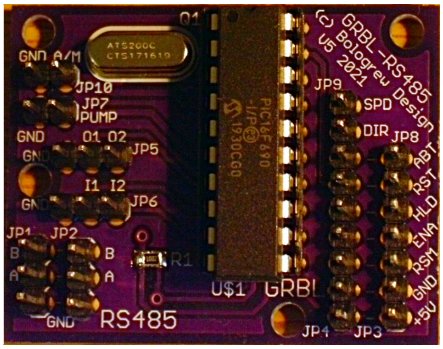
GRBL-RS485 VFD Controller - Huanyang Version

Manual Version 2 Hardware Version 5 Firmware Version 3

Description

This module connects a GRBL CNC controller to a Huanyang Variable Frequency Drive to provide direction and speed control of a water or air-cooled spindle via an RS485 link.
Once configured, the module works standalone as the RS485 bus master providing open or closed loop control of the VFD. The module can be switched temporarily into slave mode for configuration or just to disable it.
While the spindle is activated, a pump header pin goes high which can be used to control a cooling pump for water-cooled spindles.

Connections



Power

This module requires +5V and GND to operate. It can be fed into JP3 or JP4 as the dash on the circuit board indicates that both pins are connected. The GND connection should be common with that of the GRBL controller. Typically this module requires around 15mA.

GRBL Direction and Speed

The GRBL Direction and Speed (drive output) signals must be connected to JP9 DIR and SPD. If the module is located some distance from the GRBL controller, consider using shielding cable for the SPD line. Ideally, all connections should be kept as short as possible. The GRBL Speed signal may also be known as SPN.EN and is a Pulse Width Modulated signal.

GRBL External Control Lines

The GRBL Abort, Hold and Resume signals must be connected to the ABT, HLD and RSM pins on either JP8 or JP9 with the dash on the circuit board indicating that both pins are connected. You should also connect the Reset and Enable(if available) signals to the RST and ENA pins. The Reset signal is connected to the microcontroller but is not used at this time. The Enable signal is only connected across JP8 and JP9 to make it easy to daisy chain that signal to an external control box.

RS485

Two RS485 headers are provided with JP1 and JP2. One should be connected to the VFD. VFDs are incredibly electrically noisy and must therefore be isolated from this module by a bus repeater or preferably a bus isolator/repeater. The other header can be used to daisy chain to another RS485 interface such as a USB to RS485 computer interface which can be used to configure the module and later optionally monitor the bus. If only one header is used then a 120Ω resistor should be fitted across “A” and “B” to terminate the bus.

Pump and User Output Lines

The module provides an output on JP7 for pump control. This output is high (+5V) whenever the spindle is activated. This line cannot provide much current so it should only be used with an active relay module or as an input into the gate of a MOSFET.
This module also provides two user output lines, O1 and O2 on JP5 which have the same driving characteristics as the JP7 pump control. The user output lines are not used directly by the module but can be controlled by another host on the bus. The supplied software shows the protocol of how this can be accomplished given that this module is normally the bus master and all other devices on the bus may only speak when spoken to.

User Input Lines

This module provides two user input lines, I1 and I2 on JP6. These lines are actually analogue to digital inputs and have a resolution of 10 bits. The input voltage range to both lines is 0V to +5V. As an example, you could use such an input to measure coolant temperature.

These inputs could also be used as digital inputs by using a switch and pull-up resistor. In fact, using a pull-up resistor chain and multiple switches, you could detect multiple inputs.

External Auto/Manual switch

This module provides an input, A/M on JP10 to switch between Auto and Manual modes. This facility has to be enabled in the module configuration to work. When enabled, the module will be in Auto mode when A/M is open and Manual mode when connected to ground.

