

## Connecting a VFD to Mach4 through Modbus (RS485)

V2

June 2019

Bill Caldwell

[bill@caldwellfam.net](mailto:bill@caldwellfam.net)

Connecting a spindle controller (VFD) to Mach4 through the Modbus (RS-485 serial port) is not an easy task, since there is no formal Mach4 Modbus documentation yet, and no overall LUA script (that I could find) to support a VFD through Modbus.

This post is a set of step by step instructions that hopefully will help someone else get their VFD up and running quickly.

For reference I am using Windows 10, Mach4 V4.2.0.3804, and FDTI serial driver V2.12.16.0

The Huanyang VFD is popular on many Internet sites, however there are several models out there and most of them are not truly Modbus compatible. I chose the Huanyang GT series 2.2KW for my Spindle since it does support the Modbus protocol and interfaces with Mach4's Modbus correctly. Other VFDs can easily be controlled as long as they are Modbus compliant.

These steps will give you the necessary setup to do the following spindle commands:

1. M3 (CW spindle)
2. M4 (CCW spindle)
3. M5 (Stop spindle)
4. Sxxxxx (Set spindle speed)

In addition the spindle will be controllable through the front panel buttons and executing G code.

### Step 1

Setup your VFD to control your particular spindle using the front panel on the VFD. This will include max frequency, min frequency, max current, max voltage, ramp parameters, etc.

Refer to the VFD manual for the instructions on how to use the front panel controls on the VFD to modify these parameters. They will be saved across power on/off cycles, so this only needs to be done once. Please note once these settings are changed, you will no longer be able to control the motor from the VFD front panel.

Once you can turn the spindle on and off and set the speed from the front panel of the VFD you are ready to modify the VFD parameters necessary to enable the Modbus communications port on the VFD so commands and status can be sent and received between the computer and the VFD, disabling the front panel motor control.

The Huanyang GT VFD required three values to be modified from the default values.

1. P0.01 =2 (default is 1) Run command source is the RS485 port
2. P0.07=7 (default is 0) Frequency source is the RS485 port
3. PC.00=1 (default is 0) Set the Modbus node address to 1

**(6/29/2019 update )** Most GT series VFDs default to RTU mode 1start, 8 data, no parity (1N8) at 19.2K BPS. However, I have been told that some do not, so make sure to check all these parameters in the VFD and correct them as necessary.

The VFD is now set to receive commands and send status to/from the serial Modbus (RS-485 port) at 19.2K baud, no parity in RTU mode.

Please note that computers do not implement RS-485 directly. You must purchase a USB to RS-485 converter. All RS-485 data will be sent out the selected USB port on the computer through the converter, across the RS-485 cable to the VFD. The RS-485 protocol implements a differential electrical connection that provides a great deal of common mode noise immunity, which is useful in the CNC environment.

Connect the VFD to the RS485 converter with the differential RS485 signals connected D+ to D+ and D- to D- on the VFD. If the signals are crossed, no communications will occur.

Plug in the USB cable from the RS485 convertor to the computer.

The Huanyang GT VFD uses the following register addresses and command bits. In the following steps you will be referencing those addresses.

Command register address 4096 decimal

Status register address 4097 decimal

Speed register 8192 decimal

The command register is set to:

Run forward 1

Run reverse 2

Stop 5

The Status register returns:

Forward 1

Reverse 2

Standby 3

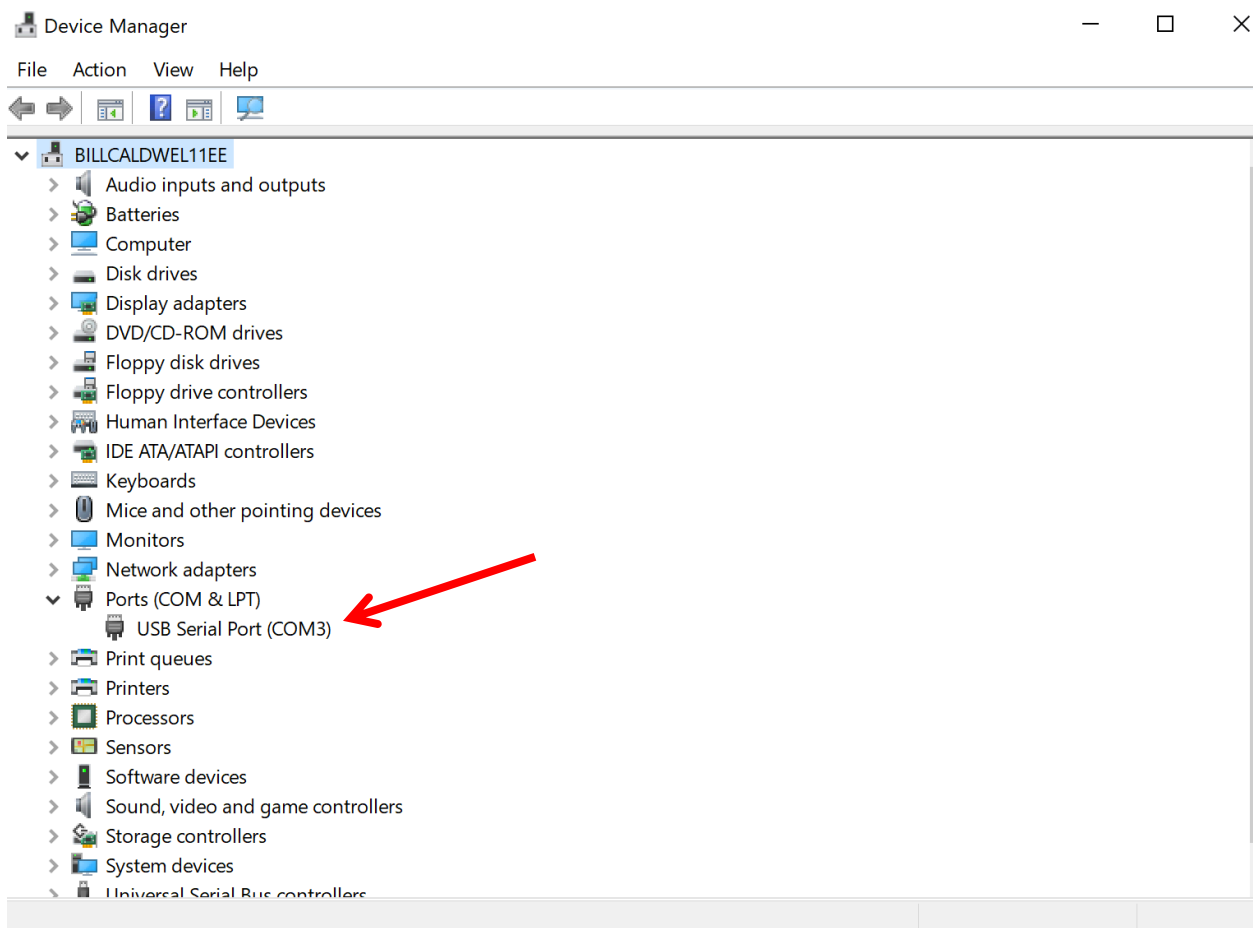
Fault 4

Power going off 5

If you use a different VFD most likely these addresses and values will need to be changed in the Modbus functions and the LUA code.

