function uses speech synthesis to “speak” the String argument. Note that your PC must have a working audio system, and speech must first be enabled by checking the “Allow Speech” checkbox in Config->GeneralConfig->GeneralConfiguration.

**Arguments:**
TextToSpeak is a String to be spoken

**Return Value:**
None

**Example:**
```
' Tell the user to load the next work piece
Speak(“Please load next work piece. Hit CycleStart
When Ready”)
Code “M00”
```

**See also:**

**StartPeriodicScript**
Function StartPeriodicScript(ByVal ScriptQFN as String, ByVal UpdatePeriod as Double) as Integer

This function causes Mach to start a script that will be invoked with the specified periodicity.

**Arguments:**
ScriptQFN: the string of the Qualified File Name (QFN) for the script to be run. The qualified path name is relative to the Mach install directory. The QFN does not include the script extension. Mach will look first for ScriptQFN.mcc and if not found, then ScriptQFN.m1s.

If the QFN is the name of a script already started by a previous call to StartPeriodicScript(), an error condition is returned. Mach does not support multiple periodic instances of a single script.

UpdatePeriod: The length of the time period between runs of the script. The time units are seconds. The minimum value is 5ms, any value less than 5ms will be ignored and the minimum value of 5ms will be used for the UpdatePeriod.
It is recommended that periodic scripts be run with the longest UpdatePeriod appropriate for the script’s task. This will help minimize the load on the machine from multiple periodic scripts.

**Return Value:**
- $0$ = requested function was performed successfully (script is now running).
- $< 0$ = error, requested function was not successful (script is not running).

Error Return Values:
- `<tbd>`

**Example:**
```vba
' Initialize a script that runs the way oiler every 30 minutes

If StartPeriodicScript(“OilerScript”, 30*60*1000) then
    MsgBox “Oiler periodic script is running”
Else
    MsgBox “Oiler script was not started”
End If
```

**See also:**
- IsPeriodicScriptRunning, StopPeriodicScript

**First Mach3 version with API:**
This API was first implemented in Mach3 version 3.43.06.
It was defined as a Boolean function with

**Return Value:**
- True = requested function was performed successfully (script is now running).
- False = error, requested function was not successful (script is not running).

This API was revised in Mach3 version 3.43.19 to be an Integer function.

**StartTHC**
Sub StartTHC()

This function turns on torch height control. It is functionally identical to THCOOn().

**Arguments:**
- None

**Return Value:**
None

**Example:**
```
StartTHC() ' Turn on torch height control
  ' Do some cutting here
EndTHC() ' Turn off torch height control
```

**See also:**
THCOn(), THCOff(), EndTHC(), ZeroTHC(), ResetTHC()

---

**StopPeriodicScript**

Function StopPeriodicScript(ByVal ScriptQFN as String) as Integer

This function is used to stop a previously started periodic script.

A periodic script only stops at the end of one of the script’s UpdatePeriod time quanta (as set by StartPeriodicScript) cycles. For example if a script is running with a 5 minute UpdatePeriod, and the stop request is issued 2 minutes into the cycle, the script will not stop until the current 5 minute cycle expires.

**Arguments:**

- **ScriptQFN**: the string of the Qualified File Name (QFN) for the script to be stopped. The QFN is relative to the Mach install directory.
  - The QFN passed must be identical to the QFN used to start the periodic script.
  - Attempts to stop a script that has not been started by StartPeriodicScript are ignored (this is not an error condition as the script is “stopped” upon return from the call).

**Return Value:**

- 0 = requested function was performed successfully (script is now running).
- < 0 = error, requested function was not successful (script is not running).

**Error Values:**

<tbdl>

**Example:**
```
' Stop a the running way oiler script

If StopPeriodicScript(“OilerScript“) then
    MsgBox “Oiler periodic script has been stopped”
Else
    MsgBox “Error, Oiler script was not stopped, check the QFN passed.”
End If
```
**See also:**
IsPeriodicScriptRunning, StartPeriodicScript,

**First Mach3 version with API:**
This API was first implemented in Mach3 version 3.43.06.
It was defined as a Boolean function with

**Return Value:**
True = requested function was performed successfully (script is now running).
False = error, requested function was not successful (script is not running).

This API was revised in Mach3 version 3.43.19 to be an Integer function.

**StraightFeed**
Sub StraightFeed(X As Double, Y As Double, Z As Double, A As Double, B As Double, C As Double)

This function performs a feed rate move to the specified position. Note that all axis positions must be specified. This is exactly equivalent to Code “G1 Xn.nnn Yn.nnn Zn.nnn An.nnn Bn.nnn Cn.nnn”.

**Arguments:**
X is a Double specifying the target position for the X axis
Y is a Double specifying the target position for the Y axis
Z is a Double specifying the target position for the Z axis
A is a Double specifying the target position for the A axis
B is a Double specifying the target position for the B axis
C is a Double specifying the target position for the C axis

**Return Value:**
None

**Example:**
```
' Send all axes to zero position at current feed rate
StraightTraverse(0.000, 0.000, 0.000, 0.000, 0.000, 0.000)
```

**See also:**
StraightTraverse(), Code()

**StraightTraverse**
Sub StraightTraverse(X As Double, Y As Double, Z As Double, A As Double, B As Double, C As Double)
This function performs a rapid move to the specified position. Note that all axis positions must be specified. This is exactly equivalent to Code “G0 Xn.nnn Yn.nnn Zn.nnn An.nnn Bn.nnn Cn.nnn”.

**Arguments:**
- X is a Double specifying the target position for the X axis
- Y is a Double specifying the target position for the Y axis
- Z is a Double specifying the target position for the Z axis
- A is a Double specifying the target position for the A axis
- B is a Double specifying the target position for the B axis
- C is a Double specifying the target position for the C axis

**Return Value:**
None

**Example:**
```
' Send all axes to zero position
StraightTraverse(0.000, 0.000, 0.000, 0.000, 0.000, 0.000)
```

**See also:**
StraightFeed(), Code()

---

**SwapAxis**

Sub SwapAxis(FirstAxis As Integer, SecondAxis As Integer)

This function swaps the STEP and DIR pins for the two specified axes. This has precisely the same effect as changing the pin settings in Config->Ports&Pins. Note that no other axis parameters or settings are changed. If the two specified axes have different acceleration and velocity settings, unreliable operation will likely result. Note also that if you exit Mach3 while the swap is in effect, the swapped pins will be written to the XML configuration file, and the swap will still be in effect the next time you start Mach3. It is illegal to perform two consecutive swaps, without first executing a ResetAxisSwap(). Doing so will likely result in incorrect operation.

**Arguments:**
- FirstAxis in an Integer value that specifies the first axis to swap. 0=X, 1=Y, 2=Z, etc.
- SecondAxis in an Integer value that specifies the second axis to swap. 0=X, 1=Y, 2=Z, etc.

**Return Value:**
None
Example:
Xaxis = 0
Yaxis = 1
Code “G0 X0 Y0”
' Cut a 45 degree diagonal line from lower right to upper left
Code “G1 X-1 Y1”
Code “G0 X0 Y0”
' Now swap X and Y axes
SwapAxis(Xaxis, Yaxis)
' Cut a 45 degree diagonal line from upper left to lower right
Code “G1 X-1 Y1”
Code “G0 X0 Y0”

See also:
ResetAxisSwap()

SystemWaitFor
Sub SystemWaitFor(Signal As Integer)

This function it is used to tell Mach that a macro script is ending, and that any further GCode execution should wait for the Signal to go active. If used, this function should always be the last line in a macro and it should never be used within a loop.

Arguments:
SignalID must be one of the pre-defined Mach3 CB output signal constants (see CB Constants), or other value or expression that evaluates to one of those values.

Return Value:
None

Example:

See also:

THCOff
Sub THCOff()

This function turns off the torch height control.
**Arguments:**
None

**Return Value:**
None

**Example:**
```
' Turn off THC
THCOff()
```

**See also:**
THCOOn(), ZeroTHC()

---

**THCOOn**

Sub THCOOn()

This function turns on the torch height control.

**Arguments:**
None

**Return Value:**
None

**Example:**
```
' Turn on THC
THCOOn()
```

**See also:**
THCOff(), ZeroTHC()

---

**ToggleScreens**

Sub ToggleScreens()

This function toggles the active screen set between the “complex” and “simple” (.set and .sset) screen sets of the same base name. For example, if the current screen set is the default “1024.set”, this function will load the “1024.sset” screen set, and vice-versa. This function can be used to toggle between any two screen sets by simply giving both the same base filename, and giving one the .set extension, and the other the .sset extension.

**Arguments:**
None
**ToolLengthOffset**

Function ToolLengthOffset() As Double

This function returns the tool length offset for the current tool.

