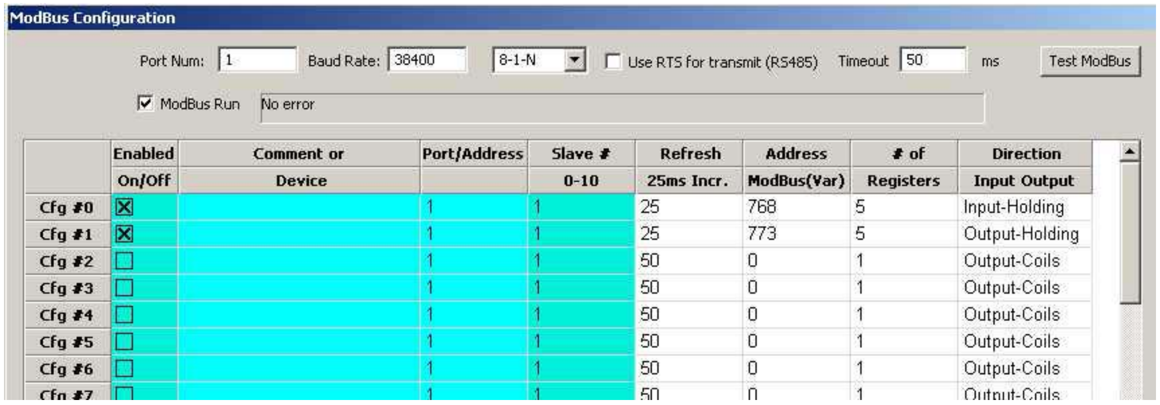
Example of Mach Lathe Setup, with 12 position turret, PLC control

Modbus Setup: PLC Used - AD 405 Series PLC with DCM module (Modbus RTU)



ModBus Configuration

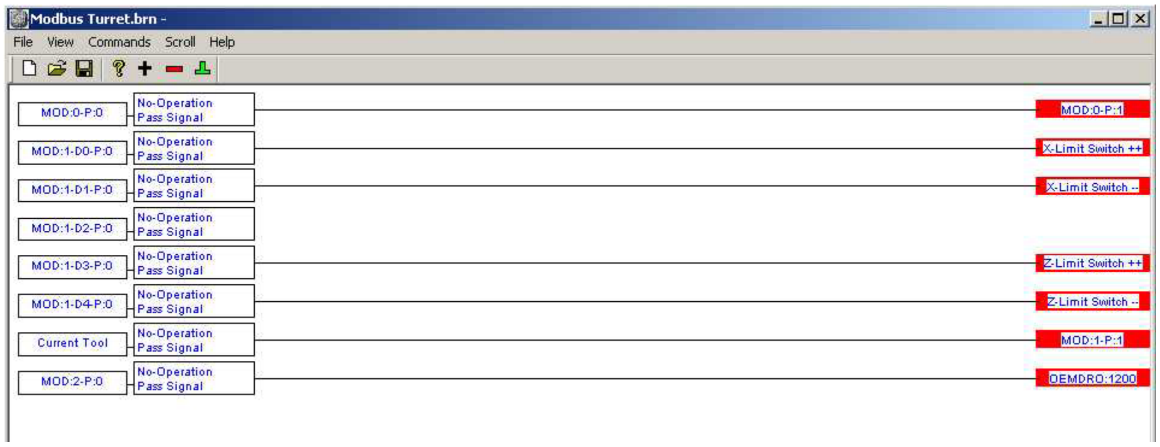
Port Num: 1   Baud Rate: 38400   8-1-N   ☐ Use RTS for transmit (RS485)   Timeout 50 ms   Test ModBus

☑ ModBus Run   No error

	Enabled	Comment or	Port/Address	Slave #	Refresh	Address	# of	Direction
	On/Off	Device		0-10	25ms Incr.	ModBus(Var)	Registers	Input Output
Cfg #0	☒		1	1	25	768	5	Input-Holding
Cfg #1	☒		1	1	25	773	5	Output-Holding
Cfg #2	☐		1	1	50	0	1	Output-Coils
Cfg #3	☐		1	1	50	0	1	Output-Coils
Cfg #4	☐		1	1	50	0	1	Output-Coils
Cfg #5	☐		1	1	50	0	1	Output-Coils
Cfg #6	☐		1	1	50	0	1	Output-Coils
Cfg #7	☐		1	1	50	0	1	Output-Coils

Line 1: Cfg#0 Get 5 – 16 bits words from V1400-V1404 (Octal) in the PLC
Line 2: Cfg#1 Send 5 – 16 bit words to V1405-V1411 (Octal)
I believe the DL06 and DL05 have Modbus Holding registers in the same locations, but check manual to be sure..