## Step 2

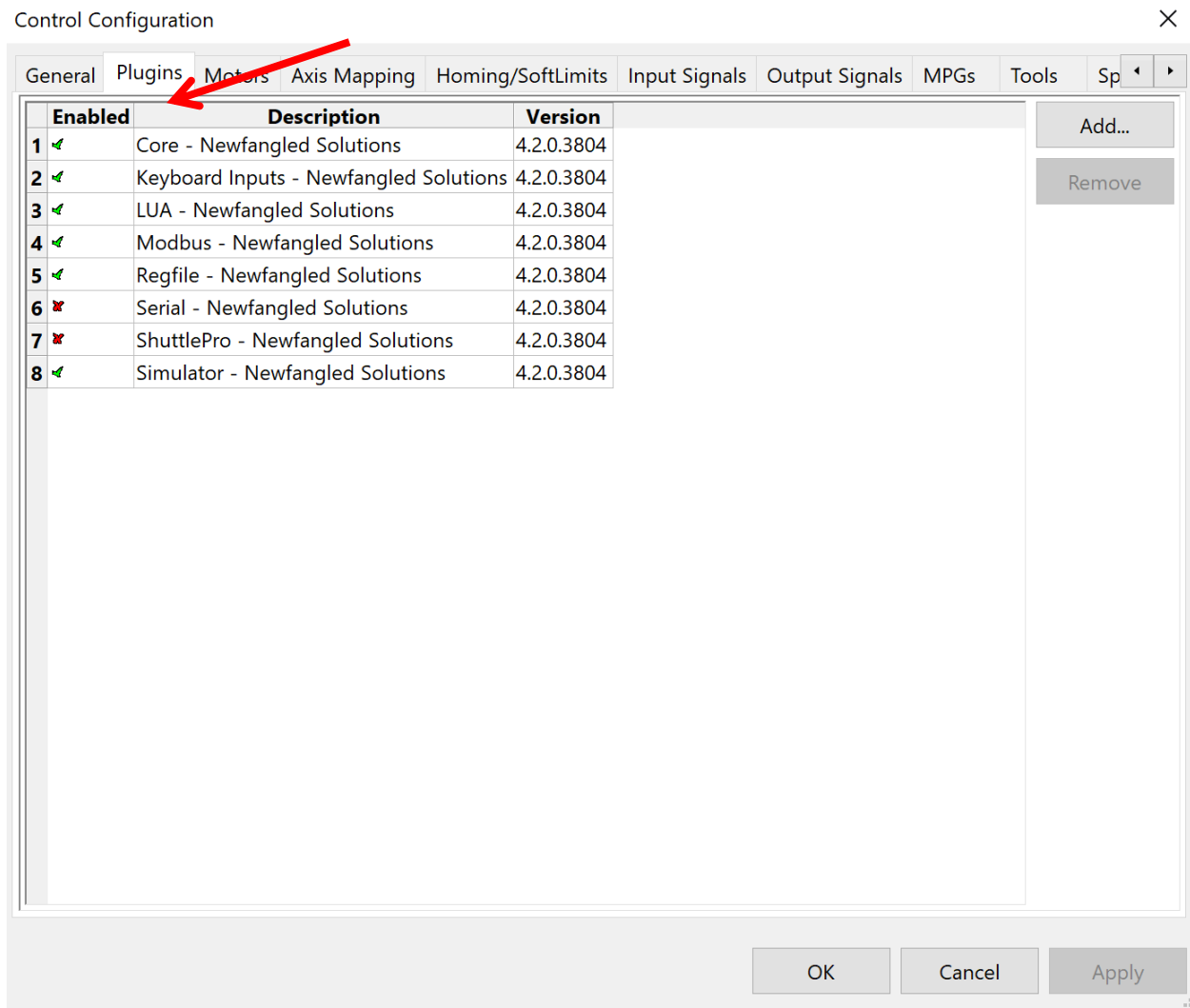
It is necessary to install the USB to RS-485 windows driver needed for the convertor. Follow the instructions supplied with your convertor to install the driver. Once the driver is installed, note the com port number it is associated with (Com3 in this case).



### Step 3

This step sets up the Mach4 Modbus controller to send commands to the selected VFD and receive status back from the selected VFD. Once these changes are made they will be permanently stored in the Mach4 profile you are currently running.

In the configuration / control menu, select the Plugins tab and enable the following plugins.



Select the spindle tab and fill in the Max RPM as shown below.

Control Configuration ✕

Plugins Motors Axis Mapping Homing/SoftLimits Input Signals Output Signals MPGs Tools Spindle To ◀ ▶

	MinRPM	MaxRPM	Accel Time	Decel Time	FeedBack Ratio	Reversed
0	0.00	24000.00	0.00	0.00	1.00000	✗
1	0.00	0.00	0.00	0.00	1.00000	✗
2	0.00	0.00	0.00	0.00	1.00000	✗
3	0.00	0.00	0.00	0.00	1.00000	✗
4	0.00	0.00	0.00	0.00	1.00000	✗
5	0.00	0.00	0.00	0.00	1.00000	✗
6	0.00	0.00	0.00	0.00	1.00000	✗
7	0.00	0.00	0.00	0.00	1.00000	✗
8	0.00	0.00	0.00	0.00	1.00000	✗
9	0.00	0.00	0.00	0.00	1.00000	✗
10	0.00	0.00	0.00	0.00	1.00000	✗
11	0.00	0.00	0.00	0.00	1.00000	✗
12	0.00	0.00	0.00	0.00	1.00000	✗
13	0.00	0.00	0.00	0.00	1.00000	✗
14	0.00	0.00	0.00	0.00	1.00000	✗
15	0.00	0.00	0.00	0.00	1.00000	✗
16	0.00	0.00	0.00	0.00	1.00000	✗
17	0.00	0.00	0.00	0.00	1.00000	✗
18	0.00	0.00	0.00	0.00	1.00000	✗
19	0.00	0.00	0.00	0.00	1.00000	✗

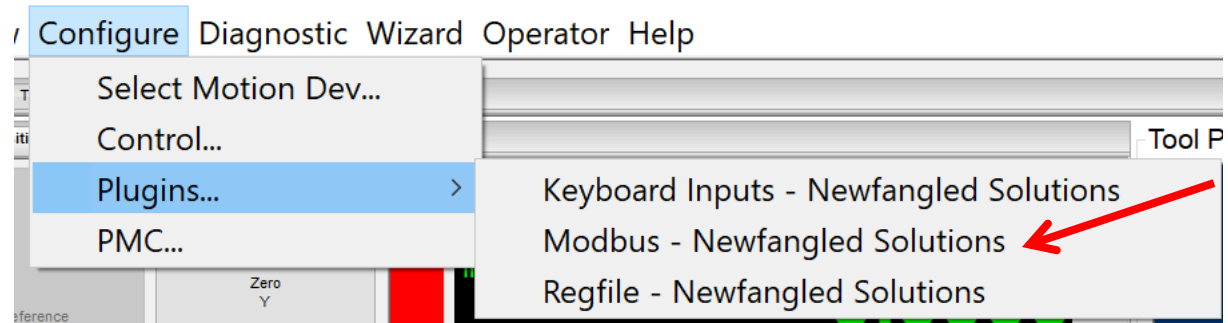
Max Spindle Motor RPM: 24000.00 ☐ Wait on spindle to stabilize to 9 percent.

Spindle Override Delay: 25 (ms)

Step/Dir Spindle Axis: None (Axis must be enabled and mapped.) ☐ Enable Step/Dir Spindle rigid tapping.

