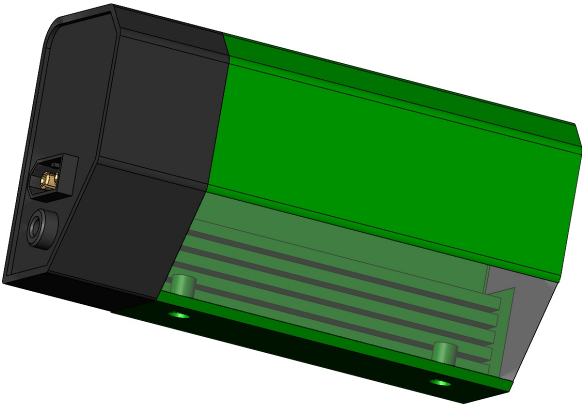
The Frankenrunner Motor Controller

User Manual – *Rev 1.01*



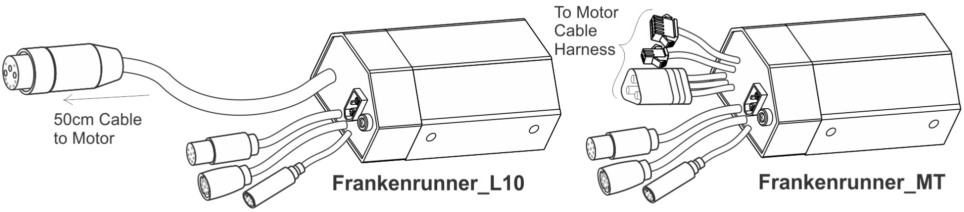
Grin Technologies Ltd
Vancouver, BC, Canada

ph: (604) 569-0902
email: info@ebikes.ca
web: www.ebikes.ca

Table of Contents

1 Introduction

Thank you for purchasing the *Frankenrunner*, a temporary alternative to the *Phaseunner* motor controller based on Accelerated Systems Inc. (ASI)'s BAC800 device.



This manual covers two production models: the *Frankenrunner_L10* and the *Frankenrunner_MT*. The **L10** model is intended for motors already using the L1019 connector, while the **MT** model employs a cable harness between the controller and motor for accommodating various other connector standards.

Features of the *Frankenrunner* include:

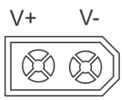
- Compact form factor
- User programmable parameters for customized tuning
- Wide operating voltage (24V - 72V nominal batteries)
- Compatible with both Cycle Analyst and 3rd party displays
- Supports Throttle, PAS and Torque sensor control
- Waterproof design with potted electronics
- Proportional and powerful regenerative braking
- Smooth and quiet field-oriented drive
- Supports thermal rollback in thermistor-equipped motors
- Remote forwards/reverse input
- Field weakening to boost top speed
- Sensorless operation with high eRPM motors

Unlike standard trapezoidal or sine wave controllers, the Frankenrunner is a field oriented controller that must be tuned to your motor, battery, and performance requirements for proper operation. This process is detailed in Section 5 Parameter Tuning .

2 Connectors

The controller has been connectorized to achieve maximum versatility with minimal wiring, using a combination of waterproof overmolded ebike plugs for signals and popular compact connectors for high current.

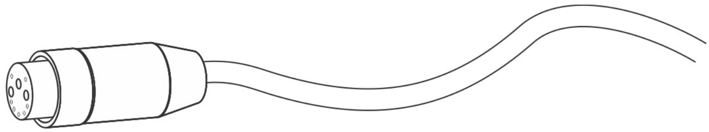
2.1 *Battery Power*

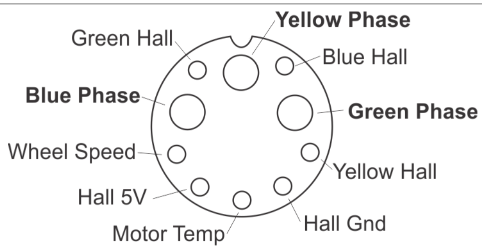
 **DC Power Pinout**	The embedded plug on the back of the device provides a reliable hookup to DC battery power through the compact and popular XT60 plug interface. A 30cm XT60 to Anderson cable is provided with each device to match the Anderson standard used on many ebike batteries.

- Warning – The controller may suffer irreparable damage if leads are hooked up in reverse polarity. Always check the polarity of the connector before applying power.

2.2 *Motor Cable – L10 Model*

The L10 model has a 40cm lead to a Higo L1019 connector for direct hookup to a compatible motor. This length is sufficient to reach a rear hub motor with the controller mounted on the seat-tube, or to a front hub motor with the controller on the stem. Both 60cm and 100cm L10 extension cables are available when longer lengths are required.