GRBL External Control Box

Whichever of the pins on headers JP8 and JP9 are unused should be used to daisy chain the GRBL external control lines to an external control box. If you don't have an external control box then it is definitely worth adding one. If you don't currently have one then, at a minimum, you must provide an external abort switch connected between ABT and GND. This is necessary as this line is used as a power on safety interlock to prevent unexpected spindle activation. The HLD line can also be used at power on in combination with the ABT line to switch the module into slave mode.

Notes on Variable Frequency Drives

This was mentioned earlier but is important and bears repeating - VFDs are incredibly electrically noisy. Your VFD should be positioned as far away as possible from your controller, ideally inside a metal cabinet. You should also connect it via a mains inlet filter to stop noise feeding back into the mains circuit. Proper four core shielded cable must be used to connect the spindle to the VFD. Make sure the earth connection pin of the spindle is actually connected to the body of the spindle and, if not, remove that connector and fit an earth wire. Consider fitting clip-on ferrites onto the cables to reduce noise.

After you have done all this and contained the noise of the VFD, connecting an RS485 cable to it provides an easy escape route for all that noise to pour straight into your controller leading to instability, crashes and even damage to circuitry. This module analyses the PWM speed signal from the GRBL controller and will be unable to do so if swamped with noise from the VFD. To prevent this, fit an RS485 repeater or preferably a bus isolator/repeater between the VFD and other equipment on the bus. Such an RS485 isolator device is also made by Bologrew Design.

VFD Configuration

Your VFD should already be configured to operate the spindle manually.

In order to use your VFD with this module, your VFD must now be configured to use RS485. The communication addresses and baud rate used can be altered in the module but are supplied with the defaults of the VFD having address 1 and the module having address 2 using a baud rate of 9600bps.

PD163:	Communication Address	Set to 1.
PD164:	Communication Baud Rate	Set to 1 (9600bps).
PD165:	Communication Data Method	Set to 3 (8N1 RTU).
PD001:	Source of Operation Commands	Set to 2 (set by communication port).
PD002:	Source of Operating Frequency	Set to 2 (set by communication port).

It is worth making sure that PD144 is set correctly. This will make sure the RPM display on the VFD correctly corresponds to the actual speed of the spindle. The value to enter is the RPM of the spindle when the output frequency to the spindle is 50Hz.

So, if your spindle runs at 24000rpm at the VFD's top output frequency of 400Hz:
PD144 = (24000 ÷ 400) * 50
PD144 = 3000

VFD Connection

The RS485 connections are marked as “RS+” and “RS-” on the terminal block. CAT 5 shielded cable can be used to connect the VFD to your RS485 isolator - use one twisted pair. If the VFD is the last device on the bus, fit a 120Ω resistor across these terminals to terminate the bus. There is some confusion between equipment and semiconductor manufacturers as to the naming convention of the RS485 signal lines with some using “+” and “-” and some using “A” and “B”. There is further confusion regarding whether “+” corresponds to “A” or “B”. I would start on the basis that “+” does correspond to “A” but be aware if you are unable to communicate that you might have to switch the connections over. Theoretically you should also connect ground to your RS485 isolator. I would suggest not connecting it unless you have communication problems.
Then using the same sort of cable, connect one twisted pair from your RS485 isolator to the “A” and “B” pins of either JP1 or JP2 on the module following the previously suggested recommendations.

PC Connection

CAT 5 cable can be used to connect the module to your PC's USB to RS485 converter. Connect one twisted pair to the “A” and “B” pins of either JP1 or JP2 and connect the other end to your USB to RS485 adapter following the same recommendations as for the “VFD Connection” above.
“PC” in this context can mean an x86 laptop, x86 desktop or a Raspberry Pi. This connection is only needed for configuration but can be used permanently to monitor the bus and the module and alter its operating state.