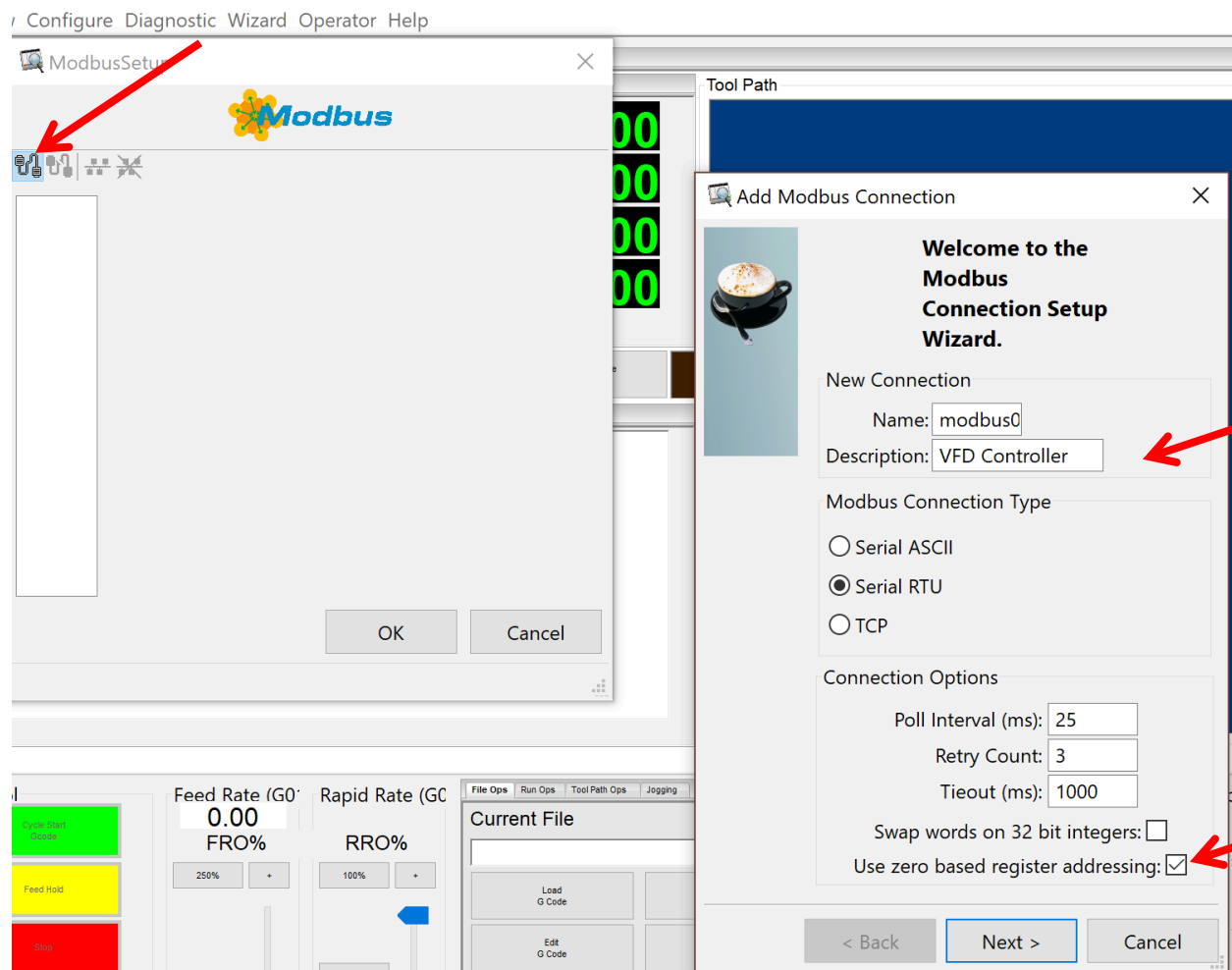
OK Cancel Apply

In the Configure / Control menu, select the Plugins pull down menu then select the Modbus plugin.

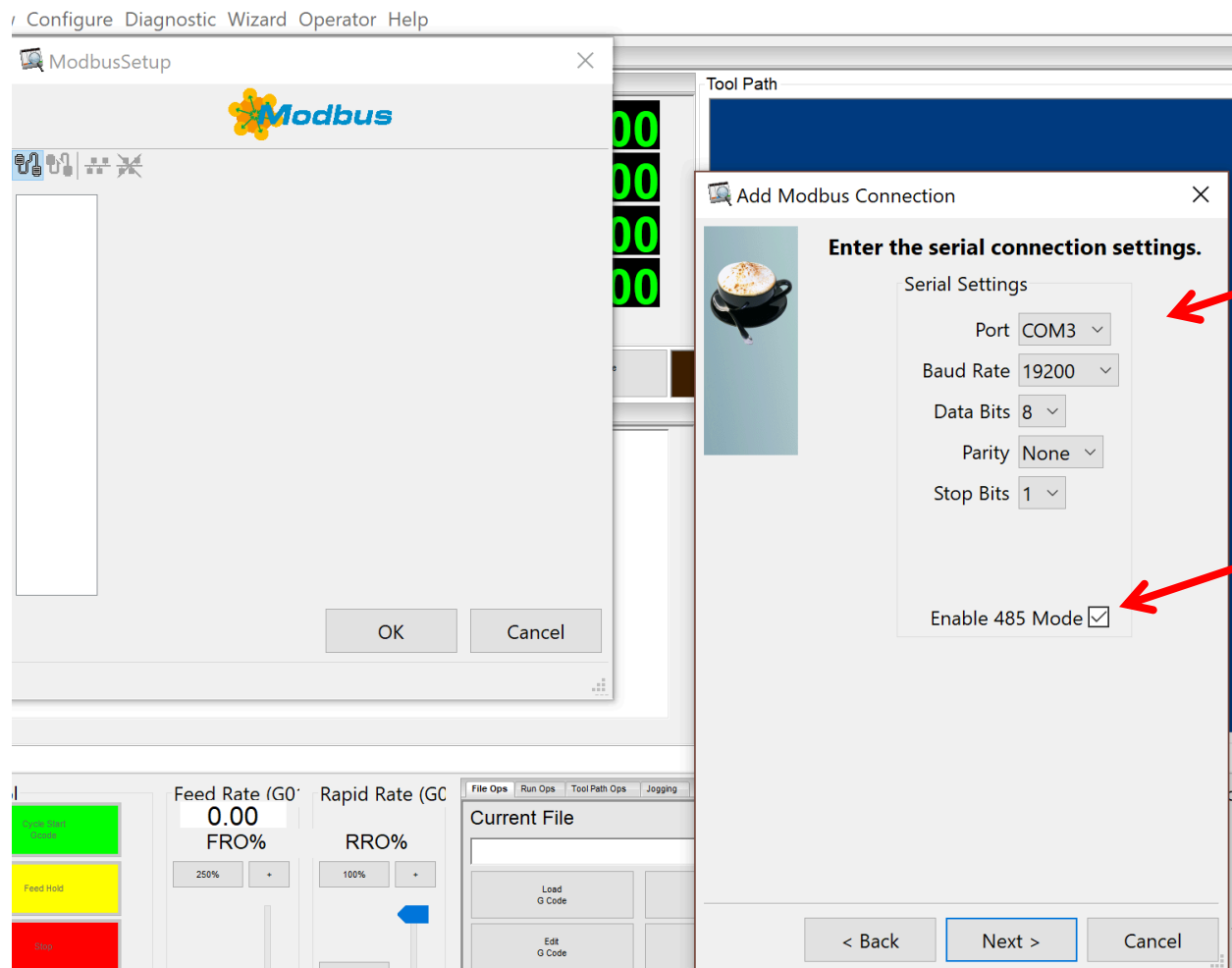


Begin creating the communications path from the Com port to the Mach4 Modbus controller. Fill in the box exactly as seen in the following screen.

This creates an instance of the Modbus controller called modbus0 set to RTU mode. Generally the default values will work. Make sure to check the zero based register addressing mode box.

