

Home

Alexander edited this page on 26 May · [23 revisions](#)

Welcome to the vfdmod wiki!

[Switch to Russian / Читать на Русском](#)

Table of contents

1. [Quick start](#)
2. [PyVcp generator](#)
3. [Configuration file description](#)
 - [Common group](#)
 - [RS485 group](#)
 - [Control group](#)
 - [SpindleRpmIn group](#)
 - [SpindleRpmOut group](#)
 - [User parameter group](#)

Quick start

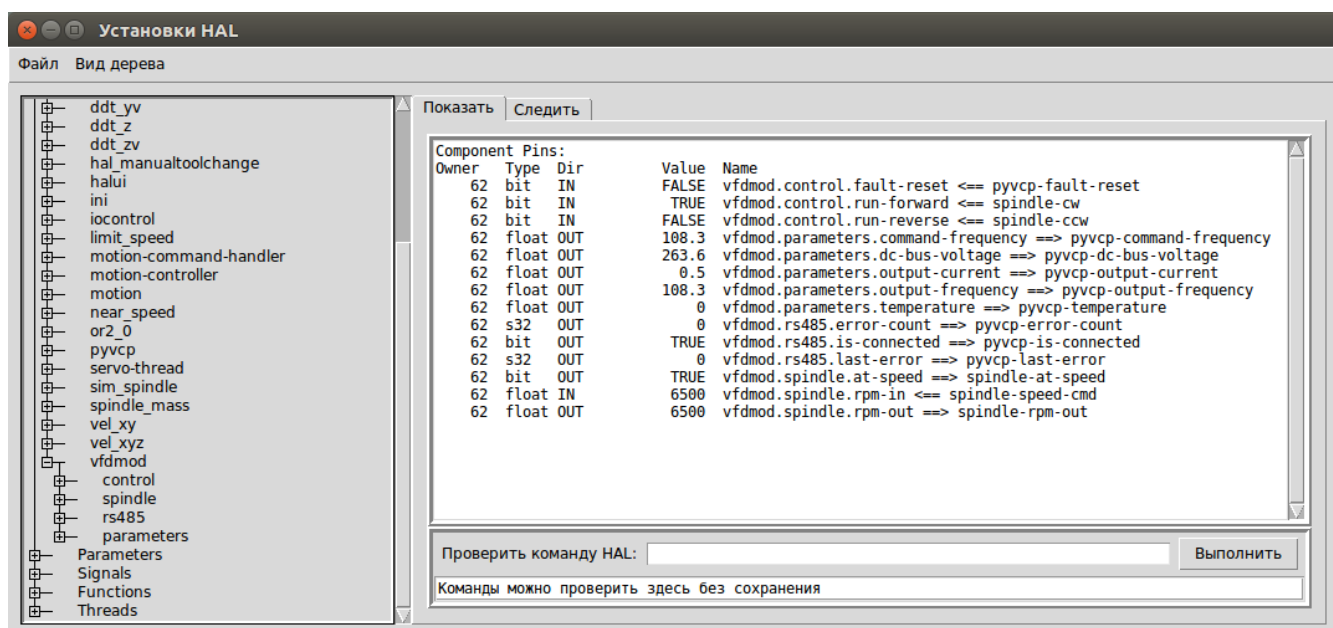
- Create a blank configuration file:
 - `vfdmod --new > my-config-file.ini`
- Open my-config-file.ini in a text editor (gedit, mcedit, mousepad etc.) and write the correct parameters of your VFD.
- Check configuration file for errors:
 - `vfdmod --check my-config-file.ini`

If everything is fine you will see a message **--- Config file is OK ---** or wrong parameter name in case of error.

- Attach vfdmod to your LinuxCNC machine configuration, in the most cases you have to append to custom.hal these lines:
 - `# If you wish to see debug messages insert --debug flag`
 - `# loadusr -W vfdmod --debug my-config-file.ini`
 - `loadusr -W vfdmod my-config-file.ini`
 - `net spindle-speed-cmd => vfdmod.spindle.rpm-in`
 - `net spindle-at-speed <= vfdmod.spindle.at-speed`
 - `net spindle-cw => vfdmod.control.run-forward`

- `net spindle-ccw => vfdmod.control.run-reverse`
- Be sure your main HAL file contains signal names above. For example, in some configurations signal `spindle-speed-cmd` may be replaced by `spindle-vel-cmd-rpm`.
- Run your LinuxCNC machine and make sure vfdmod component is loaded and it's pins are available:

