VESC – Open Source project

By Benjamin Vedder

Motor Control Techniques and Algorithms

Sensored and sensorless FOC with auto-detection of all motor parameters

- Duty-cycle control
- Speed control
- Current control.
- RPM-Based Phase Advance (Field Weakening): This technique increases the motor’s maximum speed beyond its base speed

Performance Characteristics and Limitations

- Voltage Range: 8V to 60V, compatible with various battery types, including 3S to 12S LiPo batteries.
- Current Capacity: 50A for continuous operation and 240A for brief periods.
- 5V 1A output for external electronics (e.g. microcontroller) from the buck converter on the DRV8302.

System Structure

Hardware:

- STM32F4 microcontroller.
- DRV8302 MOSFET
- IRFS7530 MOEFETs

Software:

- Firmware
- GUI Tool

STM32F4 Microcontroller

STM32F4 Microcontroller: This is the central processing unit in the system. Receives inputs, processes these inputs and generates control signals to be sent to the DRV8302 driver.
Is responsible for communicating with the different interfaces of the system.

DRV8302 MOSFET Driver

Takes control signals from the STM32F4 microcontroller and uses these signals to control the operation of the IRFS7530 MOEFETs

It has built-in current shunt amplifiers that provide feedback to the microcontroller about the current flowing through the motor.

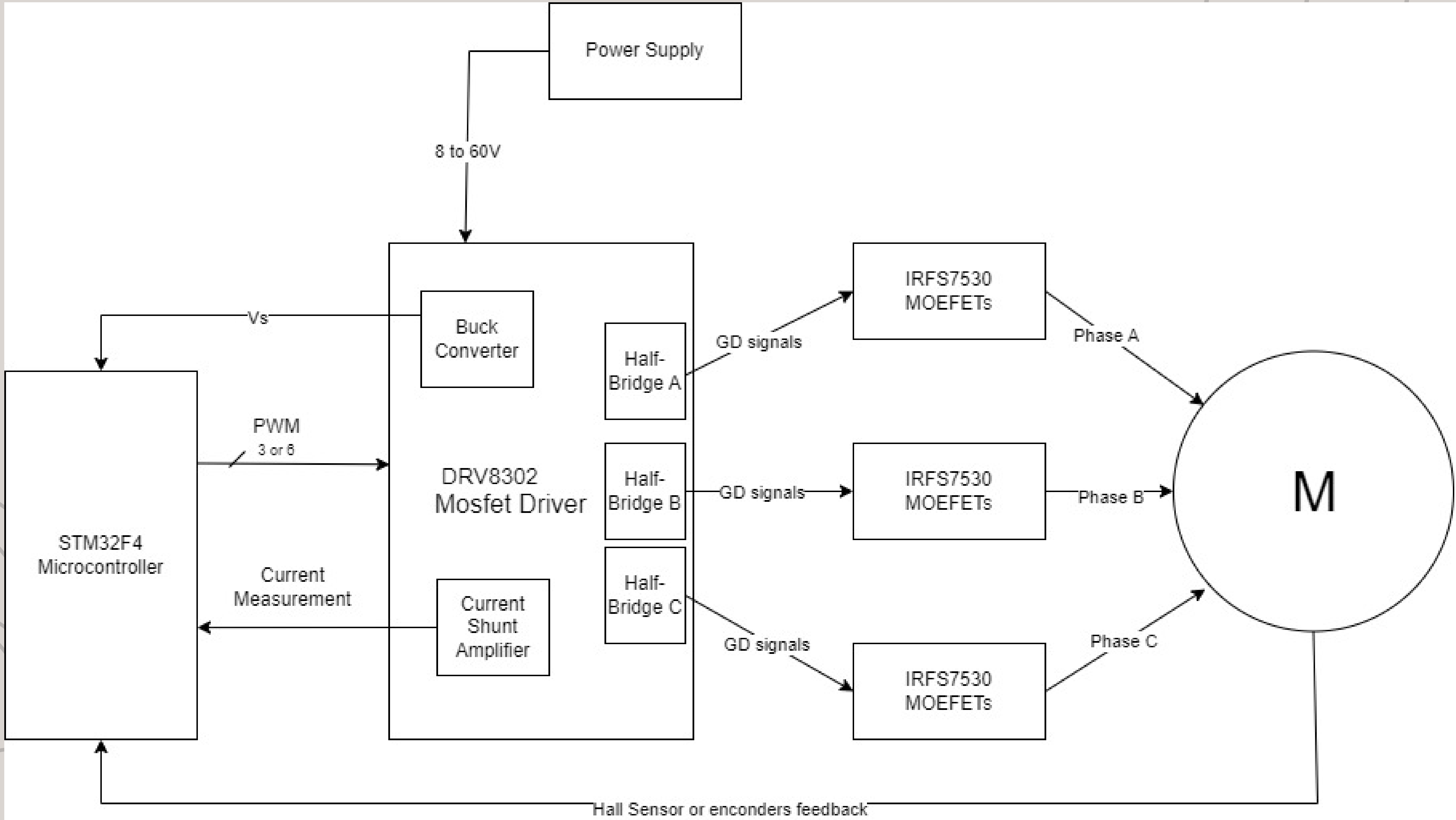
The DRV8302 includes a buck converter that steps down the voltage from the main power supply to power the microcontroller.