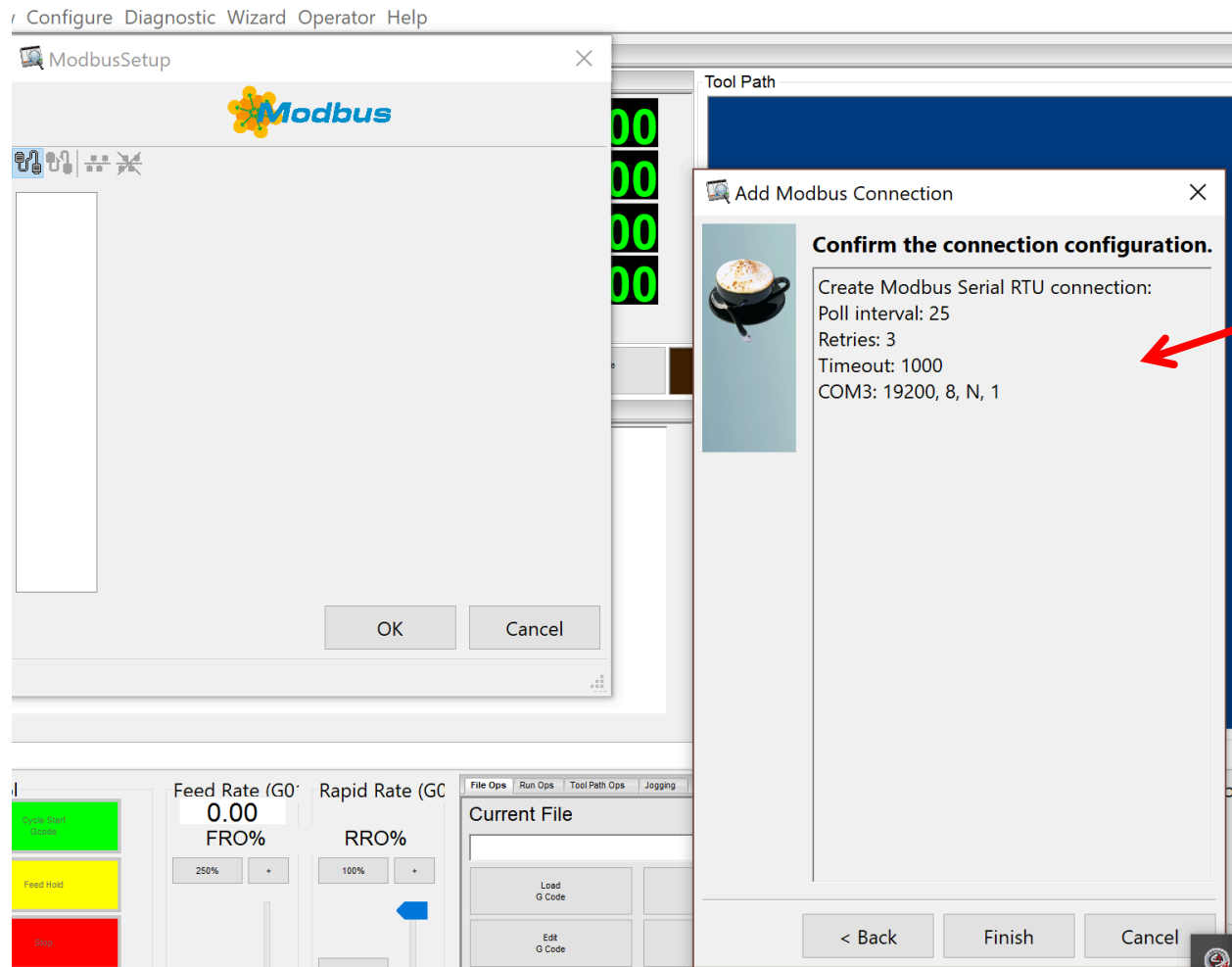


Select the port to correspond to the port attached to the USB to RS484 converter shown in the Windows device manger window (Com3 in this example). Continue setting the parameters as shown below.






When you are done with this step you will see the confirmation window as seen below.  
Your com port may be different than shown in this example.



You will now see the Modbus controller, just created, in the left column and a summary of the Modbus settings just selected on the right. Select OK.

Configure Diagnostic Wizard Operator Help

ModbusSetup



modbus0

Desc: VFD Cont Connection Type: Serial RTU Poll Interval (ms): 25

Retry Count: 3 Timeout: 1000

☐ Swap words on 32 bit integers. ☒ Use zero based register addressing.

Serial Settings

Port COM3

Baud Rate 19200

Data Bits 8

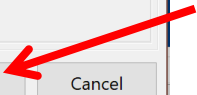

Parity None

Stop Bits 1

Enable 485 Mode ☒

OK Cancel

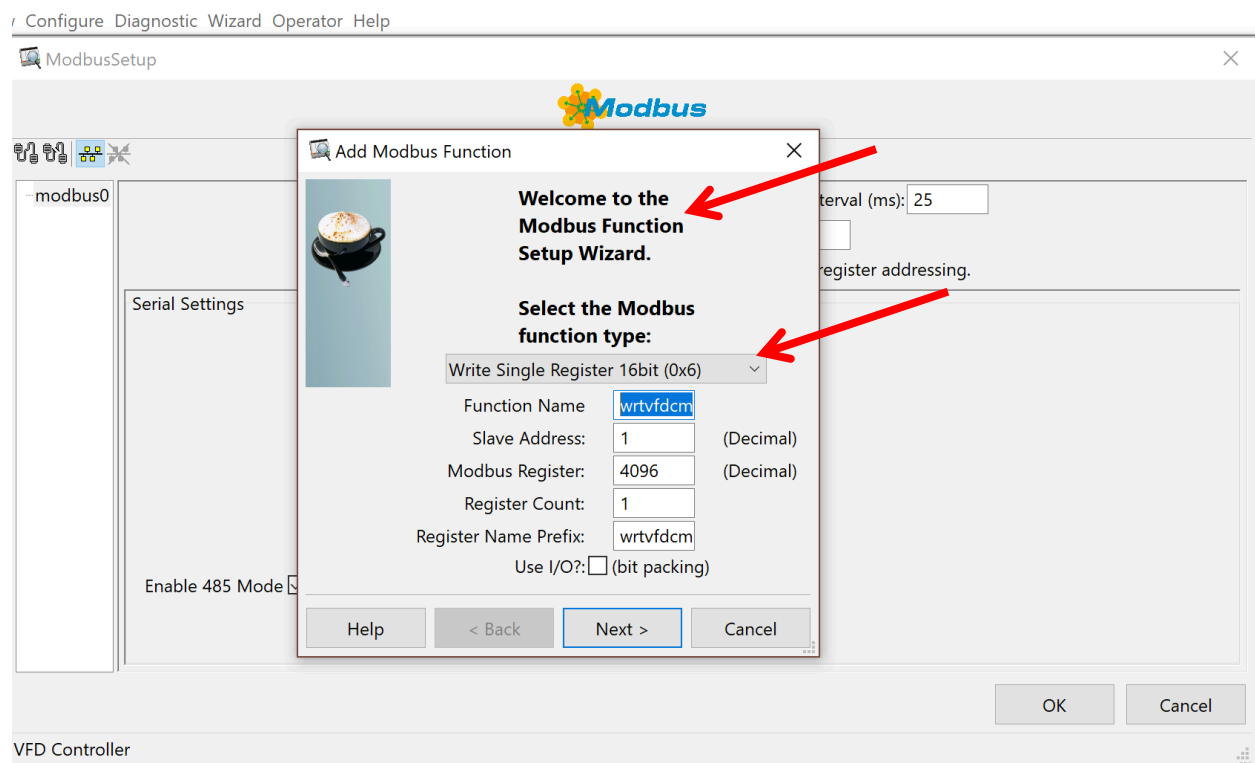
VFD Controller



## Step 4

The next steps create three necessary Modbus functions to write to the VFD command register, write to the VFD speed register, and read the VFD status register. Please make sure the function names are entered exactly as shown. They need to match the corresponding function names contained in the LUA script to follow.


Highlight the Modbus controller and select the create function button. From the pull down command list, select (0x06). Fill in the various parameters exactly as shown below. The name of this function is wrtvfdcmd (typed in 2 places). Press next.