GRBL Configuration

Your GRBL controller must be configured to match the parameters of your spindle. Set parameter 30 to the maximum RPM speed of your spindle. Set parameter 31 to the minimum speed of your spindle.
Of course, you can set the maximum speed parameter to a lower value than your spindle maximum but you shouldn't set it to a higher value.
You can set the minimum speed parameter to zero but if you have an air-cooled spindle then they should not be run below a defined minimum speed as they will overheat.
Water-cooled spindles should not exhibit the same overheating but will have reduced torque at lower speeds so you may still want to set a non-zero minimum. If you intend to use a mechanical edge finder in your spindle you may want to set the minimum speed to that required for that tool – possibly 1000 to 2000rpm.

How the module operates depends on whether or not you set have a minimum speed of zero. The rpm calculated from the GRBL output level is set to the mid-point of the speed range for that level. If you have a non-zero minimum speed then the first output level is instead set to that minimum speed.
e.g.
You have a spindle with a maximum speed of 24000rpm.
You set GRBL parameter 30 to 24000.
You set GRBL parameter 31 to 0 for zero minimum speed.
The speed range is 24000 – 0 = 24000.
The GRBL controller drive output has 256 states: off, 254 PWM levels and fully on (static +5V or 100% PWM duty cycle).
The speed interval for each level is 24000 ÷ 254 = 94.488rpm. Half the interval is ~47rpm.
The first level starts at 1rpm and continues to 94rpm. This produces an output of ~47rpm.

The next level starts at 95rpm and continues to 188rpm. This produces an output of ~141rpm.
The next level starts at 189rpm and continues to 283rpm. This produces an output of ~235rpm.
The penultimate level will produce an output of ~23952rpm.
Fully on will produce an output of 24000rpm.

You have a spindle with a maximum speed of 24000rpm.
You set GRBL parameter 30 to 24000.
You set GRBL parameter 31 to 1000 for a minimum speed of 1000rpm.
The speed range is 24000 – 1000 = 23000.
The GRBL controller drive output has 256 states: off, 254 PWM levels and fully on (static +5V or 100% PWM duty cycle).
The speed interval for each level is 23000 ÷ 254 = 90.55rpm. Half the interval is ~45rpm.
The first level is overridden to produce the minimum speed of ~1000rpm.
The first level starts at 1rpm and continues to 1090rpm. This produces an output of ~1000rpm.
The next level starts at 1091rpm and continues to 1181rpm. This produces an output of ~1136rpm.
The next level starts at 1182rpm and continues to 1271rpm. This produces an output of ~1226rpm.
The next level starts at 1272rpm and continues to 1362rpm. This produces an output of ~1317rpm.
The penultimate level starts at 23819rpm and continues to 23999rpm. This produces an output of ~23954rpm.
Fully on will produce an output of 24000rpm.

Module Configuration 1
Download the configuration and monitoring software from here:

https://design.bologrew.net/GRBL-RS485-HY.zip

Module Configuration 2
If you have a PC running Linux, unzip the archive to a new directory, open a terminal in that directory and type “make”. The “monitor” program should be created.
Plug in your USB to RS485 adapter and find its device name. Check you can access the device from the terminal where you built the software. If you cannot, either change the device permissions so that you can or run the configuration software with root privileges (the former is recommended).
You may then jump ahead to “Module Configuration 4”.
If you only have a PC running Windows then extra steps are required so continue with the next section.

Module Configuration 3
In order to build and run the configuration software on a Windows PC, you will need to boot your machine into a “Live” Linux distribution. This will provide a temporary Linux environment and will not affect your Windows installation.
Go to the Linux Mint web site and download the “Mate” version:

https://linuxmint.com/download.php

You will need a 4GB or larger USB flash drive to install Linux onto. Follow the instructions on this page to create the bootable drive:

https://linuxmint-installation-guide.readthedocs.io/en/latest/burn.html

Onto a second flash drive, unzip the contents of the GRBL-RS485-HY.zip archive.