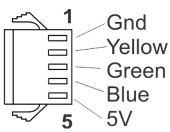
 **L10 Motor Plug Pinout**	The Higo L1019 cable has three motor phase pins capable of 80 amps peak, along with 7 small signal wires for Hall position, speed encoder, and motor temperature.

2.3 Motor Cable – MT Model

The motor connection on the MT model uses an MT60 plug for the 3 phase power, and both 5 pin and 2 pin JST-SM plugs for Hall sensor and temp/speed signals. These leads coming from the controller are short, with an expectation that a separate cable harness will be employed to reach the motor terminals.

Motor Harness Phase Pinout

The MT60 connector supports 3 phase power in a single plug. These connectors can be readily soldered for making custom harnesses.



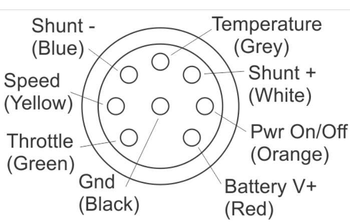
Motor Harness Hall Pinout

The 3 Hall signals and 5V Hall power are provided through the 5 pin female JST-SM connector. This is a user crimpable plug.

Motor Harness Signal Pinout

If the motor has a temperature and/or wheel speed sensor, these signals can be passed to the controller via the 2-pin plug. Both of these signals must be referenced to the Hall ground pin.

2.4 Cycle Analyst WP8 Plug

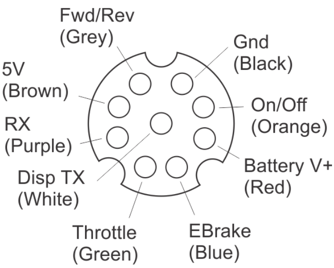


WP8 Pinout

The connector for the Cycle Analyst cable uses the waterproof 8-pin Z812 Higo standard.

This connector taps into the controller's shunt resistor for analog current and power sensing, passing through the motor's speed and temperature signals as well. It commands the controller via the throttle line.

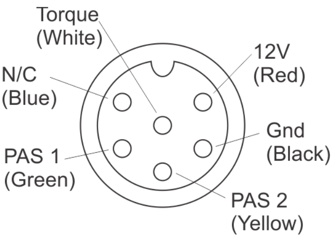
2.5 Mains Signals Plug



Mains Pinout

An alternate interface is provided via the 9 pin Mains cable. This uses uses the Signal D 1109 Connector from Cusmade, and supports conventional ebike wiring strategies for 3rd party display consoles. It shares many signals with the CA-WP plug, but rather than using the shunt resistor for current sensing, it has TX and RX pins that communicate digitally to the display.

2.6 PAS / Torque Plug



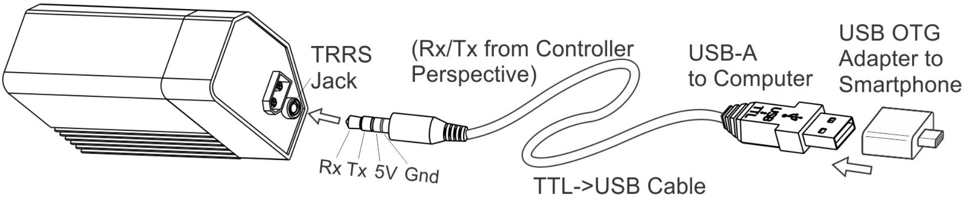
PAS Pinout

Finally there is a 6 pin HiGo MiniB Z609 plug for connection of a PAS sensor or Torque Sensor.

Note that the **PAS 2** pin shares the same signal as the **Fwd/Rev** input of the Mains cable, and can be configured for either function. (See section 6.2)

2.7 Communication Port

A TRRS jack embedded in the controller can be used for connecting to a computer, Android smart phone, or potential Bluetooth dongle (future product).



The communication standard uses a 0 to 5V level serial bus. Grin produces a 3m long TTL->USB adapter cable to connect the unit with the USB port of a standard computer. This is the same communication cable used with the *Cycle Analyst* and *Satiator* products. Third party USB->Serial cables, such as FTDI's part number TTL-232R-5V-AJ are also compatible.

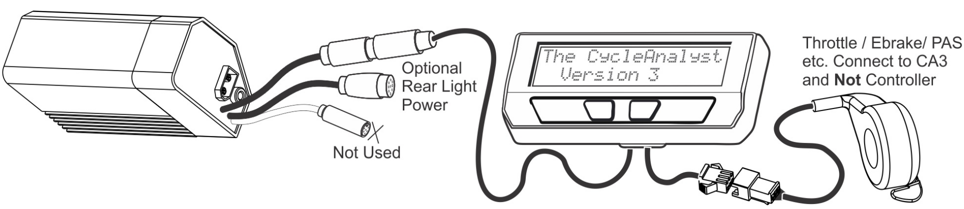
An additional USB-OTG adapter is needed when connecting to an Android smartphone via the phone's smaller Micro USB or USB-C port.