The next screen confirms your settings for the first function. Press OK

Configure Diagnostic Wizard Operator Help

ModbusSetup



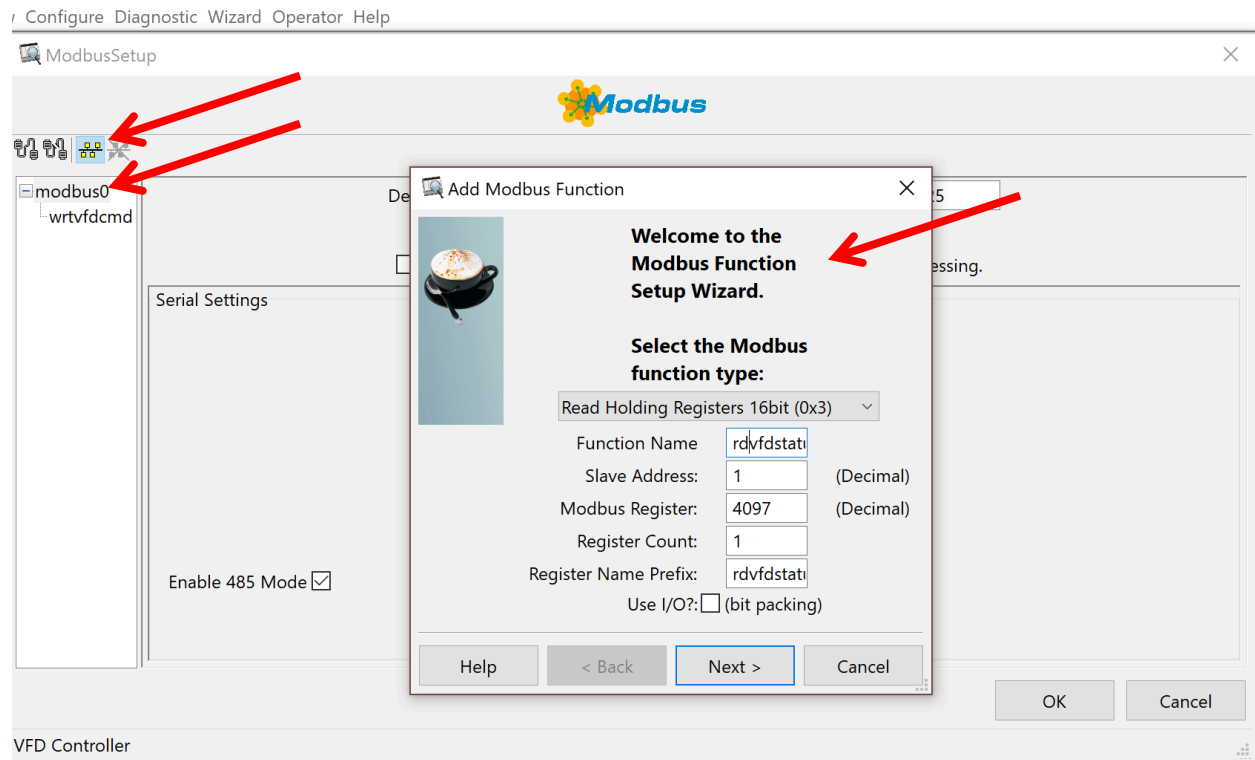
Modbus Function: Write Single Register 16bit (0x6) Slave Address: 1  
Modbus Register: 4096 Nbr Regs: 1

	Type	Name	Description
Reg 1	Mach Output Register	wrtvfdcmd0	wrtvfdcmd0

OK Cancel

wrtvfdcmd


Highlight the Modbus controller and select the create function button. From the pull down command list, select (0x03). Fill in the various parameters exactly as shown below. The name of this function is rdvfdstatus (typed in 2 places). Press next.




The next screen confirms your settings for the second function. Press OK

Configure Diagnostic Wizard Operator Help

ModbusSetup





modbus0  
wrtvfdcmd  
rdvfdstatus

Modbus Function: Read Holding Registers 16bit (0x3)

Slave Address: 1

Modbus Register: 4097

Nbr Regs: 1

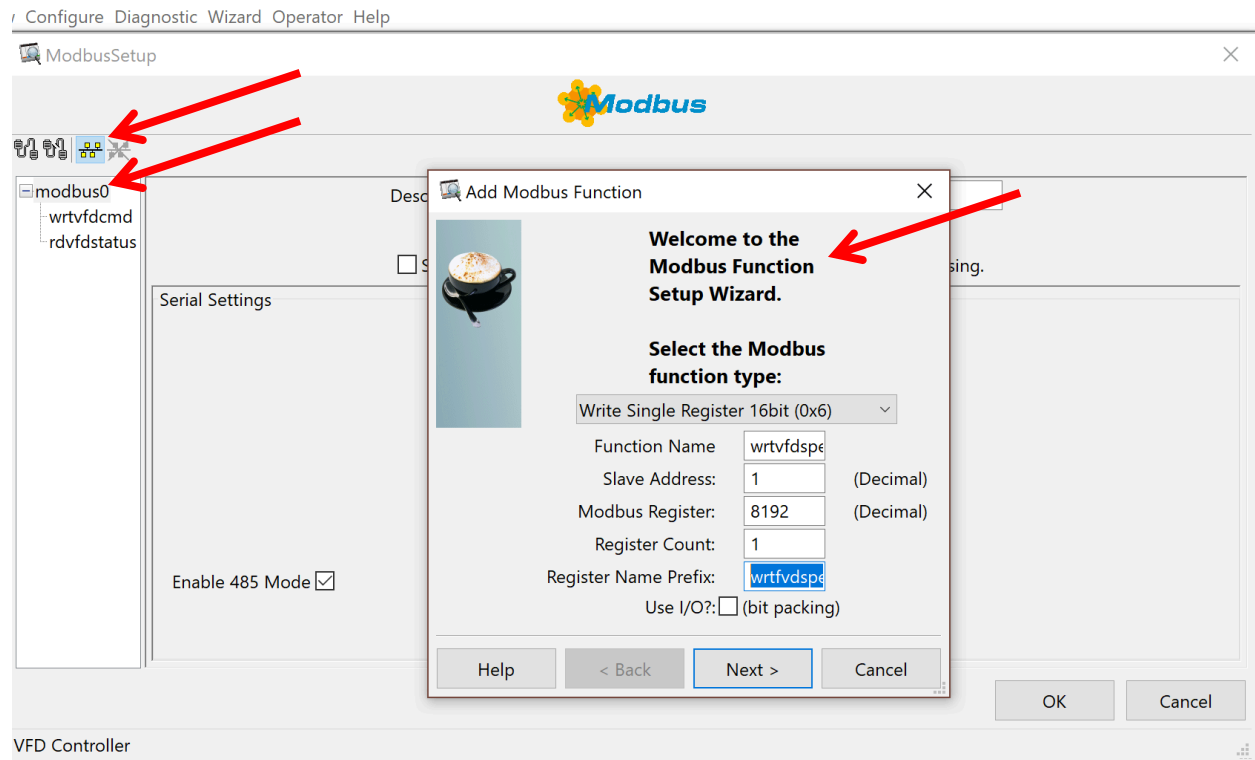
	Type	Name	Description
Reg 1	Mach Input Register	rdvfdstatus0	rdvfdstatus0

< >

OK Cancel

rdvfdstatus


Highlight the Modbus controller and select the create function button. From the pull down command list, select (0x06). Fill in the various parameters exactly as shown below. The name of this function is wrtvfdspeed (typed in 2 places). Press next.




The next screen confirms your settings for the third function. Press OK

Configure Diagnostic Wizard Operator Help

ModbusSetup





modbus0

wrtvfdcmd

rdvfdstatus

wrtvfdspeed

Modbus Function: Write Single Register 16bit (0x6)

Slave Address: 1

Modbus Register: 8192

Nbr Regs: 1

	Type	Name	Description
Reg 1	Mach Output Register	wrtvfdspeed0	wrtvfdspeed0

<

>

OK

Cancel


wrtvfdspeed



This is the final confirmation screen. If everything looks as it should, press OK

Configure Diagnostic Wizard Operator Help

ModbusSetup



modbus0  
wrtvfdcmd  
rdvfdstatus  
wrtvfdspeed

Desc: VFD Cont Connection Type: Serial RTU Poll Interval (ms): 25  
Retry Count: 3 Timeout: 1000  
☐ Swap words on 32 bit integers. ☒ Use zero based register addressing.