Now boot your PC from the flash drive.
Once the Linux Mint Mate desktop appears, follow these instructions (they are case-sensitive and a copy of them is available in the “Extras” sub-directory of the unzipped archive as “Mate_Live.txt”):

Click the "Menu" option on the bar at the bottom of the screen - the menu will appear.
Click on "Terminal" - a terminal window will appear.
Enter "`sudo su`".
Enter "`mkdir /tmp/ramdisk`"
Enter "`chmod 777 /tmp/ramdisk`"
Enter "`mount -t tmpfs -o size=4m myramdisk /tmp/ramdisk`"

Now plug in your flash drive containing the unzipped archive - the file manager window will open.
Right click on the white space on the right-hand panel and click "Open in Terminal" OR
click on the file manager "File" option on the menu bar and click "Open in Terminal" - another terminal window will appear.
Enter "`cp -R * /tmp/ramdisk`"
Enter "`exit`" - the terminal window will close.
Click the "X" at the top right of the file manager window OR
click on the file manager "File" option on the menu bar and click "Close" - the file manager window will close.
In the remaining terminal window, enter "`cd /tmp/ramdisk`".
Enter "`ls`" and you should see the files required to build the configuration software.
Enter "`make`" and the configuration software should be created.
Enter "`ls`" and you should now see a program in green called "monitor".

Enter "`ls /dev/ttyUSB*`" - it will probably say "No such file or directory" but may show some names depending on the hardware plugged in to your PC.
Now plug your USB to RS485 adapter in and wait a couple of seconds.
Enter "`ls /dev/ttyUSB*`" again - it should show exactly one more entry than it did before.
This is the device name for your adapter and if no entries were shown before, will be "/dev/ttyUSB0".

Module Configuration 4 – PWMMAX
Your VFD should be turned off during this configuration procedure.

If your spindle produces 24000rpm when the VFD is running at 400.0Hz, then the conversion factor is 24000 ÷ 400 = 60. Use this value with the “-f” FREQ2RPM option below.

With the module set at its default address of 2 and the VFD set at its default address of 1, run the configuration software using the device name of the USB to RS485 adapter at 9600bps with the FREQ2RPM value calculated above:

```
./monitor -m MASTER -p 2 -v 1 -r 9600 -s SIMVFD -f 60 -d /dev/ttyUSB0
```

Power the module up and wait at least five seconds. If the external abort switch is already latched, release it. Then press the abort switch and release it again. This will release the module's safety interlock.

Messages should start to appear on the screen.

Press “s” or “S” followed by “<Enter>” to enter Slave mode.
The messages should stop and a menu will be displayed:

```
PIC Commands
V: Read Firmware Version Number D: Debug Screen Toggle
1: Read ADC on UIP1             2: Read ADC on UIP2
L: Write I/O lines              R: Read memory
P: Read raw PWM                 W: Write memory
M: Read System Mode             F: Flash EEPROM
G: Ramp Stop Toggle             B: Reboot
S: Slave/Master Toggle          C: Change settings
A: Auto/Manual Toggle           T: ANSI TTY Mode
Q: Quit
```

Enter “p” with GRBL in a stopped state and you should see:

```
Send read raw PWM
PIC reply with raw PWM value: 0    OUTPUT OFF
```

Set the GRBL controller to run at a speed of 1rpm (G code “S1 M3”).
Enter “p” and you should see something similar to:

```
Send read raw PWM
PIC reply with raw PWM value: 20394
```

Enter “p” again and you should see something similar to:

```
Send read raw PWM
PIC reply with raw PWM value: 20395
```

Enter “p” again and you should see something similar to:

```
Send read raw PWM
PIC reply with raw PWM value: 20393
```

Repeat this procedure a few times to get a range of values. All of them should be very close together. If any value is far away from the average value then ignore it. Once you have a few values within that narrow range, select the largest one – i.e. 20395 in the above example. This value is your “PWMMAX”.