3 Wiring Strategies

The *Frankenrunner* can be hooked up to the controls of an ebike system in one of three ways. It can be connected under the full control of a V3 Cycle Analyst, under the control of a 3rd party display, or “headless” with no display at all.

3.1 Cycle Analyst Based Hookup

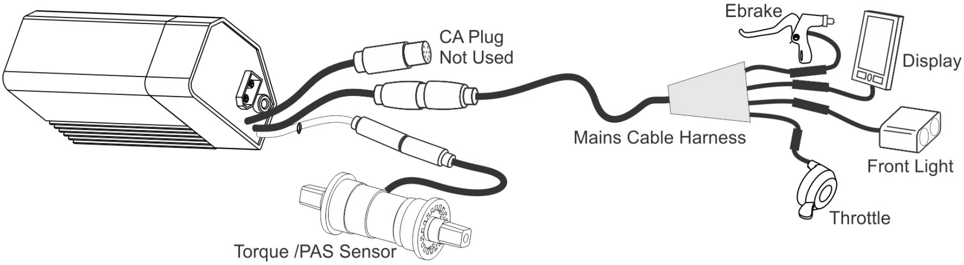
The setup using the latest V3 Cycle Analyst (CA3-WP) provides the most versatility with mode presets, customizable PAS behavior, advanced regen features, and easy performance adjustments on the road. In this arrangement, all throttle, ebrake, and PAS or torque sensors are plugged in directly to the Cycle Analyst. The Cycle Analyst is responsible for determining the desired ebike behavior and sending a suitable throttle command to the controller.



The 6 pin PAS plug of the controller is typically not used in this arrangement except as a possible Fwd/Rev input source. The 9 pin Mains cable either can be left unterminated, or it can be used as a power tap for running rear lights, using a 9 to 2 pin Power Adapter cable.

3.2 3rd Party Display Hookup

The *Frankenrunner* can be used with certain third party displays (from King Meter, Star-Union, Bafang etc.) that communicate using the KM5s digital protocol. This is achieved with a 9 pin Mains cable harness which splits out the 9 signals into separate plugs for throttle, display, ebrakes, and front light, along with an (optional) PAS or Torque sensor hooked up to the 6 pin PAS plug as well.



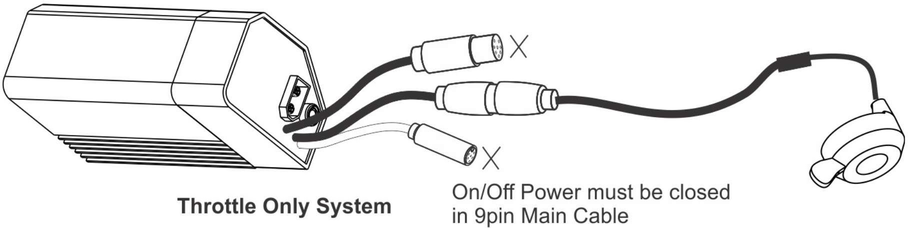
All the key control signals (throttle, ebrake, PAS/Torque) go directly to the *Frankenrunner* and not the display, so additional controller parameters must be set up to achieve desired control response. The display has the on/off power control and up/down settings to set the assist level, but otherwise plays a passive role in the actual system behavior.

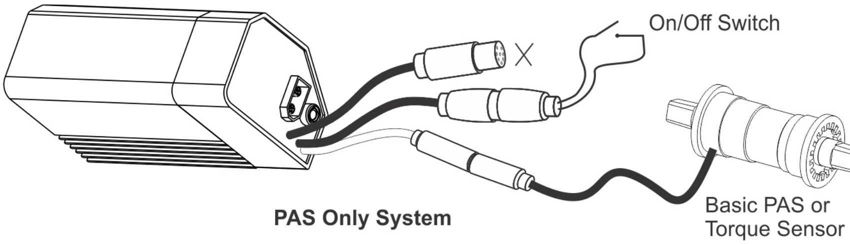
In this wiring approach, the WP8 Cycle Analyst plug is not needed, but it can be used as a convenient tap point to power a rear bike light as well.

At present Grin only provides support for this third-party display hookup to OEM customers, and does not offer support or the components for this at the retail level. Configuration and additional controller settings will be detailed in a separate document to this manual.

3.3 Headless System

Finally, the *Frankenrunner* can be run with only a throttle on the Mains plug, or a PAS / Torque sensor plugged into the 6 pin PAS plug. In these arrangements, it is essential to wire up the on/off power switch on either the WP8 plug or the Mains connector for the controller to turn on.