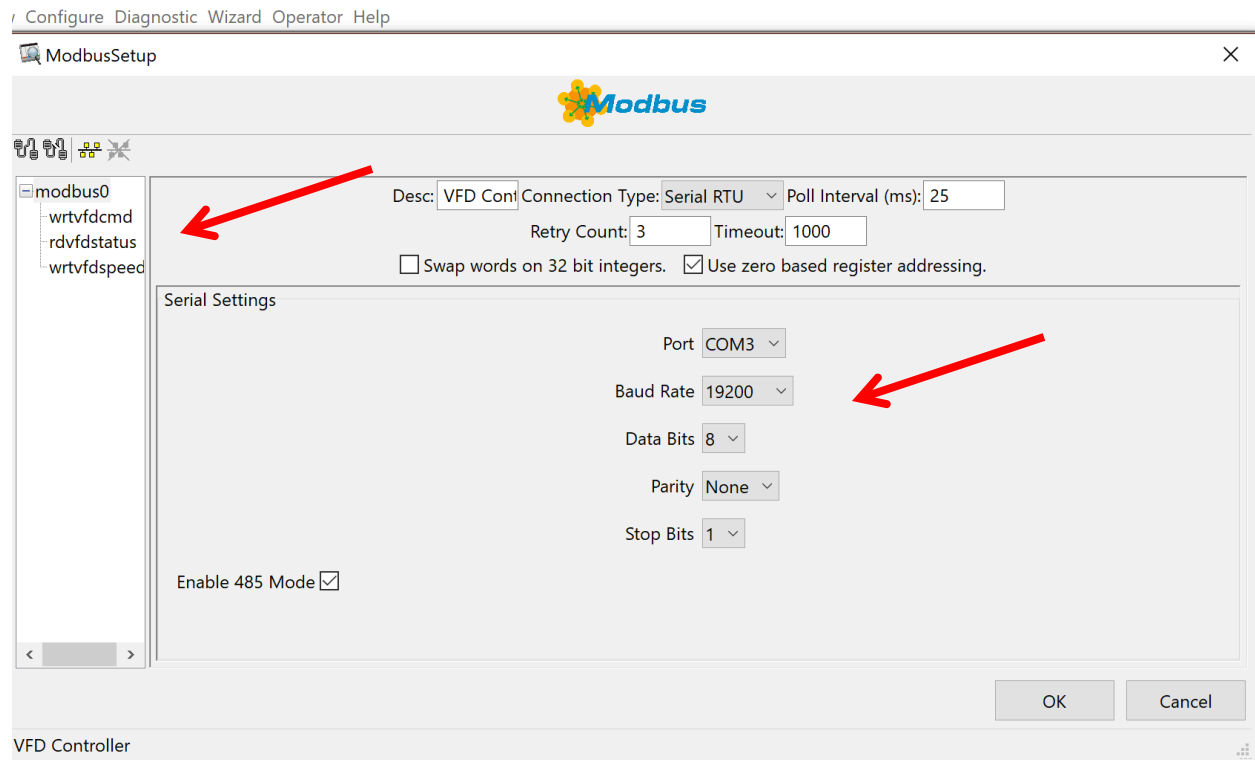
Serial Settings

Port COM3  
Baud Rate 19200  
Data Bits 8  
Parity None  
Stop Bits 1

Enable 485 Mode ☒

OK Cancel

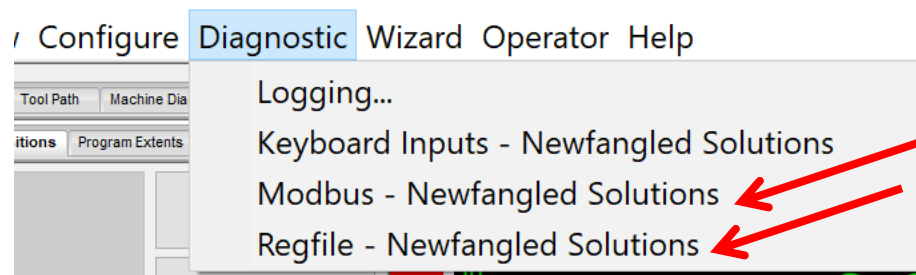
VFD Controller



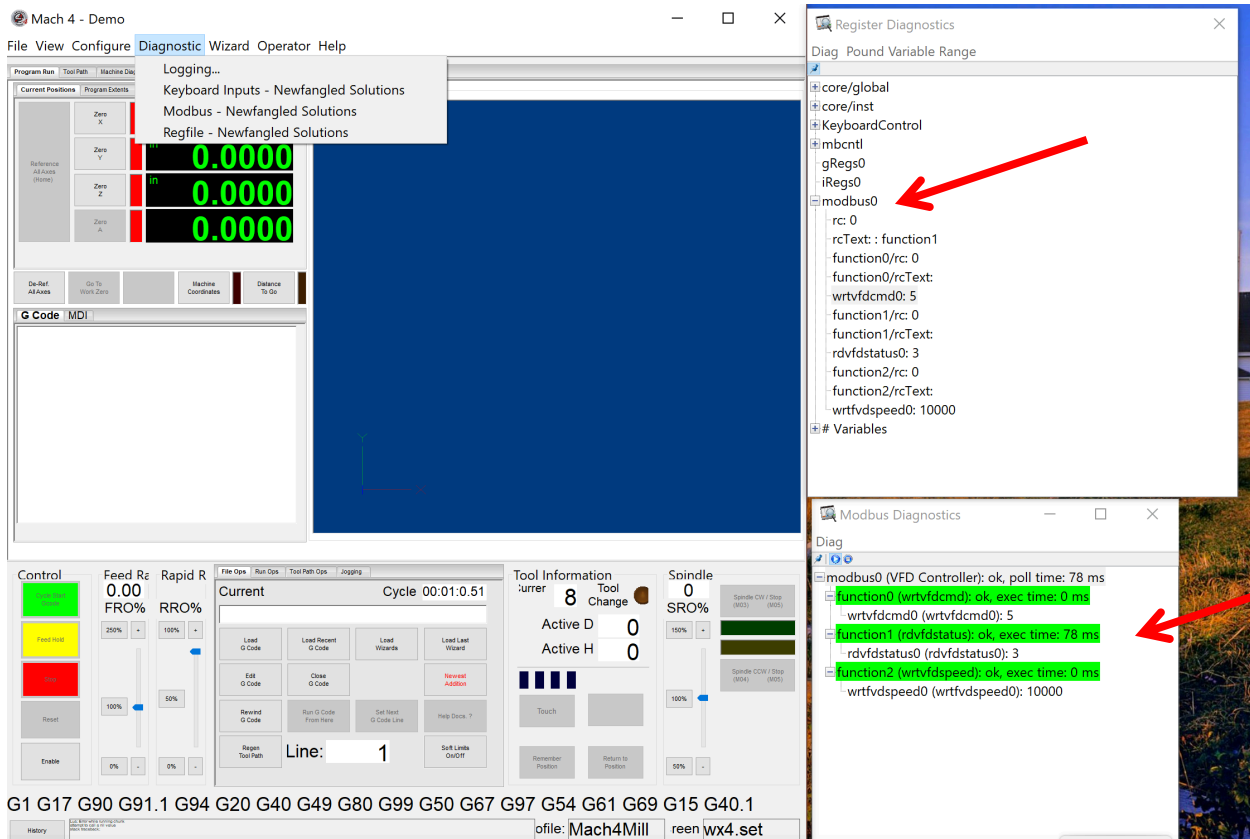
## Step 5

Set up the diagnostics windows to test the Modbus to VFD connection.

From the diagnostics menu select Modbus and Regfile .



This will open two windows. The Modbus window, once expanded, will show you the three functions you just created and their current status along with some timing information. The Regfile window, once expanded, will show you the current Modbus register contents for the command, status, and speed registers for the VFD. The status register is reading the VFD status register based on the poll rate set in the Modbus setup. The write registers contain the values set by Mach4 to be sent to the VFD. Once a value is put into either write register, it will be sent to the VFD based on the Poll rate.



Double click on the Modbus speed register and enter the value 10000. This represents 100.00 % of the maximum speed set in the VFD previously (max speed is 400Hz).  $400\text{Hz} * 60 \text{ sec} = 24,000 \text{ RPM}$ .

The screenshot displays the Mach4 Mill software interface. The main window shows the 'Program Run' tab with a 'Tool Path' visualization. A red arrow points to the 'Edit Register' dialog box, which is open over the 'Register Diagnostics' window. The 'Edit Register' dialog shows the 'Value' field set to 10000. The 'Register Diagnostics' window lists various registers, including 'wrtvfdcmd0', 'rdvfdstatus0', and 'wrtvfdspeed0'. The 'Modbus Diagnostics' window shows the status of the Modbus communication, including 'function0 (wrtvfdcmd): ok, exec time: 0 ms' and 'function2 (wrtvfdspeed): ok, exec time: 0 ms'.