Brains and Macro Programming:



Modbus Turret.brn -

MOD:0-P:0	No-Operation Pass Signal	MOD:0-P:1
MOD:1-D0-P:0	No-Operation Pass Signal	X-Limit Switch ++
MOD:1-D1-P:0	No-Operation Pass Signal	X-Limit Switch --
MOD:1-D2-P:0	No-Operation Pass Signal	
MOD:1-D3-P:0	No-Operation Pass Signal	Z-Limit Switch ++
MOD:1-D4-P:0	No-Operation Pass Signal	Z-Limit Switch --
Current Tool	No-Operation Pass Signal	MOD:1-P:1
MOD:2-P:0	No-Operation Pass Signal	OEMDRO:1200

Line1: Get word 0 from Config 0 (V1400) and send to Modbus word 0 in Config 1 (V1405) This has the Mach S/W looping data from the PLC - back to the PLC. If the data flow stops, the PLC will recognize that the connection has been lost and shut down the lathe. This along with an input from the charge pump to a PLC input insures that the Mach S/W and the PLC are tightly interlocked.

Line2: Get bit D0, from word 1 (V1401), Config 0 and send to the X++ Limit Sw internal bit

Line3: Get bit D1, from word 1 (V1401), Config 0 and send to the X—Limit Sw internal bit

Line 4: Nothing happening

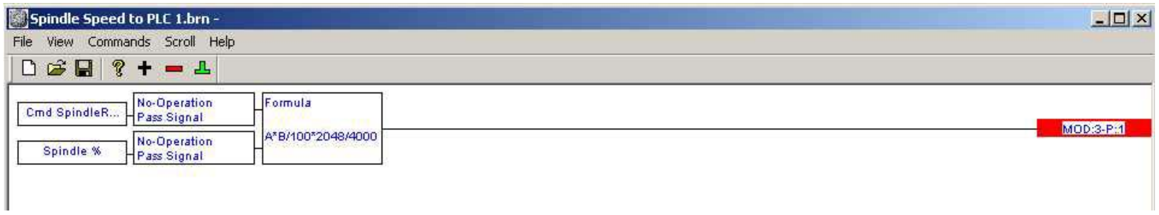
Line 5: Get bit D3, from word 1 (V1401), Config 0 and send to the Z++ limit Sw internal bit

Line 6: Get bit D4, from word 1 (V1401), Config 0 and send to the Z—limit Sw internal bit

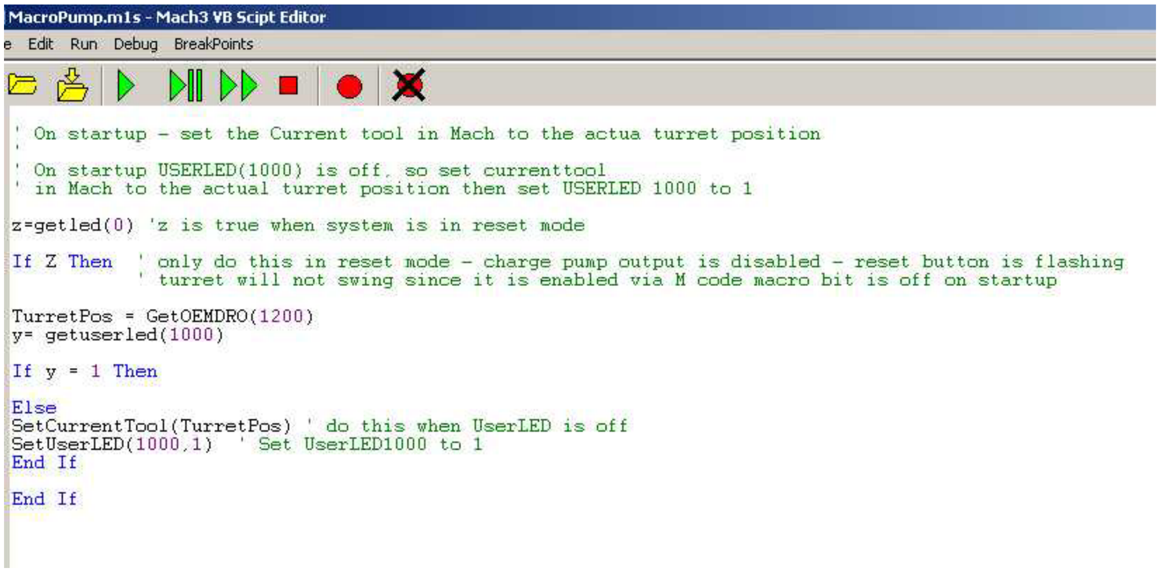
Note – these axis limits are meant to inform the software that a limit has been hit. On this machine the limits are also hardwired into the controls such that hitting a limit creates a fast stop by braking the servo motors and ramping down the spindle drive as fast as possible.