**Arguments:**

None

**Return Value:**

Tool length offset for current tool, as a Double

**Example:**

```
' Get the current tool length offset
LengthOffset = ToolLengthOffset
' Tell the user
Message “Tool length offset = “ & LengthOffset
```

**See also:**

GetCurrentTool(), SetCurrentTool(), GetSelectedTool(), GetToolParam(), SetToolParam(), SetToolX(), SetToolZ(), GetToolDesc()

---

**VerifyAxis**

Sub VerifyAxis(Silent As Boolean)

This function verifies the positions of all axes by performing a home operation on all axes simultaneously, then moving all axes to position 0.0000 of the current fixture. If Safe_Z is enabled, a move to Safe_Z will be performed first. If the Silent argument is True, then upon completion, a message is displayed on the status line showing the offset errors of all axes from their home position. This function I used to ensure that no
position loss has occurred, and/or to correct any position loss that may have occurred since the previous home or verify operation.

**Arguments:**
Silent is a Boolean value that, if true, causes a message to be displayed on completion of the verify indicating the offsets, if any, of each axis from its correct position.

**Return Value:**
None. All axes machine zeroes are reset to their correct position, and all axes are moved to position 0.0000 of the current fixture.

**Example:**
```
' Make sure all axes are in correct position
VerifyAxis()
' Load the G-code file
LoadFile("MyGCodeFile.nc")
' Wait for Load to Complete
While IsLoading()
    Sleep 100
Wend
' Run the File
DoOEMButton(CycleStartButton)
' Wait for it to complete
While (GetOEMLED(StartLED))
    Sleep 100
Wend
```

**See also:**
SingleVerify(), SingleVerifyReport(), RefCombination()

---

**ZeroTHC**

Sub ZeroTHC()

This function zeroes the torch height correction factor.

**Arguments:**
None

**Return Value:**
None

**Example:**
```
' Zero THC correction
ZeroTHC()
```
See also:
THCON(), THCOff()
Legacy Functions Grouped By Function

**Digitizing**
- CloseDigFile – Close digitization points file
- OpenDigFile – Open digitization points file

**G-Code & G-code Files**
- Code – Execute a line of G-code
- FileName – Get current G-code filename
- GetVar – Get a G-code Var value
- IsLoading – Get current G-code file loading status
- LoadFile – Load a G-code file
- LoadRun – Load and run a G-code file
- MaxX – Get maximum X extent for current G-code file
- MaxY – Get maximum Y extent for current G-code file
- MinX – Get minimum X extent for current G-code file
- MinY – Get minimum Y extent for current G-code file
- Param1 – Get M-macro P parameter value
- Param2 – Get M-macro Q parameter value
- Param3 – Get M-macro Q parameter value
- RunFile – Run currently loaded G-code file
- SetVar – Set a G-code Var value

**Lathe-only Functions**
- GetTurretAng – Get current tool turret angle
- IsDiameter – Get current diameter/radius mode status

**Mach3 Configuration & Status**
- GetIJMode – Get current IJ mode
- GetMainFolder – Get Mach3 main folder path
- GetParam – Get a Mach3 named parameter value
- GetScale – Get current scale factor for an axis
- GetTimer – Get a timer value
HelpAbout – Get current CB version
QueueDepth – Get current Mach3 trajectory queue depth
ResetAxisSwap – Un-swap swapped axes
RetractMode – Get current peck cycle retract mode
SetFormula – Set a new axis formula
SetIJMode – Set IJ mode
SetParam – Set a Mach3 names parameter
SetScale – Set a new axis scale factor
SetTimer – Reset a timer
SwapAxis – Swap axis outputs

Referencing, Verifying & Zeroing Axes
GetABSPosition – Get machine position for an axis
RefCombination – Reference any combination of axes
SetMachZero – Set machine zero to current position
SingleVerify – Verify position of a single axis
SingleVerifyReport – Verify position of a single axis, and report if error
VerifyAxis – Verify position of a single axis

SafeZ
GetSafeZ – Get current SafeZ height
GotoSafeZ – Go to current SafeZ height
IsSafeZ – Find out if machine is currently at or above SafeZ height
SetSafeZ – Set SafeZ height

Wizards& Plugins
LoadWizard – Load a wizard by name
NotifyPlugins – Send a notification to all plugins
SaveWizard – Save current wizard settings

Machine Status & Control
IsEstop – Find out if Mach3 is currently in E-Stop
CoupleSlave – Couple slave axis to its master for referencing
Motion Control
- FeedRate – Get current feed rate
- JogOff – Disable jogging for a single axis
- JogOn – Jog a single axis in a specified direction
- SetFeedRate – Set a new feed rate
- StraightFeed – Perform a feed rate move
- StraightTraverse – Perform a rapid move
- IsMoving – Find out if machine is currently moving
- IsStopped – Find out if machine is currently stopped

Spindle Control
- DoSpinCCW – Turn spindle on CCW
- DoSpinCW – Turn spindle on CW
- DoSpinStop – Turn spindle off
- GetRPM – Get current spindle RPM
- SetPulley – Set a new spindle pulley
- SetSpinSpeed – Set a new spindle speed

Tool Parameters and Tool Changes
- GetCurrentTool – Get currently loaded tool
- GetSelectedTool – Get newly selected tool
- GetToolChangeStart – Get axis positions at start of tool change
- GetToolDesc – Get the tool descriptor for the specified tool number
- GetToolParam – Get value of a named tool parameter for the specified tool number
- SetCurrentTool – Set the current tool to the specified value
- SetToolParam – Set value of a named tool parameter for the specified tool number
- SetToolX – Set the X axis offset
- SetToolZ – Set the Z axis offset
- ToolLengthOffset – Set the Z offset for the current tool

Torch Height Control
- EndTHC – Turn off THC
ResetTHC – Reset THC
StartTHC – Start THC
THCOff – Turn off THC
THCOn – Turn on THC
ZeroTHC – Zero THC

**Screen sets**

DoMenu – Invoke a menu function
DoOEMButton – Invoke an OEM Button function
GetDRO – Get current value for specified DRO
GetDROString – Get current value for specified DRO as a String
GetLED – Get current state of specified OEM LED
GetOEMDRO – Get current value for specified OEM DRO
GetOEMLED – Get current state specified OEM LED
GetPage – Get current screen set page number
GetUserDRO – Get current value of specified User DRO
GetUserLabel – Get current value of specified User Label
GetUserLED – Get current state of specified User LED
LoadStandardLayout – Load default screen set
Message – Display a message on the Error SmartLabel
SetButtonText – Change the label text on the currently active button
SetDRO – Set the value of the specified OEM DRO
SetOEMDRO – Set the value of the specified OEM DRO
SetPage – Select a new screen set page by number
SetTicker – Set the value of a Ticker SmartLabel
SetUserDRO – Set the value of the specified User DRO
SetUserLabel – Set the value of the specified UserLabel SmartLabel
SetUserLED – Set the state of the specified User LED
ToggleScreens – Toggle between “complex”/.set and “simple”/.sset screen sets

**User Dialogs**

AskTextQuestion – Ask the user a question and get a String response
GetACoor – Get a coordinate given by last GetCoord() call
GetCoord – Get axis coordinates from user
GetXCoor – Get X coordinate given by last GetCoord() call
GetYCoor – Get Y coordinate given by last GetCoord() call
GetZCoor – Get Z coordinate given by last GetCoord() call
MachMsg – Display a dialog with one or more buttons on it
Question – As the user a question, and get a numeric response

**Signals and Port I/O**
ActivateSignal – Activate/Assert an output signal
DeactivateSignal – Deactivate/Deassert an output signal
GetPortByte – Read a byte from an I/O port
IsActive – Find out if the specified named input signal is in its active state
IsOutputActive – Find out if the specified named output signal is in its active state
IsSuchSignal – Find out if the specified named signal is defined
PutPortByte – Write a byte to an I/O port
SetTriggerMacro – Associate an M-macro with OEM code 301
SystemWaitFor – Pause script execution until specified named input signal is asserted

**Teach Files**
AppendTeachFile – Open a Teach file for append
CloseTeachFile – Close currently open Teach file
LoadTeachFile – Load current Teach file for execution
OpenTeachFile – Open a new Teach file for writing

**Miscellaneous**
nFmt – Round a Double value to a specified number of decimal places
PlayWave – Play a .WAV file
Random – Get a pseudo-random number
Roun – Round a Double value to four decimal places
Sleep – Pause script execution for specified length of time
Speak – Use voice synthesis to “speak” a String
Modbus Functions Grouped Alphabetically

GetInBit
Function

This function

Arguments:
F

Return Value:
A

Example:
A

See also:

GetInput
Function

This function

Arguments:
F

Return Value:
A

Example:
A

See also:

ResetOutBit
Function

This function

Arguments:
F

Return Value:
A
**SetHomannString**

Sub SetHomannString(Col As Integer, Row As Integer, Text As String)

This function writes the Text string to the LCD display on ModIO slave. Row and Col specify the row and column position of the first character of the string. This function is exactly equivalent to the SetModIOString function with a SlaveID of 1.

**Arguments:**
- Slave is the ModBus Slave ID of the device whose LCD the string is written to
- Col is the 0-based column to which the first character of the Text will be written. Col must be an even number (multiple of 2).
- Row is the 0-based row this which the Text will be written
- Text is the String to be written to the LCD

**Return Value:**
None

**Example:**
Write “Hello, world!” to the second line of the 2x16 LCD on Modbus Slave 2
SetModIOString(2, 0, 1, “Hello, world!”)