**Register Diagnostics**

Diag Pound Variable Range

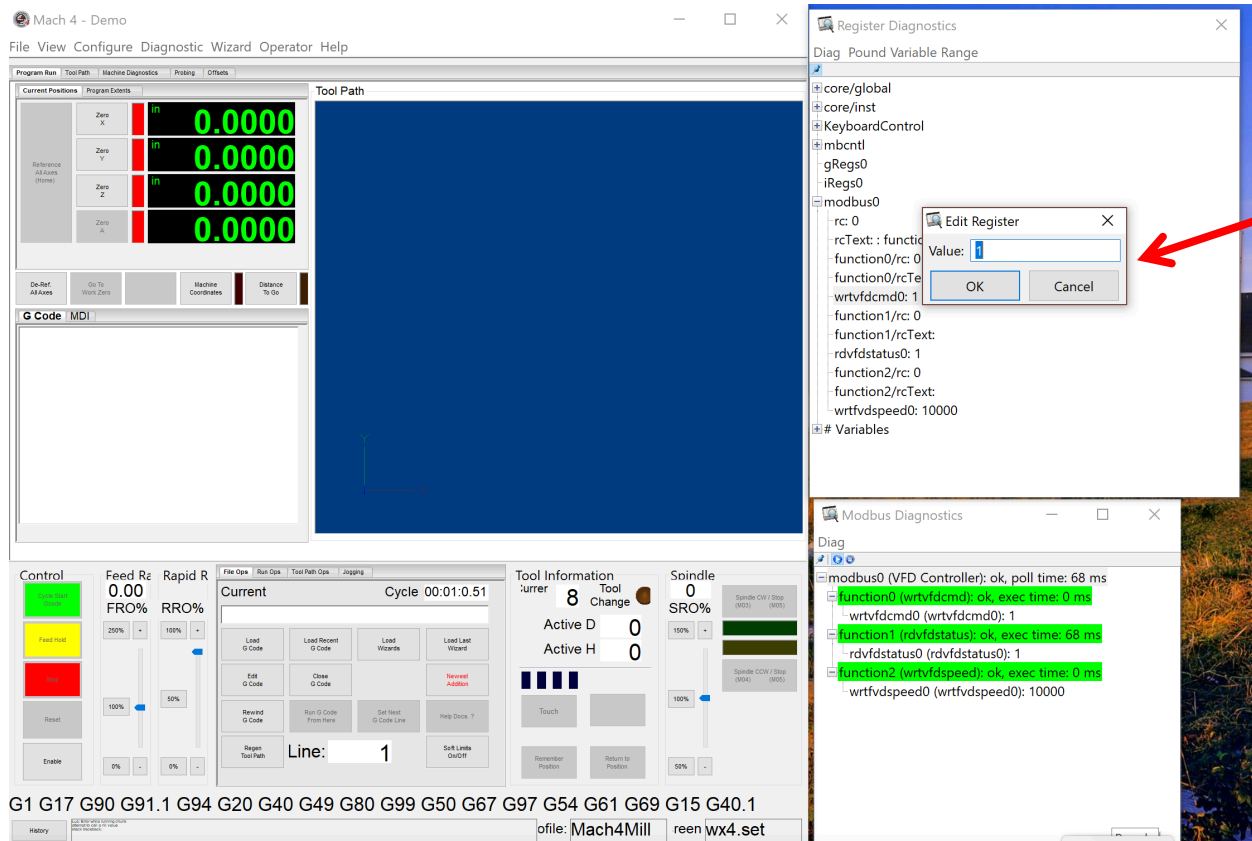
- core/global
- core/inst
- KeyboardControl
- mbcctl
- gRegs0
- iRegs0
- modbus0
  - rc: 0
  - rcText: : function0/rc: 0
  - function0/rc: 0
  - wrtvfdcmd0: 1
  - function1/rc: 0
  - function1/rcText: rdvfdstatus0: 1
  - function2/rc: 0
  - function2/rcText: wrtfdspeed0: 10000
- # Variables

**Modbus Diagnostics**

Diag

- modbus0 (VFD Controller): ok, poll time: 63 ms
- function0 (wrtvfdcmd): ok, exec time: 0 ms
- wrtvfdcmd0 (wrtvfdcmd0): 1
- function1 (rdvfdstatus): ok, exec time: 63 ms
- rdvfdstatus0 (rdvfdstatus0): 1
- function2 (wrtvfdspeed): ok, exec time: 0 ms
- wrtvfdspeed0 (wrtvfdspeed0): 10000

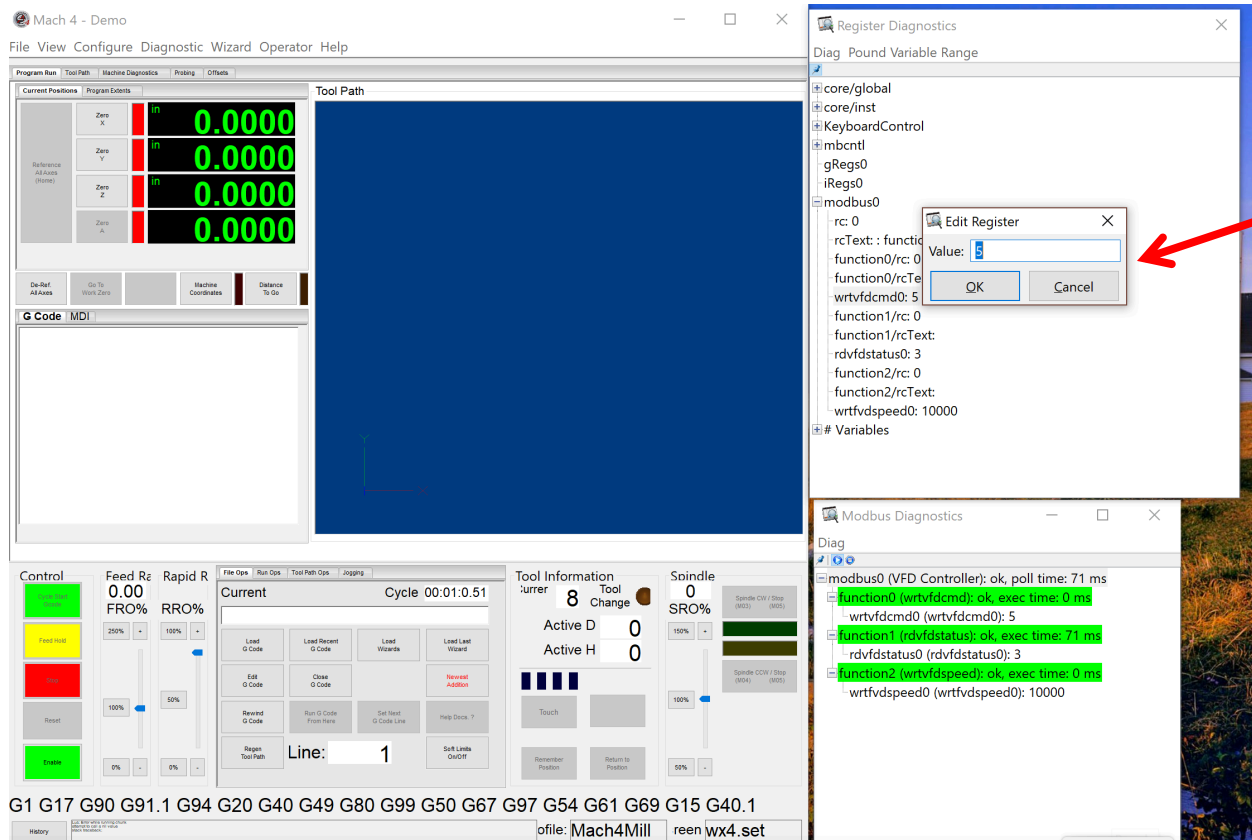
Double click on the command register and enter the value 1. The spindle should run clockwise at full speed.



Enter the value 5 to stop the spindle.

Enter the value 2 to reverse the spindle.

At this point you should be able to set the speed to any value between 0 and 10000 and run the spindle in either direction.