IRFS7530 MOSFETS

The IRFS7530 MOEFETs are the system's power switches.

The gate drive signals generated by the DRV8302 MOSFET driver switch the power MOSFETs on and off, effectively controlling the power flow from the battery to the motor.

Hardware



Simple block diagram

Firmware

The firmware of the VESC project is based on the ChibiOS/RT real-time operating system (RTOS).

The firmware runs on the STM32F4 microcontroller and executes the control algorithms that drive the motor.

These algorithms take input from the user interface and sensors to generate control signals for the MOSFET driver.

Firmware

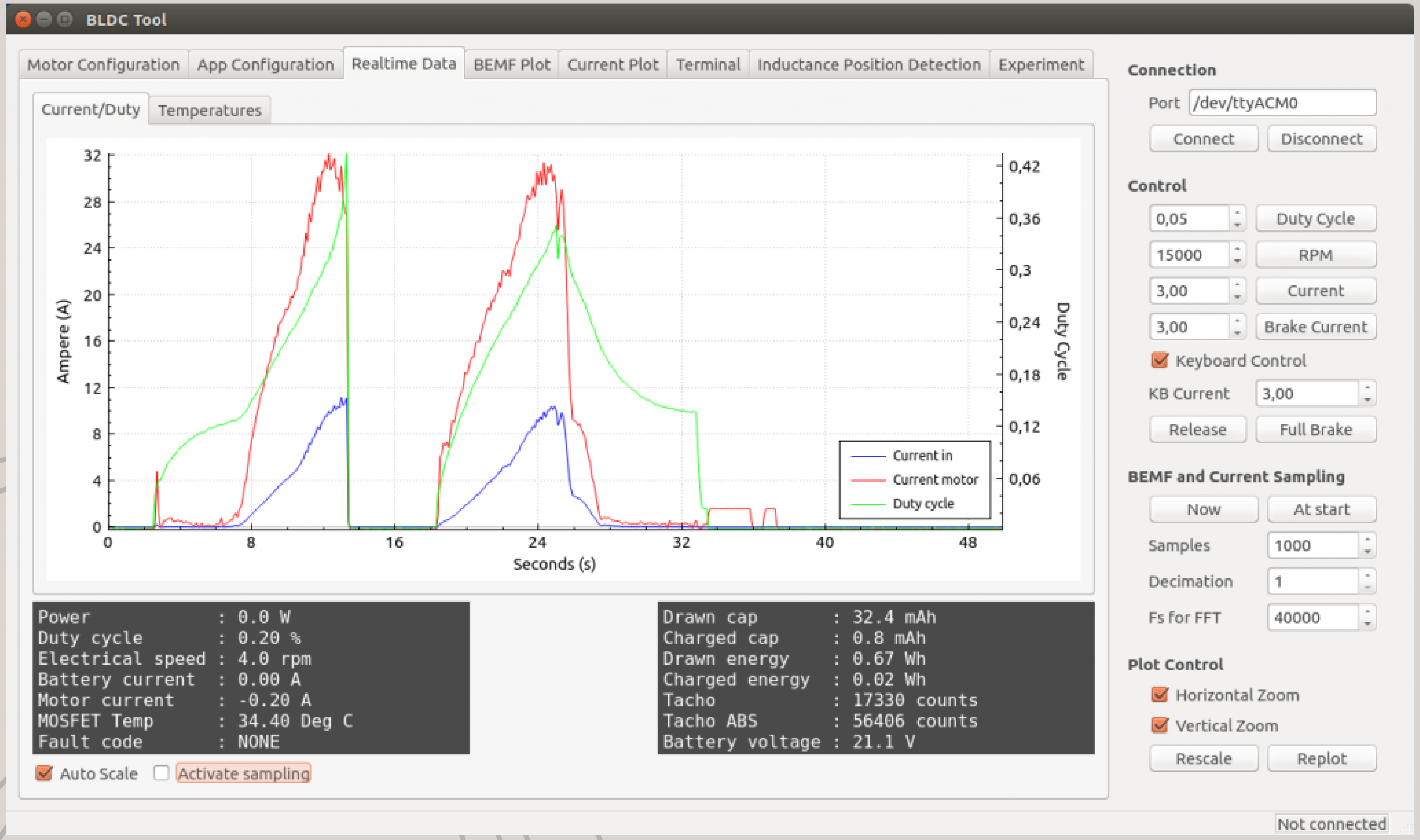
A critical aspect of the firmware of the VESC project is that provides an easy way to write custom applications while keeping the code up to date without having many conflicts when pulling updates using git