Module Configuration 5 – PWMMIN

Set your GRBL controller to run at a speed of maximum rpm minus one rpm. So if your GRBL controller has a value of 24000 for parameter 30, set your speed to “23999” (G code “S23999 M3”).
Enter “p” and you should see something similar to:

```
Send read raw PWM
PIC reply with raw PWM value: 15343
```

Enter “p” again and you should see something similar to:

```
Send read raw PWM
PIC reply with raw PWM value: 15345
```

Enter “p” again and you should see something similar to:

```
Send read raw PWM
PIC reply with raw PWM value: 15344
```

Repeat this procedure a few times to get a range of values. All of them should be very close together. If any value is far away from the average value then ignore it. Once you have a few values within that narrow range, select the smallest one – i.e. 15343 in the above example. This value is your "PWMMIN". You may now stop your GRBL controller (G code "M5").

Module Configuration 6 – Setting the major module parameters

To set the module parameters we must first read the stored values.
Enter "r" to read memory and you should see something similar to:

```
Send read memory  offset: 0
PIC reply with Mem offset: 0  Location1: 0x4F  Location2: 0xA8

Send read memory  offset: 2
PIC reply with Mem offset: 2  Location1: 0x3B  Location2: 0xEF

Send read memory  offset: 4
PIC reply with Mem offset: 4  Location1: 0x5D  Location2: 0xC0
…
1 PWMMAX  (PWM at minimum RPM):    20392
2 PWMMIN  (PWM at maximum RPM):    15343
3 VFDMAX  (VFD maximum frequency): 24000
4 VFDMIN  (VFD minimum frequency): 0
5 VFDMIS  (VFD ramp stop RPM):     120
6 VFDFAC  (VFD frequency to RPM):  60
7 PICADDR (PIC RS485 address):     2
8 VFDADDR (VFD RS485 address):     1
9 BAUD    (RS485 baud):            9600
  WDTCNT  (Watchdog reboot count): 0
0 SYSMODE (PIC System mode):         SLAVE - AUTO - RAMP STOP OFF - EXT MODE
DISABLED
```

Module Configuration 7 – Setting PWMMAX

Enter "c" to change settings and you should see something similar to:

```
1 PWMMAX  (PWM at minimum RPM):    20392
2 PWMMIN  (PWM at maximum RPM):    15343
3 VFDMAX  (VFD maximum frequency): 24000
4 VFDMIN  (VFD minimum frequency): 0
5 VFDMIS  (VFD ramp stop RPM):     120
6 VFDFAC  (VFD frequency to RPM):  60
7 PICADDR (PIC RS485 address):     2
8 VFDADDR (VFD RS485 address):     1
9 BAUD    (RS485 baud):            9600
0 SYSMODE (PIC System mode):         SLAVE - AUTO - RAMP STOP OFF - EXT MODE
DISABLED
Select entry to change (q to quit):
```

Enter "1" to change PWMMAX and you should see something similar to:

```
The highest PWM count for the lowest RPM.
Test at minimum non-zero speed - typically 1 RPM.
For air-cooled spindles with minimum operating speeds, set GRBL $31 to at least
minimum spindle RPM.
Make sure this setting corresponds to (4) VFDMIN.

PWMMAX (20392):
```

Enter your value of PWMMAX.

Module Configuration 8 – Setting PWMMIN
Enter “2” to change PWMMIN and you should see something similar to:

```
The lowest PWM count for the highest RPM.
If GRBL $30 is 24000 then test at 23999 RPM.
Make sure this setting corresponds to (3) VFDMAX.

PWMMIN (15343):
```

Enter your value of PWMMIN.

Module Configuration 9 – Setting VFDMAX
VFDMAX is the maximum spindle rpm and should match GRBL parameter 30.
Enter “3” to change VFDMAX and you should see something similar to:

```
The maximum spindle speed GRBL $30 RPM.
Make sure this setting corresponds to (2) PWMMIN.

VFDMAX (24000):
```

Enter your value of VFDMAX.

Module Configuration 10 – Setting VFDMIN
VFDMIN is the minimum spindle rpm and should match GRBL parameter 31.
Enter “4” to change VFDMIN and you should see something similar to:

```
The minimum spindle speed GRBL $31 RPM.
Make sure this setting corresponds to (1) PWMMAX.

VFDMIN (0):
```

Enter your value of VFDMIN.