Line 7: Get the current tool (number in the box in the tool page) and send to word 1, config 1(V1406) This will let the PLC know what tool is desired by Mach.

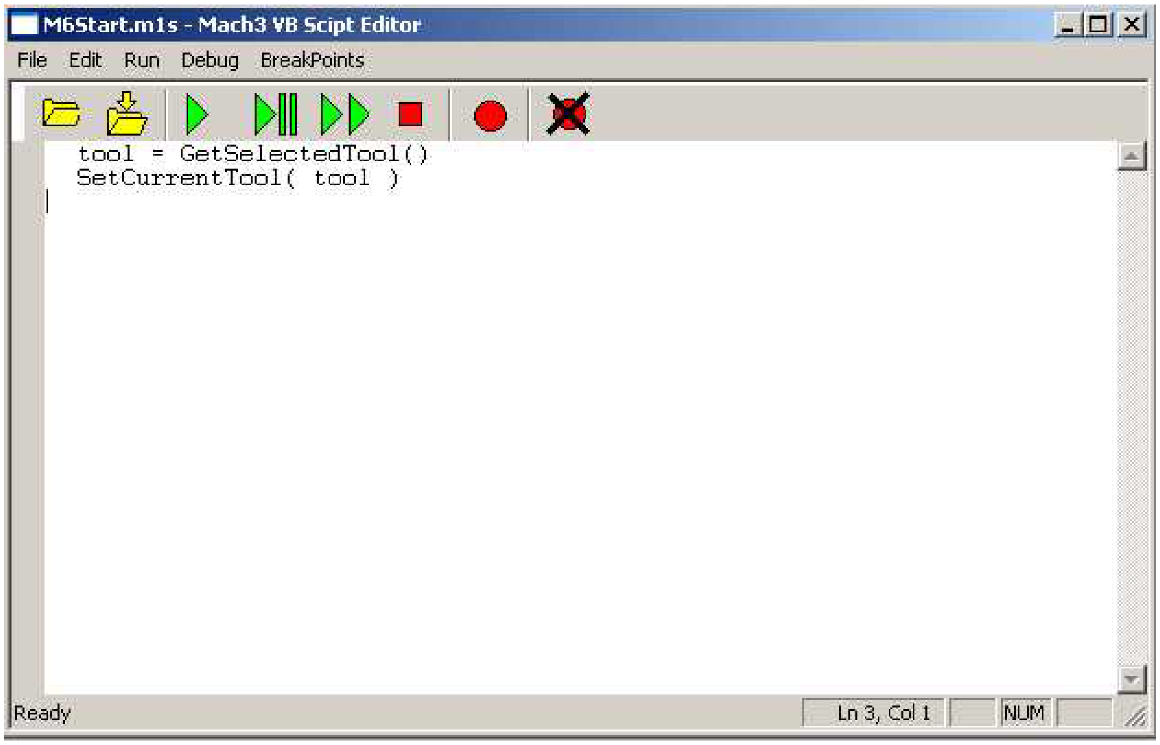
Line 8: Get word 2 (V1402) from the PLC and send to OEMDRO:1200 to inform Mach S/we where the turret is currently positioned (which tool is active).



