Motor Control with VESC 6

First a disclaimer: This document is about the VESC 6. As this is an ongoing project, certain things can change. Even more, not all things are checked. This can lead to falsehoods. This document is not intended for research nor does it give any warranties about the correctness of the information listed in this document. However, as the code for the VESC 6 in online, it is possible to verify all the information listed below.

This document aims to explain different ways to control electrical motors with the VESC 6 (Vedder Electronic Speed Control). This ESC was built primarily for BLDC (BrushLess Direct Current) motor, but can control brushed DC motors as well.

To control a BLDC motor, the first part is detecting its position or phase. The VESC 6 supports 3 different ways to do this. Sensorless (via BEMF, more on this later), Hall Sensor and Encoder. Both Incremental encoders and absolute encoders (AS5047) are supported.

Next, there are different ways to control the motor. There are 2 main ways to control a BLDC motor, trapezoidal commutation and FOC, or Field Oriented Control. The difference will be explained further down below.

# Downloading the VESC-tool software

Assuming you’ve already have a VESC 6 in your possession, the first step to using a VESC 6 is downloading the accompanied software: the VESC-tool. This tool is the successor of the BLDC-tool, and is used to communicate with the VESC via PC. Currently, as the VESC project is in beta, the only place is it available on the internet is [www.vesc-project.com](http://www.vesc-project.com) . You’ll need to log in to be able to download it.

# Connecting the VESC to power

Contrary to the VESC 4, the VESC 6 cannot be powered by USB. So, to connect it to a PC, you’ll need to hook up power to the VESC. This can be done by any battery or power supply, as long as the voltage is between 8V and 57 V. These setting are found in the VESC tool under Motor Settings -> General -> Advanced -> Minimum/Maximum Input Voltage and are best left unchanged. The manual rates the VESC for voltages between 6V and 60 V, so some margin of error was taken here. The manual mentions a lot of safety measures, such as a fuse, safety cut-off and a anti spark switch. These are best implemented, especially for the first time, as faults in the construction of the VESC can lead to shorts.

The VESC comes with a female XT90 connector. So, connecting it to power requires a male connector of the same type, or changing both connectors, which isn’t advised.

# Connecting VESC and PC

When the VESC is powered, just connect a micro USB cable from the micro USB connector to your PC. For more information about the connections of the VESC, look at the manual, page 3.[[1]](#footnote-1)

On your PC with Linux or Windows, start the VESC-tool and connect the pc to the VESC. There is an auto connect-functionality in the VESC-tool under Welcome & wizards -> Connect. This wizard will try to automatically connect the VESC to the PC, and works in a lot of cases. Also, the first two buttons on the bar in the right are reconnect the previous connection and disconnect. If needed though, it is possible to connect in the connection (USB-serial) tab, where you can adjust the baud rate and refresh the list with possible ports. When the VESC is not found by the tool, try shortening the cable, or simply another cable as not every micro-USB cable supports data. When connected, the VESC Tool should give a green status update in the right down corner and the status should be connected.

When connecting the VESC board for the first time, it might give a warning about outdated firmware. If this is the case, try to update the firmware in the Firmware tab. However, if the VESC is unable to reboot itself when done updating and a power on-off reboots it in the old firmware (so, when reconnected, the VESC-tool still warns about outdated firmware), the bootloader is probably missing. This is a known issue and more information can be found on the forums.[[2]](#footnote-2) When it is sure that the bootloader is missing due to the symptoms described above, you can upload the bootloader in the bootloader tab in Firmware. However, doing this without the bootloader missing can cause significant problems, so being sure the bootloader Is missing is important.

When the bootloader is uploaded – no reboot should be required – updating the firmware should be possible, and after the reboot, the VESC tool should be running with the latest firmware.

# Connecting the motor to the VESC – power

The first important thing is the type of connections. Currently, the VESC uses 4 mm female bullet connectors. If your motor doesn’t have these connectors, you can find them on the internet. 4 mm male bullet connectors should fit.