## Step 6

Download the LUA script from the following location:

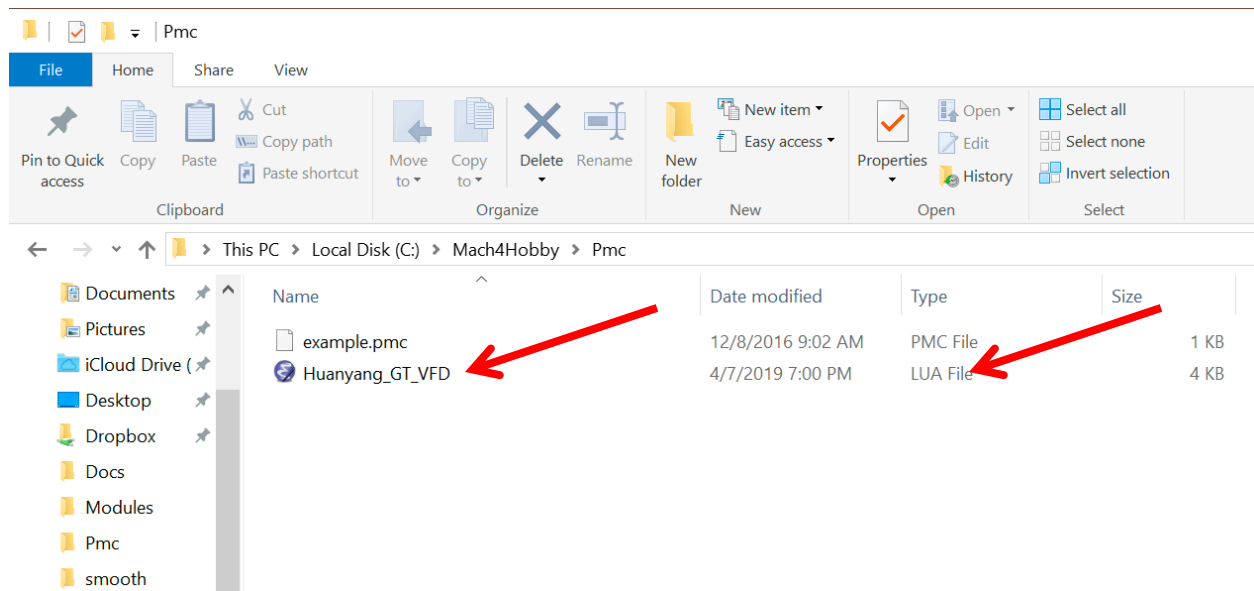
[https://code.porkrind.org/david/mach4-vfd/blob/master/Huanyang\\_GT\\_VFD.lua](https://code.porkrind.org/david/mach4-vfd/blob/master/Huanyang_GT_VFD.lua)

This code was originally written with the PMC editor, but was modified by my son, David to add the necessary logic and values that I couldn't get the PMC editor to do.

Place the downloaded code in the PMC folder inside the Mach4 main folder as shown below.

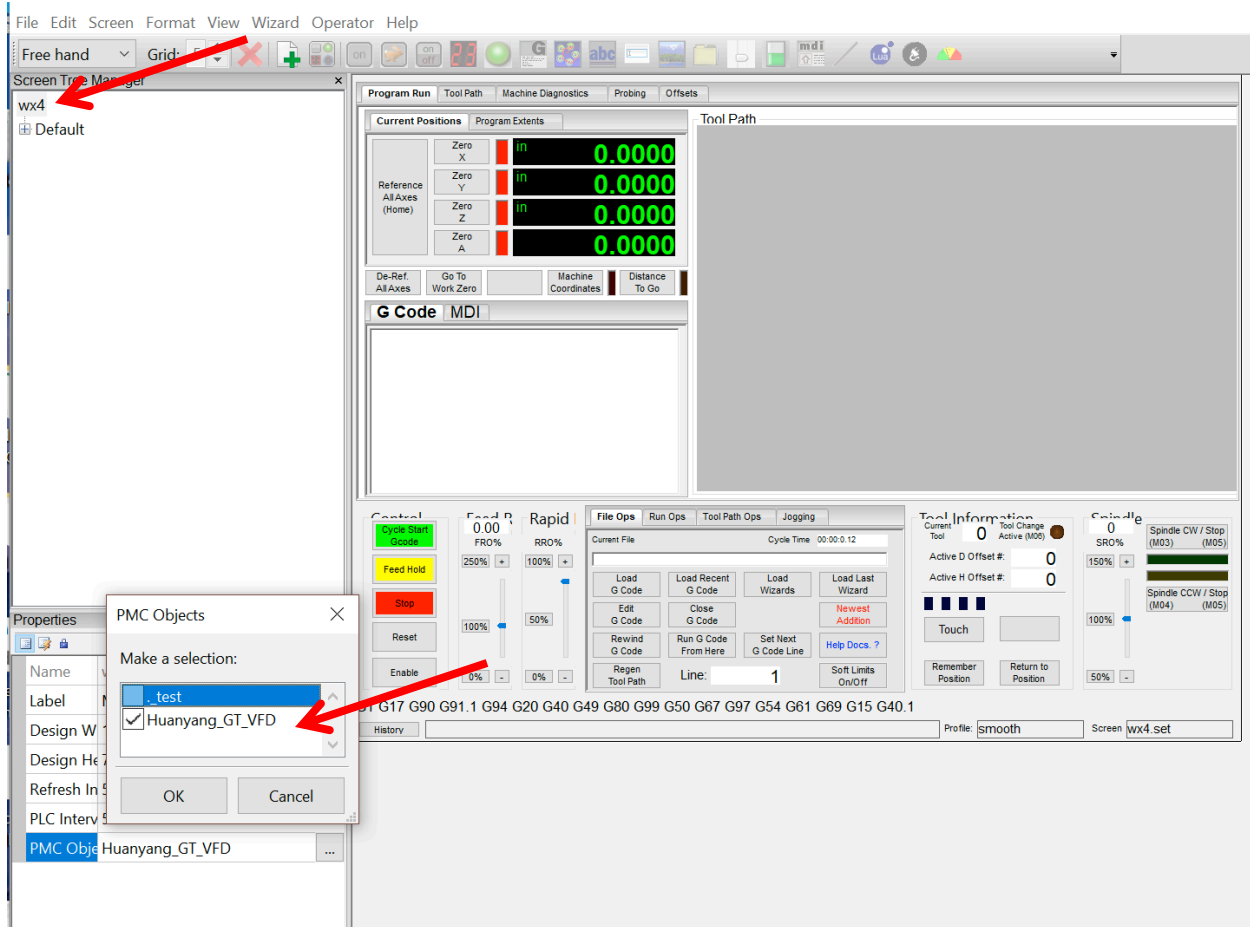
Note the file type and make sure it is a LUA file and not a text file. Sometimes things get altered when downloading to different machines.

This code will need to be slightly modified if the command bits in your VFD are different than the Huanyang GT.



Restart Mach4 and select the screen editor in the operator menu.

Select the top-level screen in the left hand column. Go to the bottom left and locate the PMC line. Click on the 3 dots on the right. Select the Huanyang\_GT\_VFD. Be sure to check the box next to the name. Exit the screen editor by clicking its button in the operator menu.





You can now control the spindle from either the buttons or with the M commands through the MDI interface. The MDI interface now requires that the start button be pressed to enter the commands. The speed can only be entered through the S command through the MDI interface.

All supported commands execute from G code.

Questions, Comments, and improvements are welcome. (bill@caldwellfam.net)

Have Fun with your new spindle controller!

File View Configure Diagnostic Wizard Operator Help

The screenshot displays the Mach4Mill software interface. At the top, there are tabs for Program Run, Tool Path, Machine Diagnostics, Plotting, and Offsets. Below these, the 'Current Positions' section shows coordinates for Z-axis (Z0, Z1, Z2, Z3) and X-axis (X0, X1, X2, X3), all set to 0.0000. The 'MDI' (Manual Data Input) section is active, showing the command 'S12000' entered. A red arrow points to this command. The 'Tool Path' section is empty. The bottom section contains various controls: 'Control' (Start, Stop, Reset, Pause), 'Feed Rate (G0) FRO%' (0.00), 'Rapid Rate (G1) RRO%' (50%), 'Current File' (G0 G17 G90 G91.1 G94 G20 G40 G49 G80 G98 G50 G67 G97 G54 G64 G69 G15 G40.1), 'Tool Information' (Current Tool 8, Tool Change Active (M06), Active D Offset #: 0, Active H Offset #: 0), and 'Spindle' (12000 SRO%, Spindle CW / Stop (M03), Spindle CCLW / Stop (M04), Spindle On/Off). A red arrow points to the 'Spindle On/Off' button. The bottom status bar shows the profile 'Mach4Mill' and screen 'wx4.set'.