Cmd Spindle is the speed in RPM set by the Sxxx command. The Spindle % is a number 0-100 that is the spindle override in percent. +/-2048 is the value sent to the analog output card depending on direction. 4000 is the spindle RPM when a 10V signal is applied to the spindle drive analog in terminal. This calculated value is sent to word 3, config 1 (V1410)

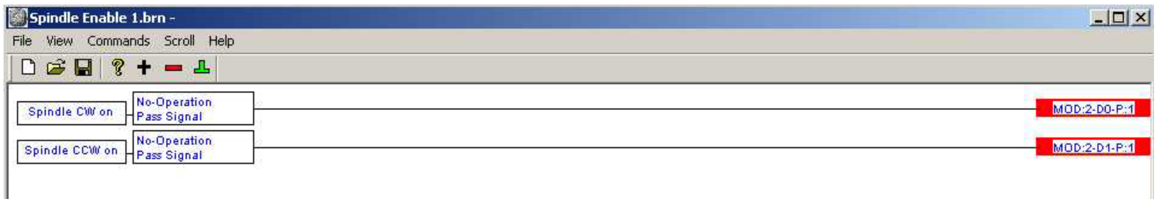

MacroPump.m1s - Mach3 VB Scipt Editor
e Edit Run Debug BreakPoints

```
' On startup - set the Current tool in Mach to the actua turret position
'
' On startup USERLED(1000) is off, so set currenttool
' in Mach to the actual turret position then set USERLED 1000 to 1

z=getled(0) 'z is true when system is in reset mode

If Z Then  ' only do this in reset mode - charge pump output is disabled - reset button is flashing
           ' turret will not swing since it is enabled via M code macro bit is off on startup

TurretPos = GetOEMDRO(1200)
y= getuserled(1000)

If y = 1 Then

Else
SetCurrentTool(TurretPos) ' do this when UserLED is off
SetUserLED(1000,1)  ' Set UserLED1000 to 1
End If

End If
```

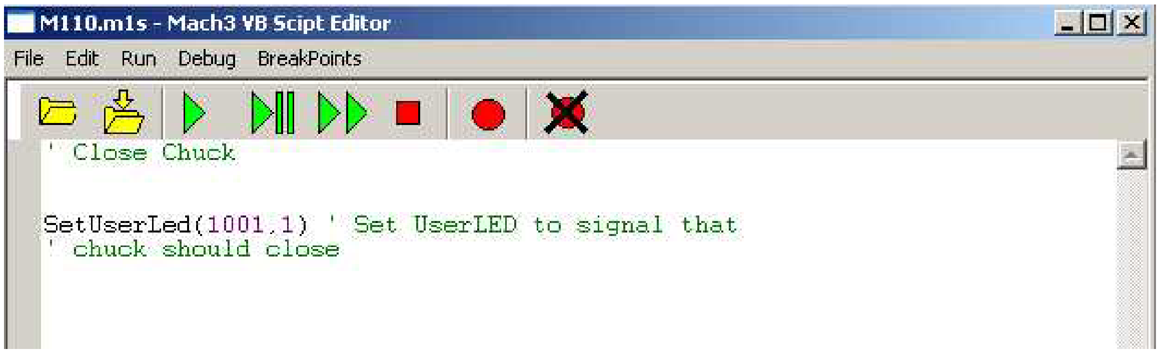

M6Start.m1s - Mach3 VB Scipt Editor
File Edit Run Debug BreakPoints
Ready
Ln 3, Col 1
NUM

```
tool = GetSelectedTool()
SetCurrentTool( tool )
```

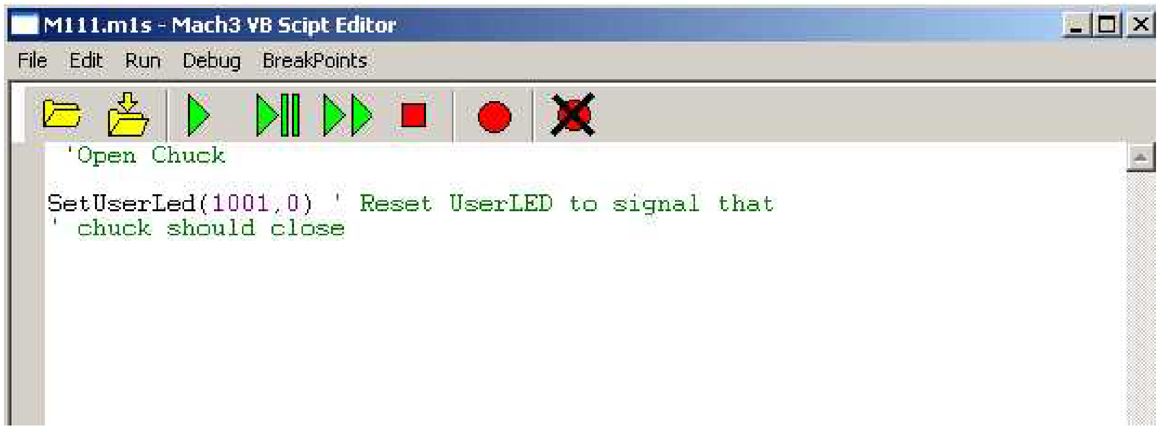
When the a tool change occurs, via a T0x0x command, M6Start.m1s runs and the CurrentTool number (shown in the tool window DRO) is changed to the tool stated in the T0x0x command.