GUI Tool

The VESC project includes a GUI known as the VESC Tool, which allows users to:

- Configure the VESC system: Select the control mode, set the motor parameters, configure the input sources (such as PPM, ADC, or UART), and adjust the system’s protection settings;
- Monitor motor operation: provides real-time monitoring of the system’s operation (displays information like motor speed, current, duty cycle, and input voltage, and it can plot these variables over time;
- Update easily the firmware via USB.

GUI Tool



```
Power            : 0.0 W
Duty cycle       : 0.20 %
Electrical speed : 4.0 rpm
Battery current  : 0.00 A
Motor current    : -0.20 A
MOSFET Temp      : 34.40 Deg C
Fault code       : NONE
```

```
Drawn cap       : 32.4 mAh
Charged cap     : 0.8 mAh
Drawn energy    : 0.67 Wh
Charged energy  : 0.02 Wh
Tacho           : 17330 counts
Tacho ABS       : 56406 counts
Battery voltage : 21.1 V
```

☑ Auto Scale ☐ Activate sampling

Connection

Port /dev/ttyACM0

Control

0,05 Duty Cycle

15000 RPM

3,00 Current

3,00 Brake Current

☑ Keyboard Control

KB Current 3,00

BEMF and Current Sampling

Samples 1000

Decimation 1

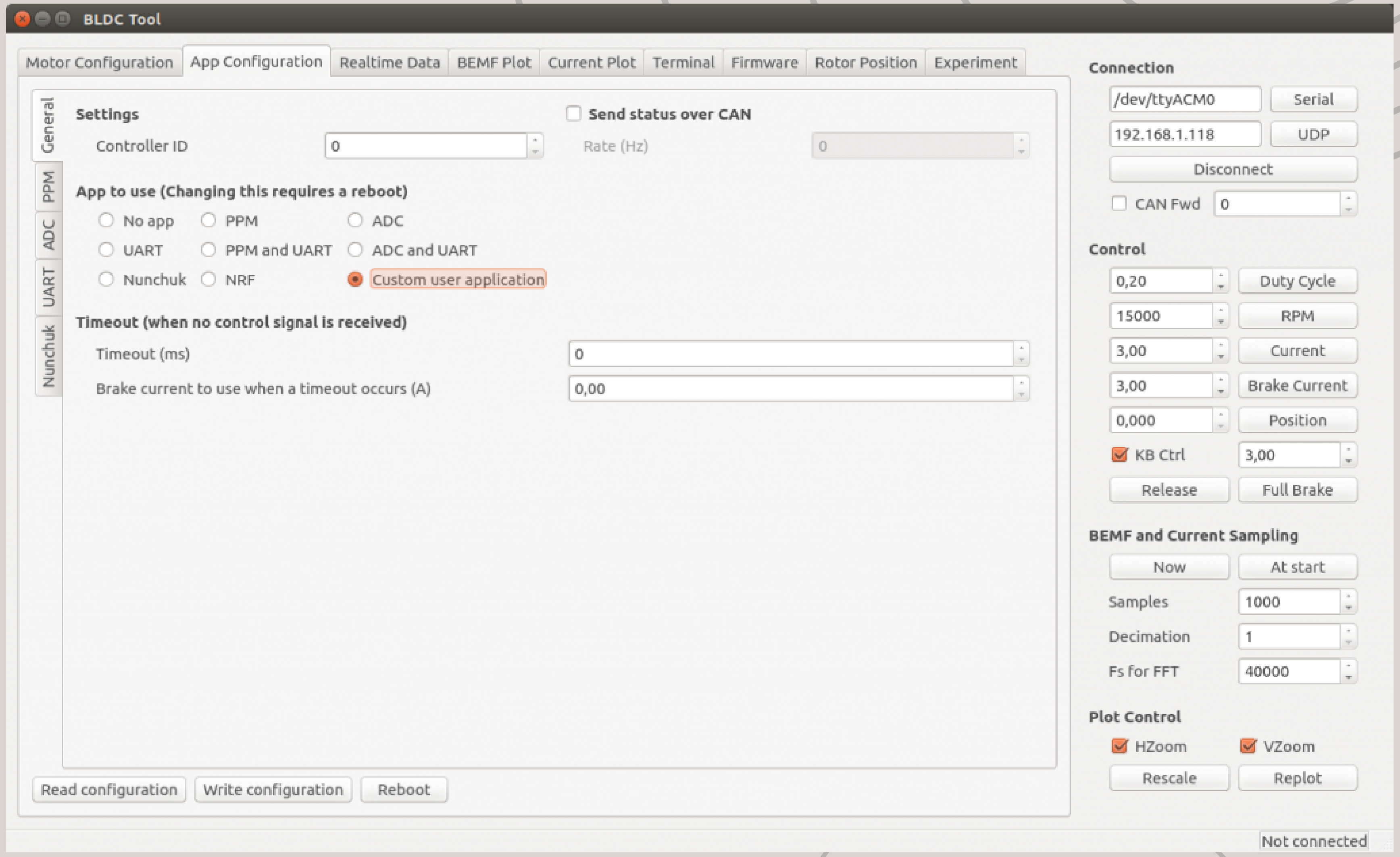
Fs for FFT 40000

Plot Control

☑ Horizontal Zoom

☑ Vertical Zoom

Not connected



Settings

Controller ID 0

☐ **Send status over CAN**

Rate (Hz) 0

App to use (Changing this requires a reboot)

○ No app ○ PPM ○ ADC

○ UART ○ PPM and UART ○ ADC and UART

○ Nunchuk ○ NRF ● Custom user application

Timeout (when no control signal is received)

Timeout (ms) 0

Brake current to use when a timeout occurs (A) 0,00

Connection