Secondly, the type of motor is an important factor. When connecting a DC motor to the VESC, only connect wire A and C (as printed on the board) to the motor. The way you connect the DC motor to the VESC should only affect the direction the motor is turning.

If the connection to the VESC is from a BLDC motor, you’ll need to make 3 connections. Again, the order in which you connect the 3 wires A, B and C to the motor, should only affect the direction the motor turns. To get correct information in the VESC-Tool, if you know the motor phases, connect Yellow to A, Blue to B and Red to C. More information can be found in the manual.

Right now, it should be possible to run the motor without sensors. Both DC and BLDC motors can be run without sensor.

# BLDC motor setup

VESC-Tool supports the autodetection of motor parameters. This functionality can be found in Welcome & Wizards -> Motor Setup Wizard.

The wizard firstly makes sure you are connected to a powered VESC. Then, it asks whether you want to upload the default values. After this, the wizard asks for the motor type. Choose the correct type depending on the motor and the way of control. If FOC is required, FOC must be selected, even though is isn’t a motor type, technically. The next step is the current limits, for both motor and the battery. For safety reasons, this limit should be set correctly. Check the motor specifications for more detail. For the battery limits: make sure you set the limit to a safe number, as not only the battery / power supply has to be able to support the limit, but the cables have to support the amperage as well. Make sure your fuse and other safety equipment support the limits. The max regen current, this is the maximum current that can flow into the battery, can be set at 0 when using a power supply.

If a battery is used, it is advisable to add some battery cut-off limits, to avoid over-discharging of the battery. Even if using a power supply, this limits seem to affect the detection results. Therefore, making sure the applied voltage is between these voltages can solve some difficulties.

The next step is choosing the sensor mode. Sensorlessly should never give problems with this. When using the ABI encoder, make sure the encoder count is set to 4 times the specified counts per revolution, as the VESC counts both flanks of a pulse, and this for the 2 signals (A and B). To avoid problems with the sensor, sensorless is a good option for the first try.

When running the motor in BLDC, the next step will be to detect Back ElectroMotive Force (BEMF) Coupling and integration limit. More information about them can be found in the BLDC -> Sensorless tab. These values can differ a lot, and their units are not mentioned. This can lead to some arbitrary numbers. For reference of “normal” values, from practical experience, Cycle Itegrator Limit can certainly be anything between 50 to 500 and BEMF coupling from 500 to 900. Note that values outside these ranges can be normal if using a very small or large motor. If problems arise with the integration limit, commutation mode in BLDC -> General ->Commutation mode can be set to delay. This removes the need for integration in the commutation, but can lead to very divergent results.

# FOC motor setup

In FOC, the required parameters are completely different, as is the control. The required parameters are no longer BEMF Coupling and integration limit, but are resistance, inductance, flux linkage and some parameters for the PID controller. For the first step where resistance and inductance are calculated, no parameters should be filled in. These values, in contrast to the values of BLDC, have physical units, and can be checked[[3]](#footnote-3).

## Resistance

The resistance of a motor can be calculated by connecting a multimeter or Ohm meter to two wires of the 3. As the motor should be balanced and symmetrical, the values should be the same for A-B, B-C, and C-A. Of course, measuring all three and taking the average probably gives a better result. If one of these values differs a lot from the others, there is probably a short or other problem with the motor. Half this number should be the phase resistance.

## Inductance

The other parameter is the inductance. This one can be measured via an LCR meter. Measure the impedance for a whole rotation in small increments, and do this again for both A-B, B-C and C-A. If the values are similar, you can average them out and obtain an estimate for the line-to-line inductance. Again, half this number should be the phase inductance.

These values can also be measured by the VESC. The motor makes a strange noise during this, but this should be completely normal. This detection has improved a lot over the detection in VESC 4.

## Flux linkage

The next parameter is the flux linkage λ of the motor. This parameter can be obtained using the following formula[[4]](#footnote-4):

Here is is the rpm constant of the motor, and # poles are the number of poles of the motor.

Of course, if you would want to calculate this by hand, you need to know the and the pole count of the motor. The annotated document can help in finding these parameters[[5]](#footnote-5). Important to note is that the is the inverse of the BEMF constant . This is also mentioned by the previous document.