Module Configuration 11 – Setting VFDFAC
VFDFAC is the conversion factor from spindle speed to VFD frequency.
VFDFAC = spindle rpm ÷ VFD frequency. It is the same value as the “FREQ2RPM” conversion factor used by the monitor program.

Enter “6” to change VFDFAC and you should see something similar to:

```
The conversion factor from spindle RPM to VFD frequency.
For spindle that runs at 24000 RPM at 400 Hz VFD frequency, value is 24000/400

VFDFAC (60):
```

Enter your value of VFDFAC.

Module Configuration 12 – Saving settings
All the major parameters are now set. You must now write them back to the module.

Enter “q” to leave the change settings menu and return to the main menu.
Enter “w” to write the settings back to the module and you should see something similar to:

```
Send write memory  offset: 0  byte1: 79  byte2: 168
PIC confirming memory write to offset: 0 with value1: 79  value2: 168

Send write memory  offset: 2  byte1: 59  byte2: 239
```

```
PIC confirming memory write to offset: 2 with value1: 59  value2: 239
...
Send write memory  offset: 16  byte1: 2  byte2: 0
PIC confirming memory write to offset: 16 with value1: 2  value2: 0
 SLAVE AUTO Ramp Stop Off External Auto Off
Don't forget to flash EEPROM to store new settings.
```

Enter “f” to write the settings to the module's flash memory and you should see something similar to:

```
Send flash EEPROM
PIC reply with FLASH EEPROM(0=okay): 0
```

These settings will take effect from the next module power up.
Enter “b” to force an immediate module reboot and you should see something similar to:

```
Send write Reboot
```

Messages should then start to appear on the screen as the module reverts to Master mode.

Module Configuration 13 – Additional settings

VFDMIS: This is the spindle speed to ramp down to when using the “Ramp Down To Stop” mode. Ramp down to stop is not recommended for use with air-cooled spindles. Ramp down to stop is designed to reduce the time between turning the spindle off and it actually stopping. Instead of just turning the spindle off when a requested spindle speed of zero is detected, the VFD is told to ramp the spindle speed down to VFDMIS and then turn the spindle off. The spindle then has less inertia and takes less time to actually come to a stop. This can be beneficial if you are doing manual tool changes. Care should be taken not to set it too low as spindle torque is low at low speeds and spindle stalling should definitely be avoided. There is a hard lower limit of 120rpm. It should be noted that if the reason for the requested zero spindle speed is the activation of the external abort switch, ramp down to stop is temporarily disabled and power is immediately removed from the spindle.

PICADDR: This is the RS485 address of the module – so called because the module is controlled by a PIC microcontroller. It should not be necessary to change it but the option is there if you want to. Just remember what it is since you need it to talk to the module.

VFDADDR: This is the RS485 address of the VFD. It should not be necessary to change it but the option is there if you want to. Just remember that it has to match parameter PD163 in the VFD.

BAUD: This is the baud rate of the module's RS485 interface. All the devices on the bus must be set to the same value. The default value is 9600bps. It can be set as high as 38400bps but the actual message throughput across the bus does not significantly increase as the MODBUS-like protocol used by the Huanyang VFDs contains at least 50ms of quiet time before and after each message transmission.

SYSMODE: The first portion of SYSMODE is whether the module should operate in Auto or Manual mode.
In Manual mode, when a speed change is detected from the GRBL controller, the appropriate frequency is sent to the VFD to set the spindle to the matching speed. A successful reply must be received from the VFD or else the GRBL external abort line will be triggered to avoid a machine crash. The machine can still be crashed if the spindle is close to the work piece and is commanded to move any axis before the spindle has got up to speed. If you want to use this mode then I

strongly recommend the use of a “G4 P10.0” instruction after a spindle start or speed change to allow time for the spindle to reach the correct speed.