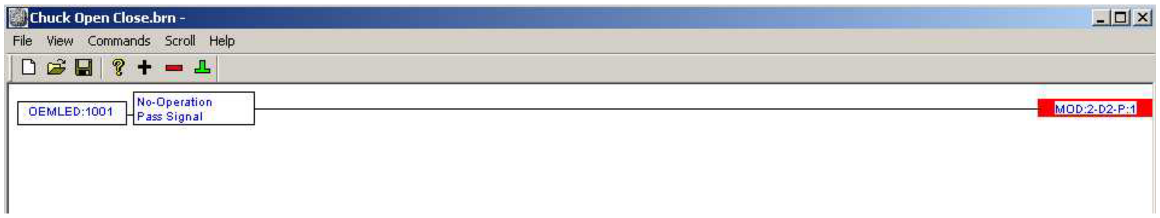
When M3 or M4 is run via Gcode, the CW or CCW bits are turned on, these bits are sent to the PLC. Set bits in Word 2 of config 1, (V1407), bits D0 and D1 to control the spindle direction via the PLC.


M110.m1s - Mach3 VB Scipt Editor
File Edit Run Debug BreakPoints

```
' Close Chuck


SetUserLed(1001,1) ' Set UserLED to signal that
' chuck should close
```

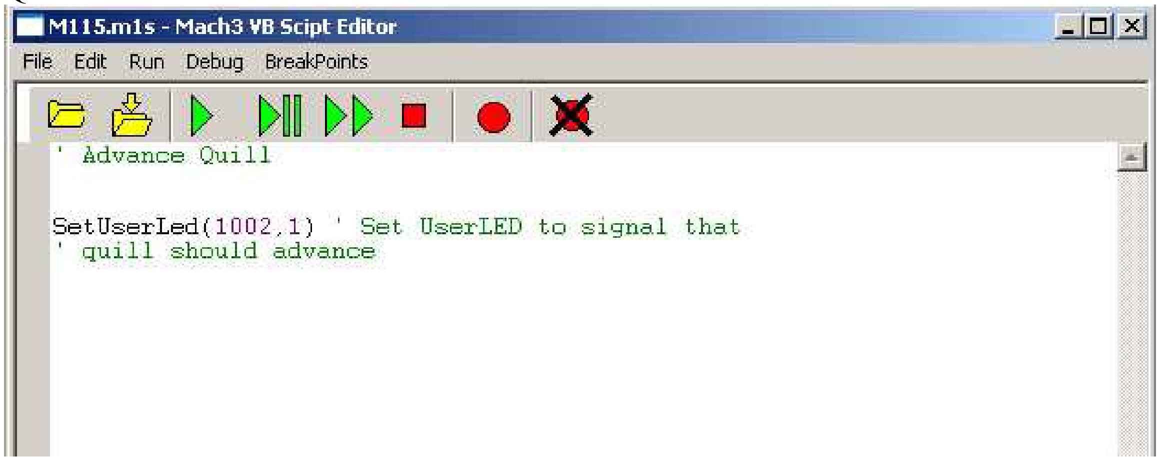

M111.m1s - Mach3 VB Scipt Editor
File Edit Run Debug BreakPoints

```
 'Open Chuck

SetUserLed(1001,0) ' Reset UserLED to signal that
' chuck should close
```