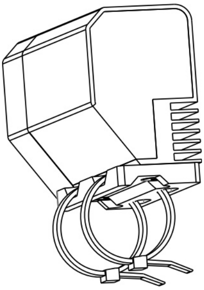




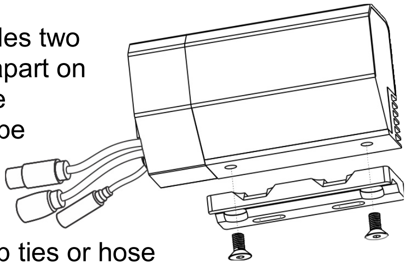
This minimal approach with a PAS sensor does not provide for any means to adjust the assist level outside of reprogramming the controller, and for that reason has limited usefulness. A display unit with up/down buttons is required in order to vary the amount of assist while riding.

4 Controller Mounting

For optimal performance, the *Frankenrunner* should be installed such that the finned heatsink is exposed to airflow to keep the controller cool. This placement will noticeably improve the maximum power at thermal rollback compared to a controller that is in still air.



The controller casing includes two threaded M5 holes 60mm apart on the narrow edge to facilitate installation. Grin offers a tube mounting bracket that converts this into a channel that securely attaches to bicycle tubing with either zip ties or hose clamps.



5 Parameter Tuning

If you purchased the *Frankenrunner* as part of a complete conversion kit that includes a battery, a motor, and so on, the controller should be pre-configured and no tuning of the parameters should be necessary. This section can be skipped entirely.

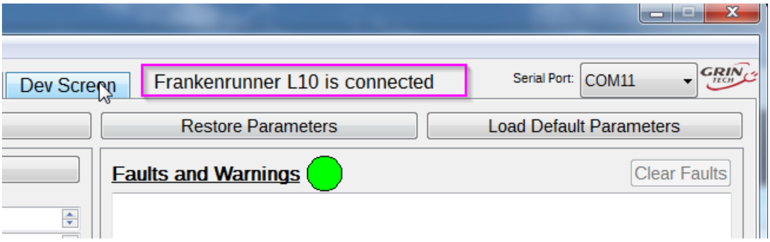
If you bought the *Frankenrunner* separately, or are changing your set-up, you should configure the controller to your motor and battery pack once it is installed and connected on your bike. You will need a computer, a TTL-USB programming cable and the **V1.7 or later** *Phaserunner Software Suite*. The V1.6 and earlier software releases will give an "unrecognized device" error message.

This software is available for Linux, Windows, MacOS and Android from our webpage:

http://www.ebikes.ca/product-info/phaserunner.html

Please Note: *When configuring your Frankenrunner via the software suite, it is essential that your bike is propped up so that the powered wheel can rotate freely, both forwards and backwards. With a rear hub motor, also ensure that the cranks can rotate freely.*

With the *Frankenrunner* powered on, plug in the TTL->USB cable from your computer to the device. After launching the *Phaserunner* software, select the COM port associated with the USB cable and you should see "*Frankenrunner is connected*" on the top.



If you see "*Controller is not connected*," check that the selected serial port is correct and that the USB->TTL device shows up in your device manager as a COM port (Windows), ttyUSB (Linux), or cu.usbserial (MacOS).

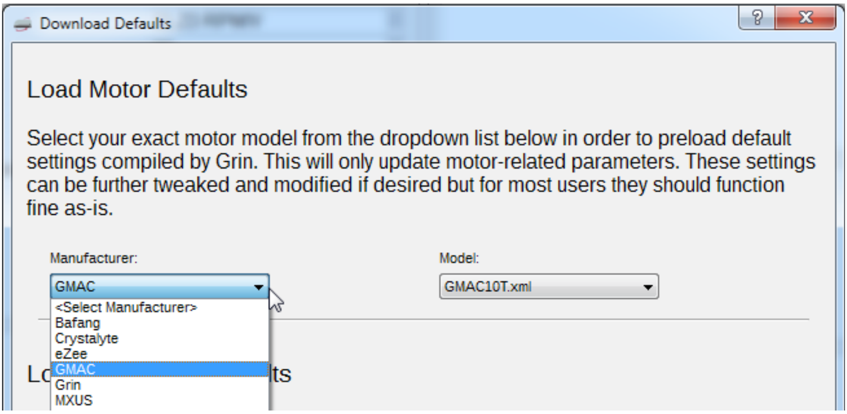
If your system does not recognize the USB serial adapter, or has frequent COM timeouts, then you may need to download and install the latest virtual COM port drivers directly from FTDI:

http://www.ftdichip.com/Drivers/VCP.htm

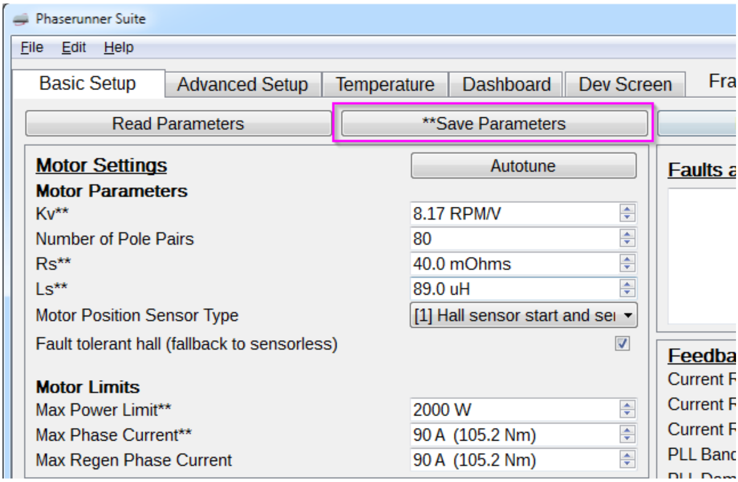
During the setup process we recommend having the controller powered by a battery pack and not a general power supply, as electrical noise from power supplies can interfere with stable communication.

5.1 Importing Default Motor Parameters

The *Phaserunner Software Suite* comes equipped with default settings for many common motors. With your *Frankenrunner* connected, click on "Load Default Parameters" and select your motor's manufacturer and model number from the new window. Clicking on "Apply" will return you to the "Basic Setup" tab with all the motor's parameter-fields populated to their correct values.



Install these new settings to the *Frankenrunner* via the “Save Parameters” button.



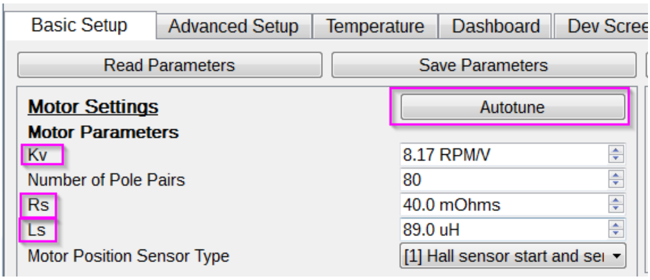
Apply some throttle and your motor should run smoothly. If it does, you can now skip over the “Motor Autotune” section, and continue with “Battery Limits.”

If your motor is not listed on the “Import Defaults” window, try choosing “Download Latest Defaults from Grin” and follow the prompts. If your motor model is still not listed, proceed to the “Motor Autotune” section that follows.

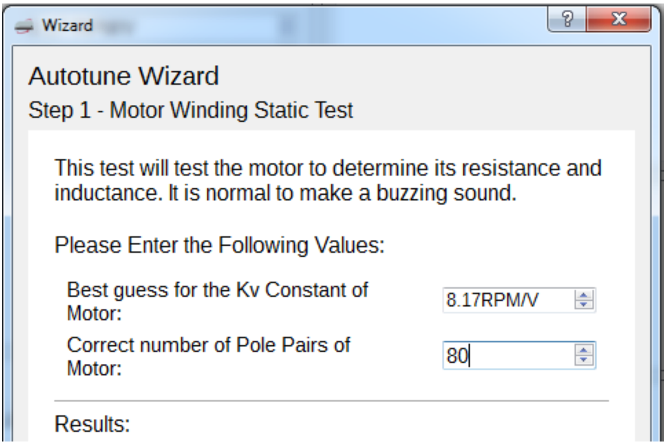
5.2 Motor Autotune

Basic Setup tab

The Autotune routine can automatically detect motor parameters like the motor speed constant (kV), resistance of one motor phase to neutral (Rs), and the inductance of motor phase to neutral (Ls).



The start of the Autotune process asks for your best guess of the motor's kV in rpm/V, as well as the number of pole pairs in the motor. The firmware uses these initial parameters for determining the test current frequency.



The Autotune routine will usually work fine even if your initial guess for the kV value is incorrect. Most ebike hub motors fall within 7-12 rpm/V and an initial guess of 10 will usually work.

The effective pole pairs is a count of how many electrical cycles corresponds to one mechanical revolution of the motor and must be set correctly. The *Frankenrunner* needs this information to correlate it's electrical output frequency with the wheel speed. In a direct drive (DD) motor, it is the number of magnet pairs in the rotor, while in a geared motor you need to multiply the magnet pairs by its gear ratio. The following table lists the effective pole pairs for many common motor series.