**See also:**
FillFromCoil(), FillFromHolding(), FillFromInput(), FillFromStatus(),
GetModWord(), ModGetInputWord(), SetModIOString(), SetModOutput(),
SetModPlugString(), ResetOutBit(), GetInBit(), GetInput(), SetOutBit(),
SetOutput(), WaitForPoll(), SetModIOString()

**SetModIOString**

Sub SetModIOString(SlaveID As Integer, Col As Integer, Row As Integer, Text As String)

This function writes the Text string to the ModIO devices LCD display. Row and Col specify the row and column position of the first character of the string.

**Arguments:**
- Slave is the ModBus Slave ID of the device whose LCD the string is written to
- Col is the 0-based column to which the first character of the Text will be written. Col must be an even number (multiple of 2).
- Row is the 0-based row this which the Text will be written

**See also:**
FillFromCoil(), FillFromHolding(), FillFromInput(), FillFromStatus(),
GetModWord(), ModGetInputWord(), SetModIOString(), SetModOutput(),
SetModPlugString(), ResetOutBit(), GetInBit(), GetInput(), SetOutBit(),
SetOutput(), WaitForPoll(), SetModIOString()
Text is the String to be written to the LCD

Return Value:
None

Example:
' Write “Hello, world!” to the second line of the 2x16 LCD on Modbus Slave 2
SetModIOString(2, 0, 1, “Hello, world!”)

See also:
FillFromCoil(), FillFromHolding(), FillFromInput(), FillFromStatus(),
GetModWord(), ModGetInputWord(), SetModIOString(), SetModOutput(),
SetModPlugString(), ResetOutBit(), GetInBit(), GetInput(), SetOutBit(),
SetOutput(), WaitForPoll(), SetHomannString()

SetModOutput
Function

This function

Arguments:
F

Return Value:
A

Example:
A

See also:

SetOutBit
Function

This function

Arguments:
F

Return Value:
A

Example:
A

See also:
WaitForPoll – Unreliable….  
Function

This function

Arguments:
F

Return Value:
A

Example:
A

See also:
Serial Output Functions Grouped Alphabetically

There are a number of serial functions in the v3 code-base, some previously documented, some not. However, testing indicates the following is the only function that actually works. Serial input is not supported in Mach3 v3.

SendSerial
Sub SendSerial(Data As String)

This function send the String Data to the serial port specified in the Config->GeneralConfig serial port configuration. This provides transmit-only capability, at any supported BAUD rate.

Arguments:
String message to be sent to configured serial device

Return Value:
None

Example:
' Send “Hello, world!” to serial device
SendSerial(“Hello, world!” & chr(10) & char(13))

See also:
Script Pre-processing Functionality

Starting with Mach3 v3 version 3.43.06, Mach provides some pre-processing capability.

#Expand

Text from external files can be included and expanded inline using the #Expand facility.

The syntax is:
<white space> --- <expand keyword> --- <white space> --
-------- <angle bracket path spec> ---------
|                                                    |
--- <Quoted path spec> -------

Examples:

#expand <path-spec>
#expand "path-spec"

<white space> = any number of spaces or tabs
<expand keyword> = #expand - the word “Expand” can be any combination of upper or lower case letters.

<angle bracket path spec> = <path-spec>
<Quoted path spec> = “path-spec”

The path-spec is a filename optionally preceded by a directory specification. The filename must name an existing file. path-spec is a relative path and filename, hence it does not start with a leading “\”.

Both syntax forms cause replacement of the #expand directive by the entire contents of the specified include file.

The difference between the two forms is where the preprocessor searches for expand files and the inclusion (or not) of the .m1s extension in the path spec.

<table>
<thead>
<tr>
<th>Syntax Form</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angle-bracket form</td>
<td>This form instructs the preprocessor to search for the expand file in a location dependant on the current runtime environment.</td>
</tr>
<tr>
<td></td>
<td>The rules are:</td>
</tr>
</tbody>
</table>
1) if a wizard is NOT loaded (i.e. a regular screen set is loaded, the preprocessor will look for the path-spec in

```
<Mach install dir>\ScreenSetMacros\<ActiveScreenSetName.set>\n```

2) if a wizard is loaded, the preprocessor will look for the path-spec in

```
<Mach install dir>\AddOns\<LoadedWizardName>\n```

For the Angle-bracket form, the extension is not included in the path-spec, and the pre-processor will only look for a corresponding .m1s file.

This form is recommended for normal Mach screen set and wizard scripts.

<table>
<thead>
<tr>
<th>Quoted form</th>
<th>This form instructs the preprocessor that the path-spec is a partially qualified File name and that the preprocessor should simply use the entire QFN as found in the #expand directive line.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The QFN is assumed to be relative to the <code>&lt;Mach install dir&gt;</code>. As this is a QFN, it must also include the extension of the file to expand. (the extension is not restricted to .m1s, the preprocessor simply opens the specified FQFN).</td>
</tr>
<tr>
<td></td>
<td>This form is recommended for user scripts which are not part of a screen set or wizard.</td>
</tr>
</tbody>
</table>

The primary intent of this facility is to enable script authors to keep script sources in one file and be able to reuse the source in multiple scripts.

For example many people prefer to define names for Mach’s magic DRO and LED numbers.
This is a trivial example which uses this idea:

Macro script:

```vbnet
Option Explicit

#expand <MachConstants>

MsgBox "Mach's X DRO is # " & MachXDRONum
```
Exit Sub   ' return to Mach

Where MachConstants.m1s contains:
    Const MachXDroNum = 800

The Mach #expand processing is recursive, so that you can include a file which in turns includes a file etc.

The mach3 script editor has been enhanced to deal with script expansions.
**Screen Set Initialization and Clean up**

As of Mach version 3.32.41 Mach now will run a macro on screen set load and on screen set unload.

The macros must be in

<mach3installdir>\ScreenSetMacros\<active screen set name with extension>\n
For a screen set called MachStdMill, this would be:

C:\Mach3\ScreenSetMacros\MachStdMill.set\n
The macro names Mach looks for within the dir are:

- ScreenSetLoad
- ScreenSetUnload

ScreenSetLoad is run by Mach before the screen set is loaded. ScreenSetUnload is run by Mach when the screen set is unloaded.

Either .m1s (interpreted) or .mcc (compiled) macros may be used.

If the Screen Set directory or macros are not present, no error is thrown by Mach. This gives backwards compatibility with prior Mach versions and screen sets.

ScreenSetLoad has a couple of special characteristics:

1) If the LoadStandardLayout API is called by ScreenSetLoad, LoadStandardLayout will not invoke another instance of ScreenSetLoad (preventing a loop scenario).
2) ScreenSetLoad is run in a separate thread from the main Mach3 thread.
Brain Auto Initialization

While this is not strictly an Mach script language API, it is mentioned here as a compliment to the Screen Set initialization and clean up facilities.

Prior to Mach version 3.43.00, Mach looked in <Mach install dir>\Brains for installed brains. If a brain was enabled, it was loaded when Mach initialized. This still happens with Mach version 3.43.00+.

As of Mach version 3.43.0, Mach can auto load Brains each time Mach is started. Mach now also looks in <Mach install dir>\Brains\AutoLoad\ and any brains found there are automatically loaded, enabled, and run as part of Mach in initialization.

This provides a way for a screen set or wizard add-on to install a brain on disk and have it loaded when Mach initializes, without the need for the user to manually use the Mach menu to enable and load the brain.
OEM Series Button, DRO and LED numbers

As Mach3 developed, functions were added. Unfortunately documentation of the corresponding function numbers did not stay current.

The following tables were complied by several volunteers and represent the collectively known published OEM numbers as of the revision date of this manual. The list is not perfect and there may be errors contained within it.