Normally though, this parameter can be obtained using the VESC-Tool detection. For this though, 3 parameters are needed, I, D and ω. For I and ω, the same values as in the BLDC detection can be used. D can be left at 0,5 or lowered if the load increases with the speed, as mentioned within the VESC Tool.

## PID constants

From these values and the desired time constant (which can be set as you want, the lower the faster the current will be adapted, but the harder the movements can be), the KP, KI and Observer gain constants are calculated. Currently, it is only possible to calculate these parameters after a successful test. If you want to calculate the parameters manually, the formulae are the following[[6]](#footnote-6):

Where λ is the flux linkage calculated in the previous step, and

Where T is the given time constant for the PID controller. Of course, these last three parameters can be manually tweaked if needed. These are just the default values.[[7]](#footnote-7)

If sensors are not required, the motor setup should be done now.

# Sensors

The VESC can use different sensors in different modes. Currently in BLDC mode, VESC can run sensorlessly, with hall sensors, or hybrid. Hybrid is a combination of sensored and sensorless where the sensors aren’t used when a certain speed is reached. This speed can be found and changed in the sensors tab of the BLDC page.

In FOC mode, VESC supports sensorless, hall sensors and encoders of 2 different types: incremental and a family of absolute encoders (AS5047).

## Sensorless

The VESC can run a detected motor sensorlessly. Both in BLDC and FOC mode. This means the sensors aren’t required to run the motor. That said, the purpose of sensors isn’t completely gone of course. Sensors can increase the performance of a motor drastically and can aid the motor controller with the start-up, something that is still very difficult without sensors.

## Hall sensors

One of the most known sensors for motor control are the hall sensors. These sensors use the Hall-effect[[8]](#footnote-8) to measure the position of the motor. These sensors are robust, long living and cheap. These are the reasons a lot of electrical motors come with hall sensors in them as standard. Sadly, they aren’t the most accurate of sensors.

Connecting hall sensors to the VESC isn’t that difficult, just make the connection to the “sense” port of the VESC. It is important to mention that a lot of VESC 6 boards have the wrong imprint on the board itself. The imprint should be mirrored. Look at the manual for more details.[[9]](#footnote-9) This means the board has the same sensor port as the VESC 4, so upgrading to the VESC 6 shouldn’t be difficult.

For the hall sensors, make sure you know your required sensor voltage. The VESC has both 5 V 1A and 3.3 V 0.5 A, which should be enough for any sensor. Switching between these voltages is as simple as changing the – admittedly hard to reach – switch on the VESC. As this can determine whether the sensor will work, or won’t work at all anymore, it pays to check your sensor’s voltage.

After connecting the voltage and ground, connect the 3 hall signals to the hall pins – the order shouldn’t matter. Still, it can’t hurt to look up the proper connection in the data sheet of the hall sensor. Detecting hall sensors is done via the detect hall sensor widget: this widget will automatically pop up in the motor detection wizard, but can also be found in the sensor tab of the BLDC page or hall sensor tab of the FOC page. During this test, the motor will be run in open loop for few cycles forwards and a few backwards. After this test, the detection should return a hall sensor table. The values of this table aren’t explained in the tool, and they differ for each mode.

## Hall table values

For BLDC mode, the hall sensor table contains a mapping of the hall sensors to the motor commutation steps. These phases go from 1 to 6 with -1 for invalid. A correctly detected hall sensor should have exactly 2 entries set to -1.[[10]](#footnote-10)

For FOC, the position of each hall sensor is measured multiple times and an average is taken

The formula is[[11]](#footnote-11):

These entries have a range from 0 to 199, or 255 when the table entry is not found. A correctly detected hall sensor should have exactly 2 entries set to 255.

Where int() is the casting of a number to a whole number by removing the decimals, and the angle is the angle of the hall sensor position to a static point.

As these tables of both modes require knowledge of inside variables, it is very difficult to obtain this table without running the tests. Fortunately, this test for FOC is one of the easier ones. For BLDC on the other hand the tests are integrated with the other BLDC parameters, and a successful test for the base parameters is most likely a successful test for the Hall detection.