Table 1: Effective Pole Pairs of Common DD and Geared Hub Motors

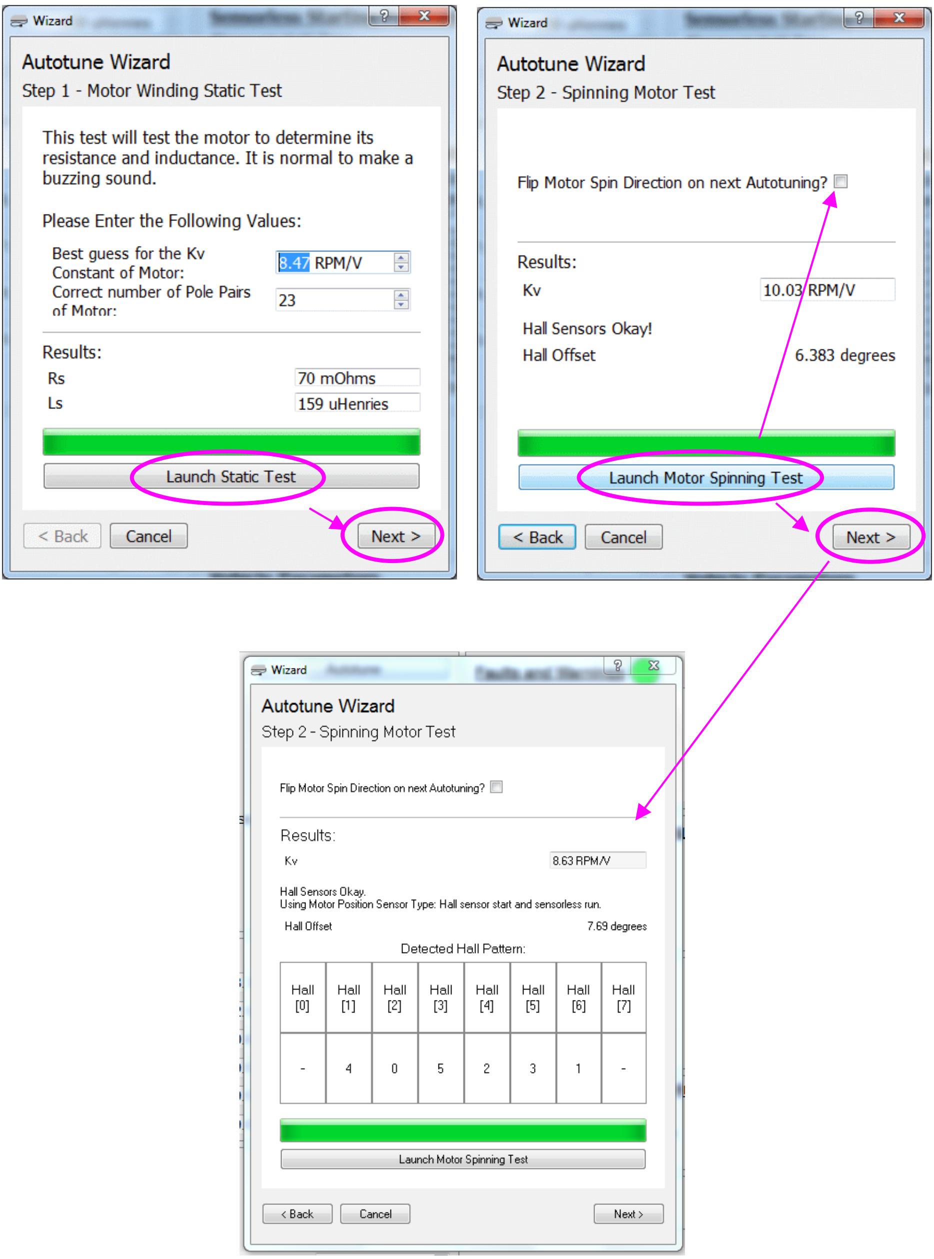
Motor Family	# Pole Pairs
Crystalyte 400, Wilderness Energy	8
BionX PL350	11
Crystalyte 5300, 5400	12
TDCM IGH	16
Crysatlyte NSM, SAW	20
Grin All Axle, Crysatlyte H, Nine Continent, MXUS and Other 205mm DD Motors	23
Magic Pie 3, Other 273mm DD Motors, RH212	26
Bafang BPM, Bafang CST	40
Bafang G01, MXUS XF07	44
Bafang G02, G60, G62	50
Shengyi SX1/SX2	72
eZee, BMC, MAC, Puma, GMAC	80
Bafang G310, G311	88
Bafang G370	112

For motors not listed, either open the motor to count the magnets pairs (and gear ratio), or count the number of Hall cycles that take place when you manually turn the wheel one revolution. You can monitor the number of Hall transitions via the “Dashboard” tab of the software suite.