**OEM Button numbers**

<table>
<thead>
<tr>
<th>Type</th>
<th>New Function Description</th>
<th>Mach3 V3 #</th>
</tr>
</thead>
<tbody>
<tr>
<td>OEM Button</td>
<td>Screen 1</td>
<td>1</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Screen 2</td>
<td>2</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Screen 3</td>
<td>3</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Screen 4</td>
<td>4</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Screen 5</td>
<td>5</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Screen 6</td>
<td>6</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Screen 7</td>
<td>7</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Screen 8...98</td>
<td>8...98</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Screen 99</td>
<td>99</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Increment</td>
<td>100</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Decrement</td>
<td>101</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Reset Interp</td>
<td>102</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Enable Off/On Toggle</td>
<td>103</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Safe Z Height Go To</td>
<td>104</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Home/Reference All Z Then X,Y,A,B,C Set Coords</td>
<td>105</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Units Inch/MM</td>
<td>106</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Multi-function DROs Machine/Work Toggle 1</td>
<td>107</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Feed Rate Override Increment (percentage)</td>
<td>108</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Feed Rate Override Decrement (percentage)</td>
<td>109</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Spindle CW Off/On Toggle</td>
<td>110</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Slow Increment</td>
<td>111</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Slow Decrement</td>
<td>112</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Coolant Flood Off/On Toggle</td>
<td>113</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Coolant Mist Off/On Toggle</td>
<td>114</td>
</tr>
<tr>
<td>OEM Button</td>
<td>G-code Edit</td>
<td>115</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Rotational axis Diameter A Zero A</td>
<td>116</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>Rotational axis Diameter B Zero B</td>
<td>117</td>
<td></td>
</tr>
<tr>
<td>Rotational axis Diameter C Zero C</td>
<td>118</td>
<td></td>
</tr>
<tr>
<td>Soft limits On/Off Toggle</td>
<td>119</td>
<td></td>
</tr>
<tr>
<td>Tool Z Zero</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>Tool Table Open</td>
<td>121</td>
<td></td>
</tr>
<tr>
<td>Work Table Open</td>
<td>122</td>
<td></td>
</tr>
<tr>
<td>Torch Height Correction Enable Off/On Toggle</td>
<td>123</td>
<td></td>
</tr>
<tr>
<td>Torch Calibration Zero</td>
<td>124</td>
<td></td>
</tr>
<tr>
<td>Encoder Load From X</td>
<td>125</td>
<td></td>
</tr>
<tr>
<td>Encoder Load To X</td>
<td>126</td>
<td></td>
</tr>
<tr>
<td>Encoder Load From Y</td>
<td>127</td>
<td></td>
</tr>
<tr>
<td>Encoder Load To Y</td>
<td>128</td>
<td></td>
</tr>
<tr>
<td>Encoder Load From Z</td>
<td>129</td>
<td></td>
</tr>
<tr>
<td>Encoder Load To Z</td>
<td>130</td>
<td></td>
</tr>
<tr>
<td>Mill/Turn</td>
<td>131</td>
<td></td>
</tr>
<tr>
<td>Tool Path Off/On Toggle</td>
<td>132</td>
<td></td>
</tr>
<tr>
<td>Encoder 1 Zero X</td>
<td>133</td>
<td></td>
</tr>
<tr>
<td>Encoder 2 Zero Y</td>
<td>134</td>
<td></td>
</tr>
<tr>
<td>Encoder 3 Zero Z</td>
<td>135</td>
<td></td>
</tr>
<tr>
<td>Tool</td>
<td>136</td>
<td></td>
</tr>
<tr>
<td>Fixture</td>
<td>137</td>
<td></td>
</tr>
<tr>
<td>Go Home (G28)</td>
<td>138</td>
<td></td>
</tr>
<tr>
<td>Part X Zero</td>
<td>139</td>
<td></td>
</tr>
<tr>
<td>Part Y Zero</td>
<td>140</td>
<td></td>
</tr>
<tr>
<td>Part Z Zero</td>
<td>141</td>
<td></td>
</tr>
<tr>
<td>Part A Zero</td>
<td>142</td>
<td></td>
</tr>
<tr>
<td>Part B Zero</td>
<td>143</td>
<td></td>
</tr>
<tr>
<td>Part C Zero</td>
<td>144</td>
<td></td>
</tr>
<tr>
<td>Tool X Touch</td>
<td>145</td>
<td></td>
</tr>
<tr>
<td>Tool Z Zero</td>
<td>146</td>
<td></td>
</tr>
<tr>
<td>Joystick Throttle</td>
<td>147</td>
<td></td>
</tr>
<tr>
<td>Tool Touch Correction Enable Off/On Toggle</td>
<td>148</td>
<td></td>
</tr>
<tr>
<td>Homing/Limits Auto Over Ride Limit/Home Switches</td>
<td>149</td>
<td></td>
</tr>
<tr>
<td>Homing/Limit Switch Over Ride (Manual)</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>SS on Act4 Off/On Toggle</td>
<td>151</td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td>152</td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td>153</td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td>154</td>
<td></td>
</tr>
<tr>
<td>Units Per Minute/Rev. Toggle</td>
<td>155</td>
<td></td>
</tr>
<tr>
<td>OEM Button</td>
<td>Function</td>
<td>Page</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Set Next Line</td>
<td>156</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Tool Path Jog Follow</td>
<td>157</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Joystick On</td>
<td>158</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Joystick Off</td>
<td>159</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Tool Path Regen Display</td>
<td>160</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Work X-Z Zero (turn)</td>
<td>161</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Coordinate Mode Abs./Incremental.</td>
<td>162</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Spindle Speed Override Increment by (percentage)</td>
<td>163</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Spindle Speed Override Decrement by (percentage)</td>
<td>164</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Laser Trigger Enable Off/On Toggle</td>
<td>165</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Laser Grid Zero</td>
<td>166</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Z Limiting Off/On Toggle Z</td>
<td>167</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Tool Change Ignore Off/On Toggle</td>
<td>168</td>
</tr>
<tr>
<td>OEM Button</td>
<td>G-Code Close current File</td>
<td>169</td>
</tr>
<tr>
<td>OEM Button</td>
<td>G-Code Re-load Last File</td>
<td>170</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog increment Selection Cycle</td>
<td>171</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Clear error Label</td>
<td>172</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Spindle CCW Off/On Toggle</td>
<td>173</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG Parallel Port Encoder 3 Jog Off/On Toggle</td>
<td>174</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG 1 Cycle Axis Controlled</td>
<td>175</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Block Delete</td>
<td>176</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Optional Stop Off/On Toggle</td>
<td>177</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Inhibit All Off/On Toggle</td>
<td>178</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Multi-function DROs Machine Coordinates Displayed</td>
<td>179</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Multi-function DROs Work + G 92</td>
<td>180</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Multi-function DROs Work</td>
<td>181</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Spindle ??? (Override cancel)Actual Off/On Toggle</td>
<td>182</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Home/Reference X Then Z (turn)</td>
<td>184</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG 1 Jogs X</td>
<td>185</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG 1 Jogs Y</td>
<td>186</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG 1 Jogs Z</td>
<td>187</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG 1 Jogs A</td>
<td>188</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG 1 Jogs B</td>
<td>189</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG 1 Jogs C</td>
<td>190</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Selection 1</td>
<td>191</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Selection 2</td>
<td>192</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Selection 3</td>
<td>193</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Selection 4</td>
<td>194</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Selection 5</td>
<td>195</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Selection 6</td>
<td>196</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Selection 7</td>
<td>197</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Selection 8</td>
<td>198</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Selection 9</td>
<td>199</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Selection 10</td>
<td>200</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Joystick Feed Rate Override Off</td>
<td>201</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Joystick Feed Rate Override Jog</td>
<td>202</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Joystick Feed Rate Override Feed</td>
<td>203</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Mode Continuous 1</td>
<td>204</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Mode Incremental 1</td>
<td>205</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Joystick On</td>
<td>206</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Joystick Off</td>
<td>207</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Tool Z Clear (turn)</td>
<td>208</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Tool X Clear (turn)</td>
<td>209</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Tool Touch Stock Correction Set to Zero</td>
<td>210</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Home/Reference X and Z (turn)</td>
<td>211</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Home/Reference X (turn)</td>
<td>212</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Home/Reference Z (turn)</td>
<td>213</td>
</tr>
<tr>
<td>OEM Button</td>
<td>G-Code Show Recent Files</td>
<td>214</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Display History</td>
<td>215</td>
</tr>
<tr>
<td>OEM Button</td>
<td>G-code Load</td>
<td>216</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Tool Post Front/Rear Toggle</td>
<td>217</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Z Limiting ON Z</td>
<td>218</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Z Limiting Off Z</td>
<td>219</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Port Bit-Test</td>
<td>220</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Torch Height Correction Off/On Toggle</td>
<td>221</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Torch Height Correction Off</td>
<td>222</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Torch Height Correction ON</td>
<td>223</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Coolant Flood ON</td>
<td>224</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Coolant Flood Off</td>
<td>225</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Coolant Mist ON</td>
<td>226</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Coolant Mist Off</td>
<td>227</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Teach File Load</td>
<td>228</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Tool Path in Machine/Work Toggle</td>
<td>229</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Wizard Display Selection Window</td>
<td>230</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Wizard is done Load normal screen</td>
<td>231</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Screen Simple/Complex Toggle</td>