In Auto mode, a similar sequence of events occurs but as soon as the speed change is detected, the GRBL external hold line is triggered to stop the GRBL controller from executing any further G code instructions. The frequency message is sent to the VFD and the actual frequency that the VFD has ramped to is monitored until it gets to within 5% of the target. When the actual frequency is within the 5% tolerance, the GRBL external resume line is triggered to allow the GRBL controller to continue executing G code. In effect, Auto mode is a full closed loop system and is the safest control mode when executing programs. It is still a good idea to include a“G4 0.5” instruction after a spindle start or speed change as it provides a little time for the module to detect the speed change and trigger the hold line before the CNC machine starts moving in any axis. If nothing else, if the spindle fails to get to the right speed, the module forces an abort or even if you force an abort, then the machine will not have actually changed its position which may be useful.
This setting controls whether the module will start up in Auto or Manual mode. It is possible to toggle between Auto and Manual modes when the module is running without the module being in Slave mode or alternatively with the use of the external Auto/Manual switch. The protocol details to achieve this are explained later.
When you are running G code programs with your GRBL controller then Auto mode is probably what you want to use. However, if you use MDI (Manual Data Input) with your controller then it is probably safer to switch to manual mode. The reason is that when you issue an “M3” or “M4” command (spindle start) in Auto mode, the module will toggle the HLD and RSM lines which your G code sender may interpret as a program start and, if you have a G code program already loaded, it may start to feed the G code instructions to your controller. If you are using the monitoring software, you can easily switch between the two modes or you can enable the external Auto/Manual mode and use a toggle switch instead.

The second portion of SYSMODE is whether “Ramp Down To Stop” mode should be activated. It is turned off by default. It is possible to toggle between ramp down to stop being enabled or disabled without the module being in Slave mode. The protocol details to achieve this are explained later.

The third portion of SYSMODE is whether the “External Auto/Manual” mode switch should be enabled. It is disabled by default. When enabled, if the A/M line is open on JP10, the system will be in Auto mode. If the A/M line is connected to GND, then the system will be in Manual mode. This will override any change you try to make through the monitoring software.

When changing settings, the same procedure should always be used:
“r” read, “c” change, “w” write, “f” flash and “b” boot.

Once the module is correctly configured, no additional RS485 interfaces are required. The module acts as the bus master and operates in a standalone fashion. However, the configuration software can also be used as monitoring software with the ability to alter the working modes of the module.

<u>Suggested Start-up Sequence</u>
When starting up your CNC machine, I recommend this sequence.
Turn on the VFD and wait until it completely boots.
Turn on the GRBL controller, module and associated hardware (including pump control if fitted).
Wait at least five seconds after GRBL/module power up. If the external abort switch is already latched, release it. Press the external abort switch. Release the external abort switch (this will release the power on safety interlock).
Turn on the RS485 isolator/repeater.

I recommend always running a warm-up G code program to ensure the module and GRBL controller are properly synchronised, the machine homed and the VFD/spindle correctly responding to commands.

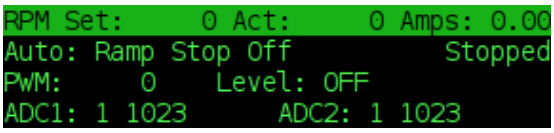
Module Protocol Details

The module always boots into bus master mode unless the external hold button is held down while the external abort switch is being toggled at which time it will boot into bus slave instead.

When the module is bus master, only it may originate communication and it doesn't require its own address to do so. It does however have an address which it uses when it is a slave. Since no other interface can communicate directly with the module, the module creates regular messages addressed to its own address as a slave – safe in the knowledge that no reply to those messages should be forthcoming. While the module waits for the message reply to time-out, other RS485 interfaces on the bus can take that opportunity to send a reply apparently from that slave address containing a message to alter the module's operating mode. The configuration software contains this functionality and can be used to not only monitor all the communications on the bus but also to change the module's operating mode “on the fly”. Switching into slave mode as performed during “Module Configuration 4” is an example of this mechanism.