Once the “kV” and the “Number of Pole Pair” values are entered, launch the “Static Test.” This test will produce three short buzzing sounds, and determine the inductance and resistance of the motor windings. The resulting values will be shown on the screen.

Next, launch the “Spinning Motor Test” which will cause the motor to rotate at about half speed for 15 seconds. During this test, the controller will determine the actual kV winding constant for the hub, as well as the pinout and timing advance of the Hall sensors if they are present. If the motor spins backwards during this test, check the box “Flip Motor Spin Direction on Next Autotuning?” and relaunch the “Spinning Motor Test.”

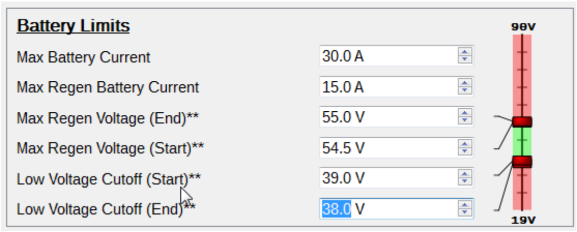
During the spinning test, the *Frankenrunner* will start the motor in sensorless mode. If the motor fails to spin and just starts and stutters a few times, adjust the sensorless starting parameters as described in section 5.5, “Tuning the Sensorless Self Start,” until the motor is spinning steadily. If the spinning test detects a valid Hall sequence, the final screen will show the hall offset, and that the “Position Sensor Type” is “Hall sensor start and sensorless run.”


Wizard
Autotune Wizard
Step 1 - Motor Winding Static Test
This test will test the motor to determine its resistance and inductance. It is normal to make a buzzing sound.
Please Enter the Following Values:
Best guess for the Kv Constant of Motor:
8.47 RPM/V
Correct number of Pole Pairs of Motor:
23
Results:
Rs
70 mOhms
Ls
159 uHenries
Launch Static Test
< Back
Cancel
Next >
Wizard
Autotune Wizard
Step 2 - Spinning Motor Test
Flip Motor Spin Direction on next Autotuning?
Results:
Kv
10.03 RPM/V
Hall Sensors Okay!
Hall Offset
6.383 degrees
Launch Motor Spinning Test
< Back
Cancel
Next >
Wizard
Autotune Wizard
Step 2 - Spinning Motor Test
Flip Motor Spin Direction on next Autotuning?
Results:
Kv
8.63 RPM/V
Hall Sensors Okay.
Using Motor Position Sensor Type: Hall sensor start and sensorless run.
Hall Offset
7.69 degrees
Detected Hall Pattern:
Hall [0] Hall [1] Hall [2] Hall [3] Hall [4] Hall [5] Hall [6] Hall [7]
- 4 0 5 2 3 1 -
Launch Motor Spinning Test
< Back
Cancel
Next >

5.3 Battery Limits

Basic Setup tab

With the controller mapped to your motor and spinning correctly, you should now set the battery voltage and current settings to appropriate values for your pack.



Set “Max Battery Current” to a value that is equal to or less than the battery’s discharge current rating. Higher battery currents will result in more power, but can also stress the battery cells, resulting in shorter battery life. Excessively high values can cause the BMS circuit to trip, shutting down the pack.

We recommend setting “Max Regen Voltage (Start)” to the same value as the full charge voltage of your battery, with the “Max Regen Voltage (End)” to about 0.5V higher than full charge. This will ensure you can do regen even with a mostly charged battery.

The “Low Voltage Cutoff (Start)” and “Low Voltage Cutoff (End)” values can be set just above the BMS cutoff point of your battery. If you are using a V3 Cycle Analyst, we recommend leaving these values at the default 19.5/19.0 volts and use the CA3’s low voltage cutoff feature instead. That way you can change the cutoff voltage without a computer if you ever need to swap batteries.

If you are setting up a system with regenerative braking and have a BMS circuit that shuts off if it detects excessive charge current, you will also need to further limit the “Maximum Regen Battery Current” to a value lower than the BMS trip current. Otherwise it's not usually an issue with modern lithium cells to have short duration peak regen current levels of 2C, ie double the battery amp-hours.