<td>232</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Output 4 On</td>
<td>233</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Output 4 Off</td>
<td>234</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Output 5 On</td>
<td>235</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Output 5 Off</td>
<td>236</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Output 6 On</td>
<td>237</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Output 6 Off</td>
<td>238</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Set Help Context</td>
<td>239</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Home/Reference All Forced De-Reference</td>
<td>240</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Tangential Off/On Toggle</td>
<td>241</td>
</tr>
<tr>
<td>OEM Button</td>
<td>G 59 Work Save Current XYZ to G 59.254 XYZ</td>
<td>242</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Coordinates Machine G 53 Do G0G53</td>
<td>243</td>
</tr>
<tr>
<td>OEM Button</td>
<td>G 59 Move to G59.254 with midpoint selection</td>
<td>244</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Mode Continuous/Incremental/MPG Toggle Through</td>
<td>245</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Home/Reference All Forced</td>
<td>246</td>
</tr>
<tr>
<td>OEM Button</td>
<td>CV Off/On Toggle</td>
<td>247</td>
</tr>
<tr>
<td>OEM Button</td>
<td>CV OFF</td>
<td>248</td>
</tr>
<tr>
<td>OEM Button</td>
<td>CV On</td>
<td>249</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Inhibit X Off/On Toggle</td>
<td>250</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Inhibit Y Off/On Toggle</td>
<td>251</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Inhibit Z Off/On Toggle</td>
<td>252</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Inhibit A Off/On Toggle</td>
<td>253</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Inhibit B Off/On Toggle</td>
<td>254</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Inhibit C Off/On Toggle</td>
<td>255</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Multi-function DROs Machine/Work Toggle 2</td>
<td>256</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Inhibit All Off</td>
<td>257</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Inhibit All On</td>
<td>258</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Encoder Jog X</td>
<td>259</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Encoder Jog Y</td>
<td>260</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Encoder Jog Z</td>
<td>261</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Encoder Jog A</td>
<td>262</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Encoder Jog B</td>
<td>263</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Encoder Jog C</td>
<td>264</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Value 1</td>
<td>265</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Value 2</td>
<td>266</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Value 3</td>
<td>267</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Value 4</td>
<td>268</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Value 5</td>
<td>269</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Value 6</td>
<td>270</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Value 7</td>
<td>271</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Value 8</td>
<td>272</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Value 9</td>
<td>273</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Increment Value 10</td>
<td>274</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Mode Incremental 2</td>
<td>275</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Mode Continuous 2</td>
<td>276</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Feed Rate Increment (units)</td>
<td>277</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Feed Rate Decrement (units)</td>
<td>278</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Reverse Run</td>
<td>279</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Wizard Switch to Last Used</td>
<td>280</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG 2 Cycle Axis Controlled</td>
<td>281</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG 2 Taper Mode Off/On Toggle</td>
<td>282</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG Dual Flag Off/On Toggle</td>
<td>283</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG Shuttle mode Off/On Toggle</td>
<td>284</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Pause Preparation Moves</td>
<td>285</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Pause (remember state)</td>
<td>286</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Feed Rate Rapid Override On/Off Toggle</td>
<td>287</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Dwell Crop current now</td>
<td>288</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Set Formulas</td>
<td>289</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG Push To Stop Jog X</td>
<td>290</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG Push To Stop Jog Y</td>
<td>291</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG Push To Stop Jog Z</td>
<td>292</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG Push To Stop Jog A</td>
<td>293</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG Push To Stop Jog B</td>
<td>294</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG Push To Stop Jog C</td>
<td>295</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Motor Screw Mapping</td>
<td>296</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Dwell Unconditional crop any now</td>
<td>297</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Reserved</td>
<td>298</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Feed Rate Alternative Bypass Off/On Toggle</td>
<td>299</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Tool Post Front/Rear Toggle</td>
<td>300</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Code for OEMTriggers runs the macro in SetTriggerMacro</td>
<td>301</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG Steps Multiple/Single/Step Velocity/Velocity Only Toggle</td>
<td>302</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG Velocity Only</td>
<td>303</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG Step/Velocity</td>
<td>304</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG Steps Single</td>
<td>305</td>
</tr>
<tr>
<td>OEM Button</td>
<td>MPG Steps Multiple</td>
<td>306</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Push To Jog X Positive</td>
<td>307</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Jog Push To Jog X Negative</td>
<td>308</td>
</tr>
<tr>
<td>OEM Button</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>--------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Jog Push To Jog Y Positive</td>
<td>309</td>
<td></td>
</tr>
<tr>
<td>Jog Push To Jog Y Negative</td>
<td>310</td>
<td></td>
</tr>
<tr>
<td>Jog Push To Jog Z Positive</td>
<td>311</td>
<td></td>
</tr>
<tr>
<td>Jog Push To Jog Z Negative</td>
<td>312</td>
<td></td>
</tr>
<tr>
<td>Jog Push To Jog A Positive</td>
<td>313</td>
<td></td>
</tr>
<tr>
<td>Jog Push To Jog A Negative</td>
<td>314</td>
<td></td>
</tr>
<tr>
<td>MPG Calibrate</td>
<td>315</td>
<td></td>
</tr>
<tr>
<td>Tool Table Save</td>
<td>316</td>
<td></td>
</tr>
<tr>
<td>Work Table Save</td>
<td>317</td>
<td></td>
</tr>
<tr>
<td>Tool Path Mouse Drag Zooms Off/On Toggle</td>
<td>318</td>
<td></td>
</tr>
<tr>
<td>Tool Path Mouse Drag Pans Off/On Toggle</td>
<td>319</td>
<td></td>
</tr>
<tr>
<td>Wizard Display Selection Dialog</td>
<td>320</td>
<td></td>
</tr>
<tr>
<td>Wizard Run Newfangled</td>
<td>321</td>
<td></td>
</tr>
<tr>
<td>Emergency bailout</td>
<td>322</td>
<td></td>
</tr>
<tr>
<td>Screen Visibility of Screen 50 Off/On Toggle</td>
<td>323</td>
<td></td>
</tr>
<tr>
<td>Tool X Touch (turn)</td>
<td>324</td>
<td></td>
</tr>
<tr>
<td>Torch Volts Controlled by Spindle Step Line Off/On Toggle</td>
<td>325</td>
<td></td>
</tr>
<tr>
<td>Tool Z Touch (turn)</td>
<td>326</td>
<td></td>
</tr>
<tr>
<td>MPG Jog Mode</td>
<td>327</td>
<td></td>
</tr>
<tr>
<td>unknown</td>
<td>328</td>
<td></td>
</tr>
<tr>
<td>Jog Push To Jog B Positive</td>
<td>329</td>
<td></td>
</tr>
<tr>
<td>Jog Push To Jog B Negative</td>
<td>330</td>
<td></td>
</tr>
<tr>
<td>Jog Push To Jog C Positive</td>
<td>331</td>
<td></td>
</tr>
<tr>
<td>Jog Push To Jog C Negative</td>
<td>332</td>
<td></td>
</tr>
<tr>
<td>Multi-function DROs Distance To Go</td>
<td>333</td>
<td></td>
</tr>
<tr>
<td>Jog Push To Jog X Stop</td>
<td>334</td>
<td></td>
</tr>
<tr>
<td>Jog Push To Jog Y Stop</td>
<td>335</td>
<td></td>
</tr>
<tr>
<td>Jog Push To Jog Z Stop</td>
<td>336</td>
<td></td>
</tr>
<tr>
<td>Jog Push To Jog A Stop</td>
<td>337</td>
<td></td>
</tr>
<tr>
<td>Jog Push To Jog B Stop</td>
<td>338</td>
<td></td>
</tr>
<tr>
<td>Jog Push To Jog C Stop</td>
<td>339</td>
<td></td>
</tr>
<tr>
<td>Soft limits Set Temporary Minimum</td>
<td>340</td>
<td></td>
</tr>
<tr>
<td>Soft limits Set Temporary Maximum</td>
<td>341</td>
<td></td>
</tr>
<tr>
<td>Reserved 343-346</td>
<td>347</td>
<td></td>
</tr>
<tr>
<td>Screen Menu Bar Turn Off</td>
<td>348</td>
<td></td>
</tr>
<tr>
<td>Screen Menu Bar Turn On</td>
<td>349</td>
<td></td>
</tr>
<tr>
<td>Spindle Speed Requested Increment by (value)</td>
<td>350</td>
<td></td>
</tr>
<tr>
<td>Spindle Speed Requested Decrement by (value)</td>
<td>351</td>
<td></td>
</tr>
<tr>
<td>Jog Together X &amp; A ++</td>
<td>352</td>
<td></td>
</tr>
<tr>
<td>OEM Button</td>
<td>Description</td>
<td>Mach3 V3 #</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>--------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Jog Together X &amp; A –</td>
<td>353</td>
<td></td>
</tr>
<tr>
<td>Jog Together Y &amp; B ++</td>
<td>354</td>
<td></td>
</tr>
<tr>
<td>Jog Together Y &amp; B –</td>
<td>355</td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>Pause Feed Hold</td>
<td>1001</td>
<td></td>
</tr>
<tr>
<td>G-Code Rewind</td>
<td>1002</td>
<td></td>
</tr>
<tr>
<td>Stop</td>
<td>1003</td>
<td></td>
</tr>
<tr>
<td>Single Block</td>
<td>1004</td>
<td></td>
</tr>
<tr>
<td>Resume</td>
<td>1005</td>
<td></td>
</tr>
<tr>
<td>Teach File Edit</td>
<td>1006</td>
<td></td>
</tr>
<tr>
<td>Work Zero All</td>
<td>1007</td>
<td></td>
</tr>
<tr>
<td>Work X Zero</td>
<td>1008</td>
<td></td>
</tr>
<tr>
<td>Work Y Zero</td>
<td>1009</td>
<td></td>
</tr>
<tr>
<td>Work Z Zero</td>
<td>1010</td>
<td></td>
</tr>
<tr>
<td>Work A Zero</td>
<td>1011</td>
<td></td>
</tr>
<tr>
<td>Work B Zero</td>
<td>1012</td>
<td></td>
</tr>
<tr>
<td>Work C Zero</td>
<td>1013</td>
<td></td>
</tr>
<tr>
<td>Feed Rate Override Cancel</td>
<td>1014</td>
<td></td>
</tr>
<tr>
<td>Estimate Job</td>
<td>1015</td>
<td></td>
</tr>
<tr>
<td>Run From Here</td>
<td>1016</td>
<td></td>
</tr>
<tr>
<td>Work Go To Zeros</td>
<td>1017</td>
<td></td>
</tr>
<tr>
<td>Multi-function DROs System</td>
<td>1018</td>
<td></td>
</tr>
<tr>
<td>Verify</td>
<td>1020</td>
<td></td>
</tr>
<tr>
<td>Reset</td>
<td>1021</td>
<td></td>
</tr>
<tr>
<td>Home/Reference X</td>
<td>1022</td>
<td></td>
</tr>
<tr>
<td>Home/Reference Y</td>
<td>1023</td>
<td></td>
</tr>
<tr>
<td>Home/Reference Z</td>
<td>1024</td>
<td></td>
</tr>
<tr>
<td>Home/Reference A</td>
<td>1025</td>
<td></td>
</tr>
<tr>
<td>Home/Reference B</td>
<td>1026</td>
<td></td>
</tr>
<tr>
<td>Home/Reference C</td>
<td>1027</td>
<td></td>
</tr>
<tr>
<td>Joystick Off/On Toggle</td>
<td>1028</td>
<td></td>
</tr>
<tr>
<td>Soft limits On/Off Toggle</td>
<td>1029</td>
<td></td>
</tr>
<tr>
<td>Unknown</td>
<td>1030</td>
<td></td>
</tr>
<tr>
<td>Jog Push to Jog Until stopped</td>
<td>1031</td>
<td></td>
</tr>
</tbody>
</table>