HAL pin name	Pin type	I/O	Description
<code>vfdmod.spindle.rpm-in</code>	float	in	Spindle command speed
<code>vfdmod.spindle.rpm-out</code>	float	out	Spindle output speed
<code>vfdmod.spindle.at-speed</code>	bit	out	Spindle is at speed
<code>vfdmod.control.run-forward</code>	bit	in	Run forward input
<code>vfdmod.control.run-reverse</code>	bit	in	Run reverse input
<code>vfdmod.control.fault-reset</code>	bit	in	Fault reset input
<code>vfdmod.rs485.is-connected</code>	bit	out	Connection status pin
<code>vfdmod.rs485.error-count</code>	s32	out	Total error count
<code>vfdmod.rs485.last-error</code>	s32	out	Last error code
<code>vfdmod.parameters.xxx</code>	*	out	User defined pins

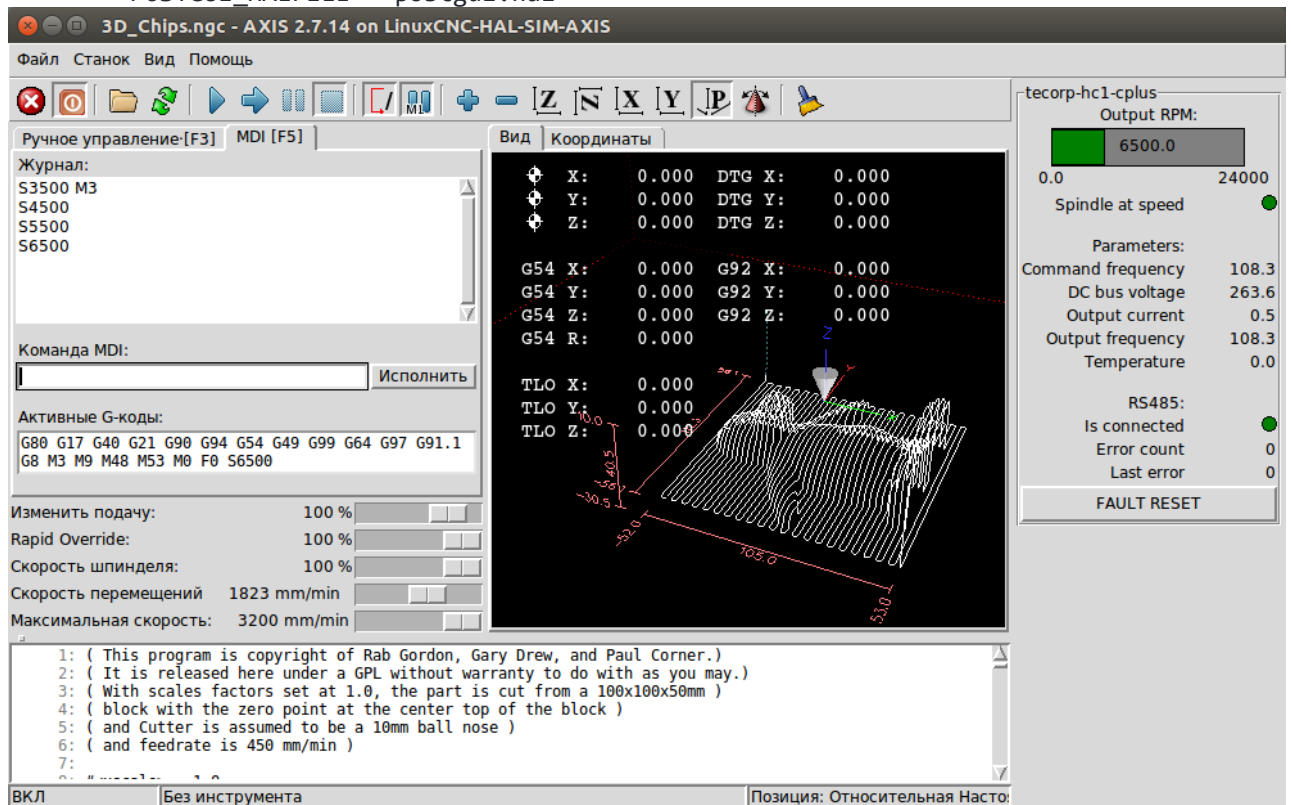


PyVcp generator

Vfdmod provides to make a very simple PyVcp control panel that indicates communication state, current spindle speed, at speed state and all user defined parameters.