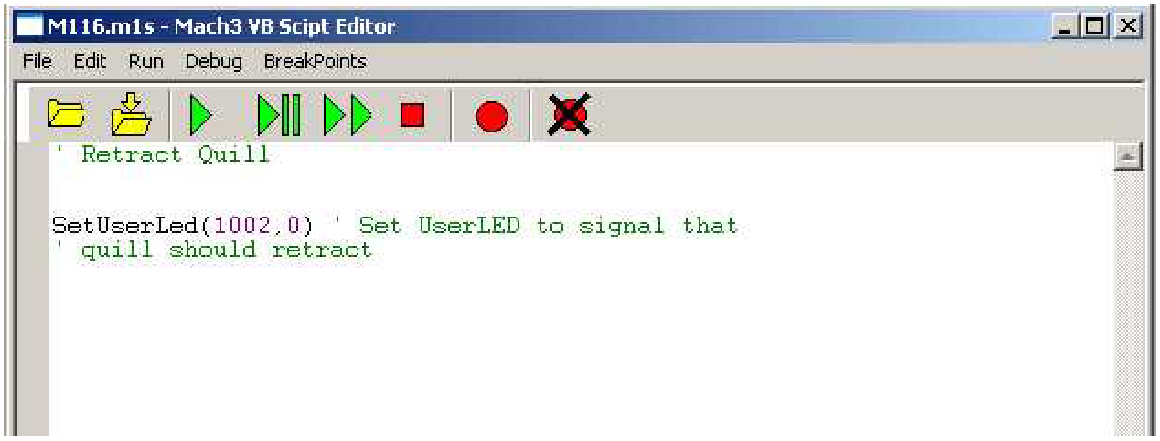


Get OEMLED:1001 which is controlled via M110 and M111 Macros and send to word 2, bit D2 in config 1, (V1407 bit 2) to control the chuck.

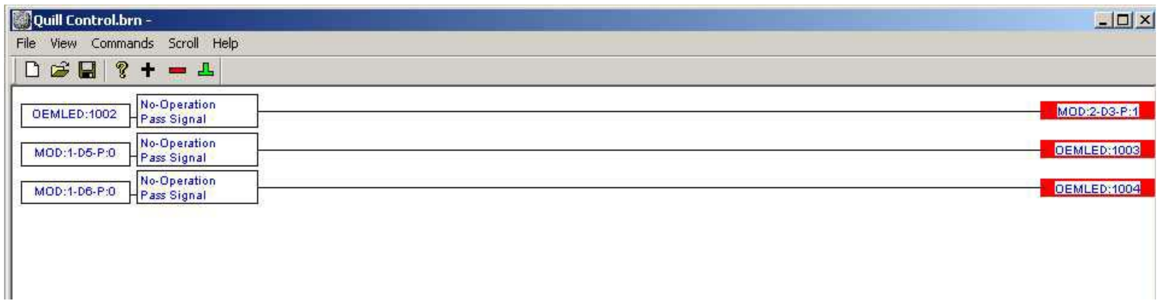
Quill Control:


M115.m1s - Mach3 VB Scipt Editor
File Edit Run Debug BreakPoints

```
' Advance Quill

SetUserLed(1002,1) ' Set UserLED to signal that
' quill should advance
```

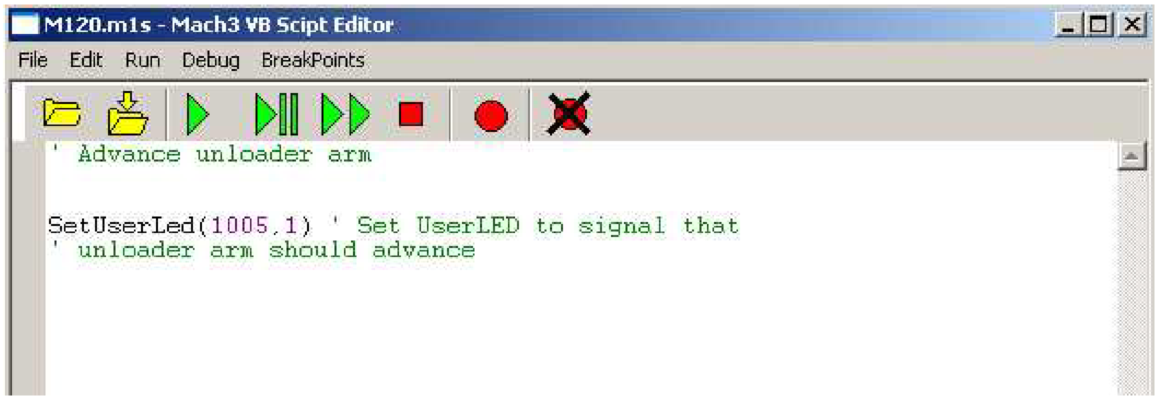

M116.m1s - Mach3 VB Scipt Editor
File Edit Run Debug BreakPoints

```
' Retract Quill

SetUserLed(1002,0) ' Set UserLED to signal that
' quill should retract
```