**OEM DRO numbers**

<table>
<thead>
<tr>
<th>Type</th>
<th>New Function Description</th>
<th>Mach3 V3 #</th>
</tr>
</thead>
<tbody>
<tr>
<td>OEM DRO</td>
<td>Jog Increment Incremental Size</td>
<td>1</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Description</td>
<td>Location</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Pulse Freq.</td>
<td>2</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Jog Slow Decrement/Increment by (percentage %)</td>
<td>3</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Path In Work Coordinates Minimum X</td>
<td>4</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Path In Work Coordinates Minimum Y</td>
<td>5</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Path In Work Coordinates Minimum Z</td>
<td>6</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Path In Work Coordinates Minimum A</td>
<td>7</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Path In Work Coordinates Minimum B</td>
<td>8</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Path In Work Coordinates Minimum C</td>
<td>9</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Path In Work Coordinates Maximum X</td>
<td>10</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Path In Work Coordinates Maximum Y</td>
<td>11</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Path In Work Coordinates Maximum Z</td>
<td>12</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Path In Work Coordinates Maximum A</td>
<td>13</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Path In Work Coordinates Maximum B</td>
<td>14</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Path In Work Coordinates Maximum C</td>
<td>15</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>G 92 Reposition/Threading X</td>
<td>16</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>G 92 Reposition/Threading Y</td>
<td>17</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>G 92 Reposition/Threading Z</td>
<td>18</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>G 92 Reposition/Threading A</td>
<td>19</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>G 92 Reposition/Threading B</td>
<td>20</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>G 92 Reposition/Threading C</td>
<td>21</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Queue Depth</td>
<td>22</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Scale Time</td>
<td>23</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Spindle Pulse Width Modulation Base</td>
<td>24</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Torch Height Correction Speed</td>
<td>25</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Torch Height Correction Speed (Current)</td>
<td>26</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Torch Height Correction Speed Maximum</td>
<td>27</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Processor CPU Load</td>
<td>28</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Encoder 1 Position X</td>
<td>29</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Encoder 2 Position Y</td>
<td>30</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Encoder 3 Position Z</td>
<td>31</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Length Minus Wear</td>
<td>32</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Home Off Distance X</td>
<td>33</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Home Off Distance Y</td>
<td>34</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Home Off Distance Z</td>
<td>35</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Home Off Distance A</td>
<td>36</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Home Off Distance B</td>
<td>37</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Home Off Distance C</td>
<td>38</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Spindle Speed TRUE</td>
<td>39</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Worst Case Interrupt</td>
<td>40</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Description</td>
<td>Location</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool X</td>
<td>41</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Length</td>
<td>42</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Diameter</td>
<td>43</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Tip Radius</td>
<td>44</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Touch Correction</td>
<td>45</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Fixture Number</td>
<td>46</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Part X</td>
<td>47</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Part Y</td>
<td>48</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Part Z</td>
<td>49</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Part A</td>
<td>50</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Part B</td>
<td>51</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Part C</td>
<td>52</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Processor CPU Speed</td>
<td>53</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Safe Z Height Z</td>
<td>54</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Feed Rate Overridden</td>
<td>55</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Spindle Pulley</td>
<td>56</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Spindle Pulley Maximum Speed</td>
<td>57</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Feed Velocity Per Revolution</td>
<td>58</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Scale X</td>
<td>59</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Scale Y</td>
<td>60</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Scale Z</td>
<td>61</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Scale A</td>
<td>62</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Scale B</td>
<td>63</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Scale C</td>
<td>64</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Torch Height Correction Speed Minimum</td>
<td>65</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Threading Entrance Angle</td>
<td>66</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Limits Entrance Points Maximum</td>
<td>67</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Rotational axis Time Error</td>
<td>68</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Threading Trigger Angle</td>
<td>69</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Time Correction Derivative ?? ?? Threading diagnostic??</td>
<td>70</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Spindle Pulse Interrupts Per Rev.</td>
<td>71</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Spindle Pulse Count</td>
<td>72</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Spindle Adder</td>
<td>73</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Spindle Speed Override (percentage of requested)</td>
<td>74</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Stock Size</td>
<td>75</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Laser Grid Spacing X</td>
<td>76</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Laser Grid Spacing Y</td>
<td>77</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Repetitions</td>
<td>78</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>OEM DRO Safe Z Height Decrement Z</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Z Limiting Distance Z</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Port Bit-test</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Torch Height Correction Feed Rate Limit %</td>
<td>82</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Coordinates Machine ABS X</td>
<td>83</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Coordinates Machine ABS Y</td>
<td>84</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Coordinates Machine ABS Z</td>
<td>85</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Coordinates Machine ABS A</td>
<td>86</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Coordinates Machine ABS B</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Coordinates Machine ABS C</td>
<td>88</td>
<td></td>
</tr>
<tr>
<td>OEM DRO CV Distance Tolerance</td>
<td>89</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Spindle Pulse Number Of Disc Slots</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>OEM DRO G 73 Peck Drill Pull-off Distance</td>
<td>91</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Tangential Lift Angle</td>
<td>92</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Tangential Lift Level Z</td>
<td>93</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Reserved</td>
<td>94</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Reserved</td>
<td>95</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Reserved</td>
<td>96</td>
<td></td>
</tr>
<tr>
<td>OEM DRO CV Rate (CV applied at this rate)</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Feed Rate Decrement/ Increment by (units)</td>
<td>98</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Spindle PWM Control Ratio</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Encoder 4 Position A</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>OEM DRO MPG 1 Count</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>OEM DRO MPG 2 Count</td>
<td>102</td>
<td></td>
</tr>
<tr>
<td>OEM DRO MPG 3 Count</td>
<td>103</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Feed Rate Rapid</td>
<td>104</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Tool Diameter (current)</td>
<td>105</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Tool Tip Direction</td>
<td>106</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Tool Tip Nose Radius</td>
<td>107</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Tool Offset X (current)</td>
<td>108</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Tool Offset Z (current)</td>
<td>109</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Tool Wear X</td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Tool Wear Z</td>
<td>111</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Tool Turret Angle</td>
<td>112</td>
<td></td>
</tr>
<tr>
<td>OEM DRO MPG 1 Velocity</td>
<td>113</td>
<td></td>
</tr>
<tr>
<td>OEM DRO MPG 2 Velocity</td>
<td>114</td>
<td></td>
</tr>
<tr>
<td>OEM DRO MPG 3 Velocity</td>
<td>115</td>
<td></td>
</tr>
<tr>
<td>OEM DRO MPG Taper Angle</td>
<td>116</td>
<td></td>
</tr>
<tr>
<td>OEM DRO Spindle Speed as Constant Surface Speed</td>
<td>117</td>
<td></td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Coordinate System Rotation Angle X/Y (G68)</td>