- Make a pair of files (xml & hal):

- `vfdmod --pyvcp my-config-file.ini > my-pyvcp-panel.xml`
- `vfdmod --postgui my-config-file.ini > postgui.hal`
- Attach these files to your machine INI file:
- ...
- `[DISPLAY]`
- `PYVCP = my-pyvcp-panel.xml`
- ...
- `[HAL]`
- `POSTGUI_HALFILE = postgui.hal`



Configuration file description

Configuration file is a plain text INI file that includes five predefined groups and any count of user defined groups. Each group includes few required and optional parameters. Predefined groups are provide VFD's basic functionality, such as: start, stop, fault reset, setting a frequency, reading an output speed and at speed state. User groups are allow to monitor any count of the additional parameters: output voltage, output current, temperature, uptime etc.

Predefined group names are:

- `[Common]`
- `[RS485]`
- `[Control]`
- `[SpindleRpmIn]`
- `[SpindleRpmOut]`

An optional user groups can be named at user's choice, the only condition is group name shall be unique and differ from other groups. For example: [User parameter], [user-parameter], [123], [Output voltage], [Uptime]. As you can see spaces are allowed, but please note all group names (and parameters too) are case insensitive, so [RS485] and [rs485] are the same.

Common group

Parameter	Required	Default value	Range
ComponentName	No	vfdmod	an ASCII string
MaxSpeedRPM	Yes		unsigned integer
MinSpeedRPM	Yes		unsigned integer
AtSpeedThreshold	No	0.05	0.00 ... 1.00

- Use ComponentName if you wish to set your own HAL component name instead of default vfdmod value. In that case don't forget to edit your custom.hal:
- # Replace -W by -Wn with HAL component name
- loadusr -Wn <component-name> vfdmod my-config-file.ini
- Spindle limits shall be defined by MaxSpeedRPM and MinSpeedRPM parameters, note that MaxSpeedRPM shall be greater than (or equal to) MinSpeedRPM, and MinSpeedRPM shall be greater than zero.
- AtSpeedThreshold sets the maximum allowed difference between command speed and output speed. If difference is lower or equal to specified value then HAL at-speed output will be set to TRUE, and to FALSE in other case. Value 1.00 means 100%, 0.25 means 25%, 0.05 means 5% (default) etc.

RS485 group

Parameter	Required	Default value	Range
SlaveAddress	Yes		0 ... 255
SerialDevice	Yes		an ASCII string
BaudRate	Yes		9600, 19200 etc.
DataBits	No	8	always 8
Parity	No	N	N, E, O
StopBits	No	1	1 or 2
LoopDelay	No	200	0 ... 10000
ProtocolDelay	No	4	0 ... 100

Parameter	Required	Default value	Range
IsConnectedDelay	No	10	1 ... 100
ConnectionErrorList	No		comma separated integers
ConnectionDelay	No	1000	0 ... 10000

- SlaveAddress sets the address of the VFD, for example 1.
- SerialDevice sets communication device path, for example /dev/ttyS0.
- BaudRate sets communication speed, for example 38400.
- DataBits sets data bits count, shall be 8.
- Parity sets parity checking mode: N - without parity, E - even, O - odd.
- StopBits sets stop bits count, shall be 1 or 2.
- LoopDelay sets the delay in milliseconds when all VFD's registers have been read and written.
- ProtocolDelay sets the delay in characters at front of every MODBUS request. MODBUS specification requires at least 3,5 characters, so a value lower than 4 has no sense. Increase it if connection is not stable and CRC errors are growing.
- IsConnectedDelay sets the minimum count of successful Modbus requests to set HAL is-connected pin to TRUE. In case of any transmit error this HAL pin will be set to FALSE immediately.
- ConnectionErrorList sets the critical errors codes that cause reconnection event. During reconnection current MODBUS connection will be closed and re-opened. If this parameter is empty then reconnection mode will be disabled.
- ConnectionDelay sets the delay in milliseconds between reconnection attempts if parameter ConnectionErrors is not empty.

Control group

Parameter	Required	Default value	Range
FunctionCode	No	0x06	0x06, 0x10, 0x05, 0x0F

- FunctionCode sets the communication function code to control a VFD, only four functions are supported:
 - write single holding register (0x06)
 - write multiple holding registers (0x10)
 - write single coil (0x05)
 - write multiple coils (0x0F)

Parameters below are valid if FunctionCode is 0x06 (default vlaue) or 0x10!