/dev/ttyACM0

192.168.1.118

☐ CAN Fwd 0

Control

0,20 Duty Cycle

15000 RPM

3,00 Current

3,00 Brake Current

0,000 Position

☑ KB Ctrl 3,00

BEMF and Current Sampling

Samples 1000

Decimation 1

Fs for FFT 40000

Plot Control

☑ HZoom ☑ VZoom

Not connected

Interfaces

The VESC project provides multiple interfaces for controlling the motor and configuring the system.

- Pulse Position Modulation (PPM) Signal;
- Analog Input;
- UART (Universal Asynchronous Receiver/Transmitter);
- USB;
- I2C (Inter-Integrated Circuit);
- CAN-bus.

Protection Mechanisms

The VESC project incorporates a range of protection mechanisms designed to ensure the safety, reliability, and longevity of the motor control system and its motors.

- **Voltage Protection:** Low and High voltage protection
- **Current Protection:** Motor Current Limiting, Input Current Limiting, Regenerative Braking Current Limiting.
- **Duty Cycle and RPM Protection:** Rapid Duty Cycle Changes (Ramping), High RPM Limits.
- **Thermal Protection**

Other features

- Regenerative braking - converts kinetic energy back into electrical energy during deceleration.
- Seamless 4-Quadrant Operation
- The motor as a tachometer - continuously monitors the rotational speed to perform odometry on a vehicle.

Other features

- Integrated energy counting (Consumed and regenerated Amp-Hour and Watt-Hour Counting)
- Advanced commutation techniques:
    - Perfect Commutation with Rapid Speed Changes (integration of the magnetic flux)
    - Commutation with Motor Rotation When the Controller is Off (calculation of the required duty cycle to match the motor’s speed)