## Detecting Hall sensors

When the hall sensors are connected, it is possible to detect them using the VESC Tool. Just select Hall sensors in the sensor tab during the motor setup wizard, or select hall sensors in the general tab of the BLDC or FOC tab, depending on which mode you’re using. The detection is done using the Hall sensor detection widget. This widget will pop up automatically in the motor setup wizard, or can be found in the sensors tab of the BLDC page or Hall sensors tab of the FOC page. This detection uses open loop commutation to spin the motor slowly. During this, the hall sensor table is detected.

## Encoder

# Controlling the motor

## VESC Tool

The VESC Tool features several different ways to control the motor.

### Arrow keys

There are different ways to control the motor. Arguably the easiest are the arrow keys. In the BLDC tool, these were always enabled. In the VESC tool, there is a button in the right toolbar that enables control via keys.

The up and down keys enable current control[[12]](#footnote-12). The target current is the current in the toolbar in the right down corner. To control the current, the VESC changes the duty cycle, which is capped at 95% (for measuring and other reasons). So, if the set current can’t be reached, the duty will stay at 95% when pressing these buttons. The down arrow key just reverses this current. Normally, this means the motor will run at the same speed in the opposite direction.

The left and right keys control duty cycle. Whereas the up and down keys just start current control, the right and left keys do so incrementally. When pressing the right key, the duty cycle goes incrementally to 18%, the left key on the other hand makes the duty go to -10 %. When the duty is on either of these boundaries, pressing the other key (simultaneously) makes the duty go further to 90% or -30 % respectively, depending on the boundary the duty was on.

The page down key activates the breaks and the ESC key disables any signal the VESC is sending to the motor.

### Buttons

There are several buttons for to control the motor in the VESC tool. In the bottom left toolbar, you can enable duty control, current control speed control and position control. Next to it, you can enable brake, handbrake and switch of the control (= ESC key).

Next to these buttons, the text fields will tell what the current setting is the VESC will try to achieve, whether you activate control by current, duty cycle, speed, position, etc.

## Apps

The practical use of the VESC is of course using it with an app. Whether you use the motor in a longboard, in a drone or even in a simple fan, normally you don’t want your pc to be connected all the time.

### PPM

1. Found at <http://www.trampaboards.com/vesc-6-complete--vedder-electronic-speed-controller-trampa-exclusive-p-24166.html> [↑](#footnote-ref-1)
2. Found at <http://www.vesc-project.com/node/47> [↑](#footnote-ref-2)
3. <https://www.quora.com/How-can-I-measure-the-impedance-and-resistance-of-a-BLDC-motor> [↑](#footnote-ref-3)
4. <http://vedder.se/forums/viewtopic.php?t=131#p693> [↑](#footnote-ref-4)
5. <http://www.nxp.com/docs/en/application-note/AN4680.pdf> [↑](#footnote-ref-5)
6. <https://github.com/vedderb/vesc_tool/blob/master/widgets/detectfoc.cpp> [↑](#footnote-ref-6)
7. <https://e2e.ti.com/blogs_/b/motordrivecontrol/archive/2015/07/20/teaching-your-pi-controller-to-behave-part-ii> [↑](#footnote-ref-7)
8. <http://www.electronics-tutorials.ws/electromagnetism/hall-effect.html> [↑](#footnote-ref-8)
9. <http://www.trampaboards.com/vesc-6-complete--vedder-electronic-speed-controller-trampa-exclusive-p-24166.html> [↑](#footnote-ref-9)
10. <https://github.com/vedderb/bldc/blob/79bfbe62344a29817c5e3ea982a2c27d34c6c881/mcpwm.c> [↑](#footnote-ref-10)
11. <https://github.com/vedderb/bldc/blob/79bfbe62344a29817c5e3ea982a2c27d34c6c881/mcpwm_foc.c> [↑](#footnote-ref-11)
12. <https://github.com/vedderb/vesc_tool/blob/master/mainwindow.cpp> [↑](#footnote-ref-12)