<td>118</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Laser Grid Spacing X/Y</td>
<td>119</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Jog Increments That Can be Buffered</td>
<td>120</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Spindle Pulley Minimum Speed</td>
<td>121</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Feed Rate Alternative</td>
<td>122</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Post Between Front and Rear</td>
<td>123</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>MPG 1 Velocity Current</td>
<td>124</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>MPG 2 Velocity Current</td>
<td>125</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>MPG Step Jog Feed Rate</td>
<td>126</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Encoder Error Encoder 1 X</td>
<td>127</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Encoder Error Encoder 2 Y</td>
<td>128</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Encoder Error Encoder 3 Z</td>
<td>129</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Processor Interrupt Handler</td>
<td>130</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Laser SLS Distance</td>
<td>131</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Look ahead</td>
<td>133</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Input Modbus 64</td>
<td>146</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Input Modbus 65</td>
<td>147</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Input Modbus 66</td>
<td>148</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Input Modbus 67</td>
<td>149</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Soft limit Maximum X</td>
<td>150</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Soft limit Maximum Y</td>
<td>151</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Soft limit Maximum Z</td>
<td>152</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Soft limit Maximum A</td>
<td>153</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Soft limit Maximum B</td>
<td>154</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Soft limit Maximum C</td>
<td>155</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Soft limit Minimum X</td>
<td>156</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Soft limit Minimum Y</td>
<td>157</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Soft limit Minimum Z</td>
<td>158</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Soft limit Minimum A</td>
<td>159</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Soft limit Minimum B</td>
<td>160</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Soft limit Minimum C</td>
<td>161</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Post Distance From Front to Rear</td>
<td>162</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Encoder 1 Units Position</td>
<td>170</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Encoder 2 Units Position</td>
<td>171</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Encoder 3 Units Position</td>
<td>172</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Encoder 4 Units Position</td>
<td>173</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Part X Touch Tool Table X Radius/Diameter X</td>
<td>175</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Part Z Touch Tool Table Z</td>
<td>176</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Torch Pierce Delay</td>
<td>177</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Work X</td>
<td>178</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>-----</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Work Y</td>
<td>179</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Work Z</td>
<td>180</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Work A</td>
<td>181</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Work B</td>
<td>182</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Work C</td>
<td>183</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>G 92/52 Control Point X</td>
<td>184</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>G 92/52 Control Point Y</td>
<td>185</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>G 92/52 Control Point Z</td>
<td>186</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>G 92/52 Control Point A</td>
<td>187</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>G 92/52 Control Point B</td>
<td>188</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>G 92/52 Control Point C</td>
<td>189</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>G 28 Home Location X</td>
<td>190</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>G 28 Home Location Y</td>
<td>191</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>G 28 Home Location Z</td>
<td>192</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>G 28 Home Location A</td>
<td>193</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>G 28 Home Location B</td>
<td>194</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>G 28 Home Location C</td>
<td>195</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Multi-function DROs Distance To Go X</td>
<td>196</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Multi-function DROs Distance To Go Y</td>
<td>197</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Multi-function DROs Distance To Go Z</td>
<td>198</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Multi-function DROs Distance To Go A</td>
<td>199</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Multi-function DROs Distance To Go B</td>
<td>200</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Multi-function DROs Distance To Go C</td>
<td>201</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Spindle Speed Overridden (DRO 817 * DRO 74)</td>
<td>202</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Spindle Pulley Reversed Direction</td>
<td>203</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool X (current) (turn)</td>
<td>204</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Encoder Reading Corrected by Offsets X</td>
<td>208</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Encoder Reading Corrected by Offsets Y</td>
<td>209</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Encoder Reading Corrected by Offsets Z</td>
<td>210</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Path In Machine Coordinates Maximum X</td>
<td>211</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Path In Machine Coordinates Maximum Y</td>
<td>212</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Path In Machine Coordinates Maximum Z</td>
<td>213</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Path In Machine Coordinates Minimum X</td>
<td>214</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Path In Machine Coordinates Minimum Y</td>
<td>215</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Path In Machine Coordinates Minimum Z</td>
<td>216</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Spindle Dwell Spin Up CW</td>
<td>217</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Processor Brains Execution Time</td>
<td>220</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Probe Tip Diameter</td>
<td>221</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Description</td>
<td>Value</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>-------</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Sub Program Depth</td>
<td>222</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Feed Rate Rapid Override</td>
<td>223</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Macros Running Number of</td>
<td>224</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Multi-function DROs Machine/Work/DTG X</td>
<td>800</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Multi-function DROs Machine/Work/DTG Y</td>
<td>801</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Multi-function DROs Machine/Work/DTG Z</td>
<td>802</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Multi-function DROs Machine/Work/DTG A</td>
<td>803</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Multi-function DROs Machine/Work/DTG B</td>
<td>804</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Multi-function DROs Machine/Work/DTG C</td>
<td>805</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Feed Rate Velocity X</td>
<td>806</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Feed Rate Velocity Y</td>
<td>807</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Feed Rate Velocity Z</td>
<td>808</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Feed Rate Velocity A</td>
<td>809</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Feed Rate Velocity B</td>
<td>810</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Feed Rate Velocity C</td>
<td>811</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>CV Blended Velocity</td>
<td>813</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Elapsed Time</td>
<td>814</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Estimate Job</td>
<td>815</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>G-code Line Number (current)</td>
<td>816</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Spindle Speed Requested</td>
<td>817</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Feed Rate</td>
<td>818</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Feed Rate Override Decrement/Increment by (percentage)</td>
<td>821</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Number</td>
<td>824</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Rotational Axis Diameter A</td>
<td>825</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Rotational Axis Diameter B</td>
<td>826</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Rotational Axis Diameter C</td>
<td>827</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Jog Increment Incremental Size</td>
<td>828</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Fixture Offset X</td>
<td>830</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Fixture Offset Y</td>
<td>831</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Fixture Offset Z</td>
<td>832</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Fixture Offset A</td>
<td>833</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Fixture Offset B</td>
<td>834</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Fixture Offset C</td>
<td>835</td>
</tr>
<tr>
<td>OEM DRO</td>
<td>Tool Length (current)</td>
<td>836</td>
</tr>
</tbody>
</table>