Configuration Software as Monitor

When the configuration software is run with the "-m VTMONITOR" option, a simple text graphics display is created like this:



The top line shows information about the VFD.
The first entry is the "set" spindle speed that has been requested by the module.
The second entry is the "actual" spindle speed currently being produced by the VFD.
The third entry is the amount of current being used by the spindle.
If there is good communication with the VFD, the top line will be green. If there is communication failure with the VFD, the line will go red.

The lower three lines show information about the module.
The first shows whether it is in Auto or Manual mode, whether Ramp Down To Stop is enabled and the current system status of "Stopped" or "Running".

The next line shows the current raw PWM value. If it says "bad" then the module is unable to successfully measure the PWM drive signal from the GRBL controller. If it says "bad tolerance" then it was able to successfully measure the PWM drive signal but the value obtained is larger or smaller than the PWMMAX or PWMMIN settings. The Module Configuration 4 and 5 procedures should be performed to see if that rectifies the problem.
If the PWM drive signal is within limits then it will display the level as "OFF" when the output is 0V, "MAX" when the output is +5V and a number, "1" to "254", for the intermediate levels.

The bottom line shows the values of the two user inputs as straight 1 or 0 digital values and as 10bit ADC values.

If there is good communication with the module, the bottom three lines will be green. If there is communication failure with the module, the lines will go red. If the module is in Slave mode, the lines will go red and say "WARNING SLAVE MODE".

When in VTMONITOR mode, several key presses can be entered into its window:
"A" toggles between Auto and Manual modes.
"G" toggles between Ramp Down To Stop On and Off modes.
"S" toggles between Slave and Master modes.
"T" toggles between regular and TTY modes.
"Q" quits the program.

Note that whatever settings are toggled, the default values will be used again after the module is power cycled.

It should be noted that the "`-s SIMVFD`" command line option is used to simulate a VFD and should not be used when operating with a real VFD.

The program looks for an RS485 device called "/dev/RS485" by default. This can be created by using a UDEV rule to map a "ttyUSB" device to that name and to set its permissions. For examples of how to do this, refer to the "Extras" sub-directory of the software archive and look at the file "UDEV_rules.txt". If you use this shortcut and all the default values for the module, you can end up with a command line that is only:
"`./monitor -m VTMONITOR -f 60`"

and the "`-f 60`" can be omitted too if that is the correct conversion factor for your system. Of course, you can always edit the source code to change the defaults and rebuild the program.

In the "Extras" sub-directory are also a couple of scripts for xterm and rxvt terminals to create windows that are only the size of the four line monitor display. If you're running this program on a Raspberry Pi with the Openbox window manager, you can remove the window borders and position the terminal window anywhere on your screen by editing the "rc.xml" file and launching the program with the "xtermmonitor.sh" script for an xterm window or "rxvtmonitor.sh" for an rxvt window. Note that those example scripts have "-s SIMVFD" set. The actual command line switches should be set as required. More details can be found in "openbox-window_position.txt" and "lxde.txt" and a script, "resetlxde.sh", is provided to activate the changes.
"xterm" and "rxvt" may not be installed on your system by default. These scripts may also work on other Linux systems with results depending on your particular window manager.

Huanyang Variable Frequency Drive Compatibility

This module should be compatible with the following VFDs:

HY00D423B
HY0D7523B
HY01D523B
HY02D223B
HY0D7543B
HY01D543B
HY02D243B
HY03D743B
HY05D543B
HY07D543B
HY001143B
HY001543B
HY18D543B
HY002243B
HY003043B
HY003743B
HY004543B
HY005543B
HY007543B
HY009043B
HY011043B
HY013243B
HY016043B
HY018543B
HY020043B
HY022043B
HY025043B
HY028043B
HY030043B
HY031543B