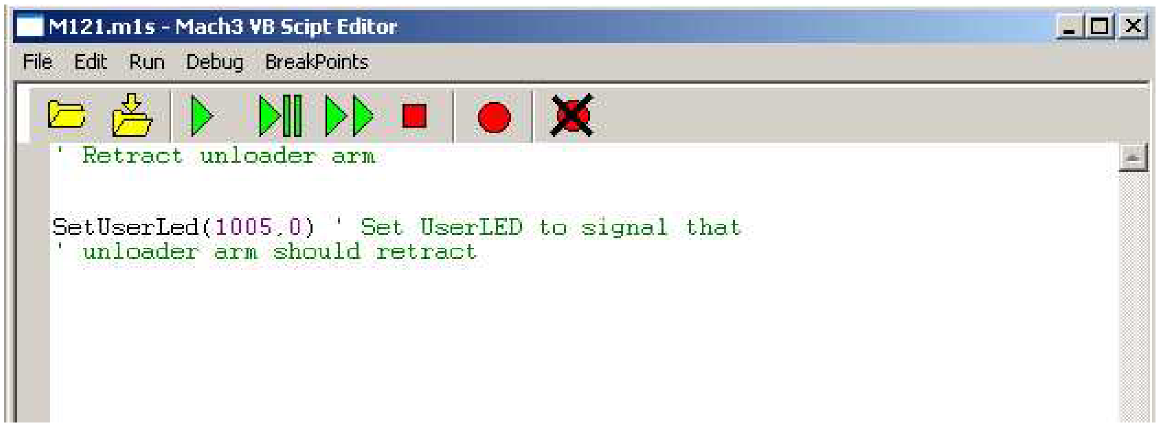


OEMLED:1002	No-Operation Pass Signal	MOD:2-D3-P:1
MOD:1-D5-P:0	No-Operation Pass Signal	OEMLED:1003
MOD:1-D6-P:0	No-Operation Pass Signal	OEMLED:1004

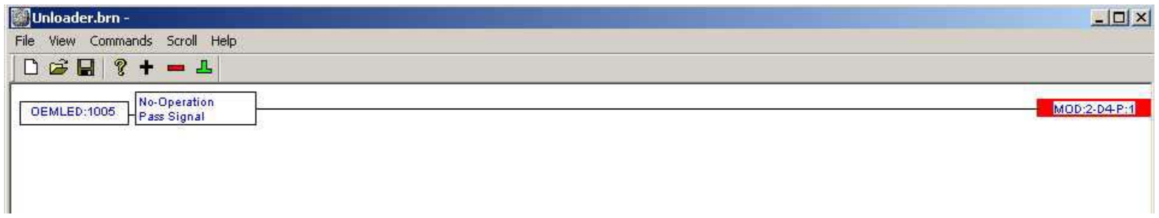
OEMLED:1002 controls the movement of the quill via word 2 (V1407), Bit 3
Bits D5 and D6 of V1401, monitor the quills status via OEMLEDs: 1003 and 1004
OEMLED:1003 is on when quill is in place with pressure
OEMLED:1004 is on when quill is retracted


M120.m1s - Mach3 VB Scipt Editor
File Edit Run Debug BreakPoints

```
' Advance unloader arm


SetUserLed(1005,1) ' Set UserLED to signal that
' unloader arm should advance
```


M121.m1s - Mach3 VB Scipt Editor
File Edit Run Debug BreakPoints

```
' Retract unloader arm


SetUserLed(1005,0) ' Set UserLED to signal that
' unloader arm should retract
```



Control loader arm via OEMLED:1005 and write value to word 2 of config 1, V1407, bit 4

WARNING:
This machine implementation is still being developed. There are roughly 90 rungs of Ladder Logic that also control this lathe. By the time the development is done there will be about 200 rungs of logic in the PLC and probably twice as many brains and macros. This was posted as an example of how Modbus, Macros, and Brains can work with a PLC to provide a more complete solution. How you implement your own machine control is your responsibility. Make sure you implement an Estop solution that is hardwired, and overrides all software controlled machine actions in case you need to shut it down to avoid machine damage or personal injury.

Good Luck,

Dave, Cole Controls Inc.