**OEM LED numbers**
<table>
<thead>
<tr>
<th>Type</th>
<th>New Function Description</th>
<th>Mach3 V3 #</th>
</tr>
</thead>
<tbody>
<tr>
<td>OEM OEM LED</td>
<td>G 92 in Effect</td>
<td>10</td>
</tr>
<tr>
<td>OEM OEM LED</td>
<td>Spindle CCW/CW (if either is requested)</td>
<td>11</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Coolant Mist</td>
<td>12</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Coolant Flood</td>
<td>13</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Jog Mode Continuous</td>
<td>14</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Jog Mode Incremented</td>
<td>15</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Multi-function DROs Machine Coordinates Displayed</td>
<td>16</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Feed Rate Override</td>
<td>17</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Estimating Job</td>
<td>18</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Emergency</td>
<td>19</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Rotational Axis Diameter A Correction A</td>
<td>20</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Rotational Axis Diameter B Correction B</td>
<td>21</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Rotational Axis Diameter C Correction C</td>
<td>22</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Soft limits On</td>
<td>23</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Torch Height Correction Enable</td>
<td>24</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Spindle Speed TRUE Accel.</td>
<td>25</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Spindle Speed TRUE Decel.</td>
<td>26</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Tool Path is On</td>
<td>27</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Tool</td>
<td>28</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Part</td>
<td>29</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Joystick Throttle Is Slow Jog</td>
<td>30</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Joystick Throttle Is Feed Rate</td>
<td>31</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Homing/Limits Auto Over Ride Limit/Home Switches</td>
<td>33</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Homing/Limit Switch Over Ride (Manual)</td>
<td>34</td>
</tr>
<tr>
<td>OEM LED</td>
<td>SS on Act4</td>
<td>35</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Torch Arc Good</td>
<td>36</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Torch Up Active</td>
<td>37</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Torch Down Active</td>
<td>38</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Feed Rate Per Revolution (G95)</td>
<td>39</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Feed Rate Per Minute (G94)</td>
<td>40</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Scale X</td>
<td>41</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Scale Y</td>
<td>42</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Scale Z</td>
<td>43</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Scale A</td>
<td>44</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Scale B</td>
<td>45</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Scale C</td>
<td>46</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Coordinate Mode Abs.</strong></td>
<td>48</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Coordinate Mode Incremental.</strong></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Threading Sync mode</strong></td>
<td>50</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Laser Trigger Enabled</strong></td>
<td>51</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Z Limiting ON Z</strong></td>
<td>52</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Tool Change Ignore</strong></td>
<td>53</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED CV On G64 Mode Active (Constant Velocity)</strong></td>
<td>54</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Repetitions Enabled (M30)</strong></td>
<td>55</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Exact Stop on (G61 Mode Active)</strong></td>
<td>56</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED MPG Jog Mode</strong></td>
<td>57</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Jog Rapid Enabled (shift key active)</strong></td>
<td>58</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED MPG 1 Jogs X</strong></td>
<td>59</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED MPG 1 Jogs Y</strong></td>
<td>60</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED MPG 1 Jogs Z</strong></td>
<td>61</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED MPG 1 Jogs A</strong></td>
<td>62</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED MPG 1 Jogs B</strong></td>
<td>63</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED MPG 1 Jogs C</strong></td>
<td>64</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Optional Stop</strong></td>
<td>65</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Block Delete</strong></td>
<td>66</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Inhibit All</strong></td>
<td>67</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Threading Feed Related to True Spindle Speed</strong></td>
<td>68</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Threading Index signal</strong></td>
<td>69</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Torch Height Correction ON</strong></td>
<td>70</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Spindle Speed Stable</strong></td>
<td>71</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED IJ Abs. Mode</strong></td>
<td>72</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED IJ Inc. Mode</strong></td>
<td>73</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Teaching File is Open</strong></td>
<td>74</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Offset In Effect</strong></td>
<td>75</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Output 4 Active</strong></td>
<td>77</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Output 5 Active</strong></td>
<td>78</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Output 6 Active</strong></td>
<td>79</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Pause</strong></td>
<td>80</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Tangential Control Active</strong></td>
<td>81</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Single Block</strong></td>
<td>82</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Jog Enable On</strong></td>
<td>83</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED CV Enabled</strong></td>
<td>84</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Enhanced Pulsing Not in Use</strong></td>
<td>85</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Jog Inhibit X</strong></td>
<td>86</td>
</tr>
<tr>
<td></td>
<td><strong>OEM LED Jog Inhibit Y</strong></td>
<td>87</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Jog Inhibit Z</td>
<td>88</td>
</tr>
<tr>
<td>---------</td>
<td>----------------</td>
<td>----</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Jog Inhibit A</td>
<td>89</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Jog Inhibit B</td>
<td>90</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Jog Inhibit C</td>
<td>91</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Diameter mode Active (turn)</td>
<td>92</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Timing Signal Active</td>
<td>93</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Hotkeys enabled</td>
<td>94</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Units Per Minute Mode</td>
<td>95</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Units Per Rev.: Mode</td>
<td>96</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Reverse Run</td>
<td>97</td>
</tr>
<tr>
<td>OEM LED</td>
<td>MPG 2 Jogs X</td>
<td>98</td>
</tr>
<tr>
<td>OEM LED</td>
<td>MPG 2 Jogs Y</td>
<td>99</td>
</tr>
<tr>
<td>OEM LED</td>
<td>MPG 2 Jogs Z</td>
<td>100</td>
</tr>
<tr>
<td>OEM LED</td>
<td>MPG 2 Jogs A</td>
<td>101</td>
</tr>
<tr>
<td>OEM LED</td>
<td>MPG 2 Jogs B</td>
<td>102</td>
</tr>
<tr>
<td>OEM LED</td>
<td>MPG 2 Jogs C</td>
<td>103</td>
</tr>
<tr>
<td>OEM LED</td>
<td>MPG 2 Taper Mode Active</td>
<td>104</td>
</tr>
<tr>
<td>OEM LED</td>
<td>MPG Dual MPGs In Use</td>
<td>105</td>
</tr>
<tr>
<td>OEM LED</td>
<td>MPG Shuttle mode In Use</td>
<td>106</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Spindle Speed as Constant Surface Speed G96</td>
<td>107</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Coordinate System Rotation Angle Active X/Y (G68)</td>
<td>108</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Feed Rate Rapid Override</td>
<td>109</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Rotational axis Formula Mapping</td>
<td>110</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Pause Feed Hold</td>
<td>111</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Reverse Run</td>
<td>112</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Feed Rate Alternative In Use</td>
<td>113</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Tool Post Front Selected</td>
<td>114</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Tool Post Rear Selected</td>
<td>115</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Spindle Running (CCW Only)</td>
<td>116</td>
</tr>
<tr>
<td>OEM LED</td>
<td>MPG Velocity Only</td>
<td>117</td>
</tr>
<tr>
<td>OEM LED</td>
<td>MPG Step/Velocity</td>
<td>118</td>
</tr>
<tr>
<td>OEM LED</td>
<td>MPG Steps Single</td>
<td>119</td>
</tr>
<tr>
<td>OEM LED</td>
<td>MPG Steps Multiple</td>
<td>120</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Jog Switch Active X Positive</td>
<td>121</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Jog Switch Active X Negative</td>
<td>122</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Jog Switch Active Y Positive</td>
<td>123</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Jog Switch Active Y Negative</td>
<td>124</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Jog Switch Active Z Positive</td>
<td>125</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Jog Switch Active Z Negative</td>
<td>126</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td></td>
<td>Jog Switch Active A Positive</td>
<td>127</td>
</tr>
<tr>
<td></td>
<td>Jog Switch Active A Negative</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>Tool Path Mouse Drag Zooms</td>
<td>129</td>
</tr>
<tr>
<td></td>
<td>Tool Path Mouse Drag Pans</td>
<td>130</td>
</tr>
<tr>
<td></td>
<td>MPG 3 Jogs X</td>
<td>131</td>
</tr>
<tr>
<td></td>
<td>MPG 3 Jogs Y</td>
<td>132</td>
</tr>
<tr>
<td></td>
<td>MPG 3 Jogs Z</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td>MPG 3 Jogs A</td>
<td>134</td>
</tr>
<tr>
<td></td>
<td>MPG 3 Jogs B</td>
<td>135</td>
</tr>
<tr>
<td></td>
<td>MPG 3 Jogs C</td>
<td>136</td>
</tr>
<tr>
<td></td>
<td>Macro Is running</td>
<td>162</td>
</tr>
<tr>
<td></td>
<td>Multi-function DROs Distance To Go</td>
<td>163</td>
</tr>
<tr>
<td></td>
<td>Spindle CW</td>
<td>164</td>
</tr>
<tr>
<td></td>
<td>Spindle CCW</td>
<td>165</td>
</tr>
<tr>
<td></td>
<td>Conditions abnormal</td>
<td>166</td>
</tr>
<tr>
<td></td>
<td>CV Distance Tolerance On</td>
<td>168</td>
</tr>
<tr>
<td></td>
<td>Regen in progress</td>
<td>179</td>
</tr>
<tr>
<td></td>
<td>Tool change Auto TC mode</td>
<td>184</td>
</tr>
<tr>
<td></td>
<td>Tool Change Stop&amp;Wait Mode</td>
<td>185</td>
</tr>
<tr>
<td></td>
<td>Reset</td>
<td>800</td>
</tr>
<tr>
<td></td>
<td>Units Inch</td>
<td>801</td>
</tr>
<tr>
<td></td>
<td>Units MMs</td>
<td>802</td>
</tr>
<tr>
<td></td>
<td>Processor Idle</td>
<td>803</td>
</tr>
<tr>
<td></td>
<td>Start</td>
<td>804</td>
</tr>
<tr>
<td></td>
<td>Pause Feed Hold</td>
<td>805</td>
</tr>
<tr>
<td></td>
<td>Tool Change In Process</td>
<td>806</td>
</tr>
<tr>
<td></td>
<td>Home/Reference X Warning</td>
<td>807</td>
</tr>
<tr>
<td></td>
<td>Home/Reference Y Warning</td>
<td>808</td>
</tr>
<tr>
<td></td>
<td>Home/Reference Z Warning</td>
<td>809</td>
</tr>
<tr>
<td></td>
<td>Home/Reference A Warning</td>
<td>810</td>
</tr>
<tr>
<td></td>
<td>Home/Reference B Warning</td>
<td>811</td>
</tr>
<tr>
<td></td>
<td>Home/Reference C Warning</td>
<td>812</td>
</tr>
<tr>
<td></td>
<td>Dwell (G04 and Spindle)</td>
<td>813</td>
</tr>
<tr>
<td></td>
<td>Joystick Enable</td>
<td>814</td>
</tr>
<tr>
<td></td>
<td>Fixture</td>
<td>816</td>
</tr>
<tr>
<td></td>
<td>Input 1 Active</td>
<td>821</td>
</tr>
<tr>
<td></td>
<td>Input 2 Active</td>
<td>822</td>
</tr>
<tr>
<td></td>
<td>Input 3 Active</td>
<td>823</td>
</tr>
<tr>
<td></td>
<td>Input 4 Active</td>
<td>824</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Description</td>
<td>Code</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td></td>
<td>Digitize Input</td>
<td>825</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Spindle Index</td>
<td>826</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Homing/Limit Switch Triggered</td>
<td>827</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Limits Active Positive X</td>
<td>828</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Limits Active Negative X</td>
<td>829</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Home Negative X</td>
<td>830</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Limits Active Positive Y</td>
<td>831</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Limits Active Negative Y</td>
<td>832</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Home Negative Y</td>
<td>833</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Limits Active Positive Z</td>
<td>834</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Limits Active Negative Z</td>
<td>835</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Home Negative Z</td>
<td>836</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Limits Active Positive A</td>
<td>837</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Limits Active Negative A</td>
<td>838</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Home Negative A</td>
<td>839</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Limits Active Positive B</td>
<td>840</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Limits Active Negative B</td>
<td>841</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Home Negative B</td>
<td>842</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Limits Active Positive C</td>
<td>843</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Limits Active Negative C</td>
<td>844</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Home Negative C</td>
<td>845</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Enable 1</td>
<td>846</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Enable 2</td>
<td>847</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Enable 3</td>
<td>848</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Enable 4</td>
<td>849</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Enable 5</td>
<td>850</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Enable 6</td>
<td>851</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Output 1/Extn1 Active</td>
<td>852</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Output 2/Extn2 Active</td>
<td>853</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Output 3/Extn3 Active</td>
<td>854</td>
</tr>
<tr>
<td>OEM LED</td>
<td>Digitize Output</td>
<td>855</td>
</tr>
<tr>
<td>OEM LED</td>
<td>System movement In effect or imminent</td>
<td>999</td>
</tr>
</tbody>
</table>