Parameter	Required	Default value	Range
Address	Yes		0x0000 ... 0xFFFF

Parameter	Required	Default value	Range
RunForwardValue	Yes		0x0000 ... 0xFFFF
RunReverseValue	Yes		0x0000 ... 0xFFFF
FaultResetValue	No		0x0000 ... 0xFFFF
StopValue	Yes		0x0000 ... 0xFFFF

- Address sets the address of the control register.
- RunForwardValue sets the value that turns spindle on in forward direction (clockwise).
- RunReverseValue sets the value that turns spindle on in reverse direction (counter clockwise).
- FaultResetValue sets the value that resets VFD fault state. If this parameter is not specified then fault reset function will be disabled.
- StopValue sets the value that stops spindle off.

Parameters below are valid if FunctionCode is 0x05 or 0x0F!

Parameter	Required	Default value	Range
RunCoil	Yes		0x0000 ... 0xFFFF
DirectionCoil	Yes		0x0000 ... 0xFFFF
FaultResetCoil	No		0x0000 ... 0xFFFF

- RunCoil sets the address of the run/stop coil.
- DirectionCoil sets the address of the direction coil.
- FaultResetCoil sets the address of the fault reset coil. If this parameter is not specified then fault reset function will be disabled.

SpindleRpmIn group

Parameter	Required	Default value	Range
FunctionCode	No	0x06	0x06, 0x10
Address	Yes		0x0000 ... 0xFFFF
Multiplier	No	1	unsigned integer
Divider	No	1	unsigned integer

- FunctionCode sets the communication function code to set VFD's command frequency, only two functions are supported:
 - write single holding register (0x06)
 - write multiple holding registers (0x10)

- Address defines address of the command speed (or frequency) register.
- Multiplier and Divider are integer values that correct command speed value before it will be written to the command speed register. For example, if command speed is 2667 rpm, Multiplier is 3, Divider is 2, then command speed register value will be $(2667 * 3 / 2) = 4000$.

SpindleRpmOut group

Parameter	Required	Default value	Range
Address	Yes		0x0000 ... 0xFFFF
Multiplier	No	1	unsigned integer
Divider	No	1	unsigned integer

- Address defines address of the output speed (or frequency) register.
- Multiplier and Divider are integer values that correct output speed after output speed register has been read. For example, if register stores 4000, Multiplier is 2, Divider is 3, then spindle output speed will be $(4000 * 2 / 3) = 2667$ rpm.

User parameter group

Parameter	Required	Default value	Range
FunctionCode	No	0x03	0x01, 0x03

- FunctionCode sets the communication function code to read a user defined parameter, only two functions are supported:
 - read multiple coils (0x01)
 - read multiple holding registers (0x03)

Function code 0x01 allows to read a bit value (coil) and creates HAL pin of bit type. Function code 0x03 allows to read a register and creates one of four HAL pins at user choice: bit, float, s32 or u32.

Parameters below are valid if FunctionCode is 0x01!

Parameter	Required	Default value	Range
Address	Yes		0x0000 ... 0xFFFF
PinName	Yes		an ASCII string w/o spaces

- Address defines an address of the coil, that stores user parameter.
- PinName sets the HAL pin name. Two pins will be created automatically: direct and inverted with -not ending. For example, if PinName=overheating, then two pins will be

available in HAL: `vfdmod.parameters.overheating` and `vfdmod.parameters.overheating-not`.

Parameters below are valid if `FunctionCode` is `0x03` (default)!

Parameter	Required	Default value	Range
Address	Yes		0x0000 ... 0xFFFF
PinType	Yes		bit, float, s32, u32
Multiplier	No	1	unsigned integer
Divider	No	1	unsigned integer
BitMask	No	0xFFFF	0x0000 ... 0xFFFF
PinName	Yes		an ASCII string w/o spaces

- Address defines an address of the register, that stores user parameter.
- PinType sets the HAL pin type.
- Multiplier and Divider are used to correct a value that has been read from user register. These parameters are valid if PinType is not bit.
- BitMask is a bit mask that logically multiplies to a value returned from register. If result is not zero then the HAL output pin will be set to TRUE, or to FALSE in other case. This parameter is valid if PinType is bit.
- PinName sets the HAL pin name. If PinType=bit then two pins will be created automatically: direct and inverted with -not ending. For example, if PinName=overheating, then two pins will be available in HAL: `vfdmod.parameters.overheating` and `vfdmod.parameters.overheating-not`.