5.4 Motor Phase Current and Power Settings

Basic Setup tab

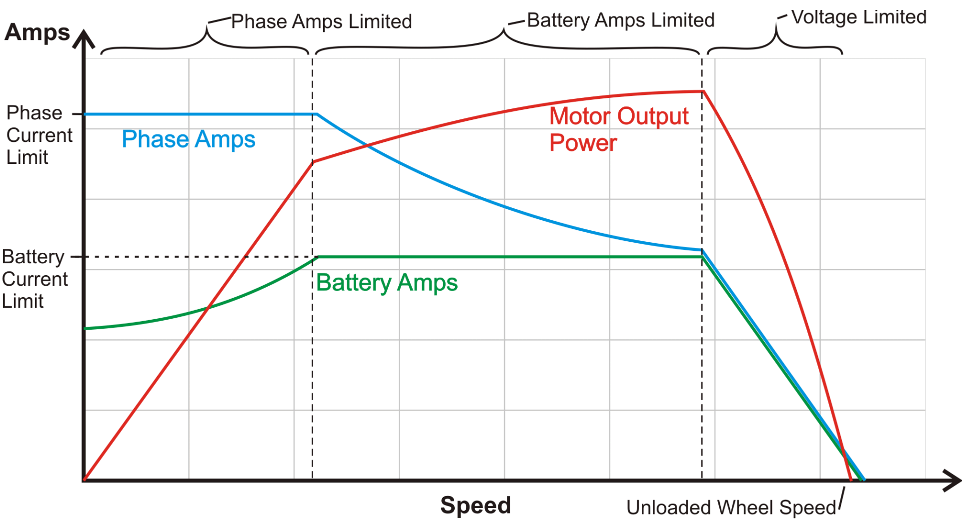
In addition to regulating the current flowing in and out of the battery pack, the *Frankenrunner* can independently control the maximum phase currents that flow to and from the motor. It is the motor phase current that both generates torque and causes the motor windings to heat up. At low motor speeds this phase current can be several times higher than the battery current you see on a *Cycle Analyst*.

The “Max Power Limit” sets an upper limit on the total watts that will be allowed to flow into the hub motor. This value has a similar effect to a battery current limit, but it is dependent on voltage. A value of 2000 Watts will limit battery current to 27 amps with a 72V pack, while allowing over 40 amps with a 48V pack.

“Max Phase Current” determines the peak amps, and hence torque, put through the motor while accelerating at full throttle assuming no other limits are reached.

The “Max Regen Phase Current” value directly sets the peak braking torque of the motor at full regen. If you want a strong braking effect, then set this to the same value as the forward phase current limit. If the maximum braking force is too intense, then reduce its value.

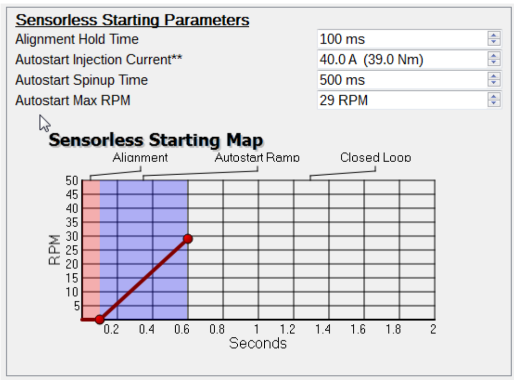
The following graph illustrates the interplay between motor phase current, battery current, and motor output power for a typical setup. When riding at full throttle, low speeds will be phase current limited, medium speeds will be battery current limited, and high speeds will be limited by the voltage of your battery pack.



5.5 Tuning the Sensorless Self Start

Advanced Setup

If you are running in sensorless mode, then you may need to tweak the sensorless self start behaviour.



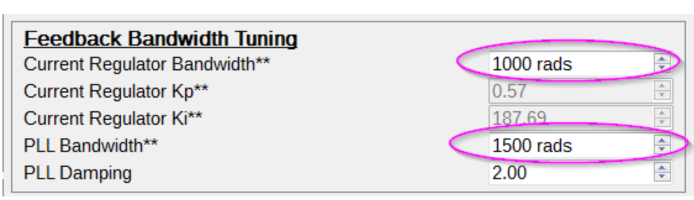
When a brushless motor is run without Hall sensors and started from a complete stop, the motor controller attempts to ramp up the motor's rpm to a minimum speed so that it can latch onto the rotation (closed loop).

It does this by first injecting a static current into the phase windings to orient the motor into a known position. The controller then rotates this field faster and faster until reaching the "Autostart Max RPM" value.

As initial values, set the "Autostart Injection Current" to half your maximum phase current, an "Autostart Max RPM" between 5% to 10% of the running motor rpm, and an "Autostart Spinup Time" anywhere from 300 to 1500 milliseconds, depending on how easily the motor can propel the bike up to speed.

On bikes that you pedal to help get you underway, a short 200 to 300 millisecond ramp will often work best, while a much longer ramp is required if you need to start moving without pedaling.

If you feel the motor repeatedly trying to start when applying throttle, the "Autostart Spinup Time" may be too short, or the "Autostart Max RPM" may be too low. You may also generate faults such as "Instantaneous Phase Overcurrent." To correct this particular fault, try increasing either the "Current Regulator Bandwidth" or the "PLL Bandwidth" parameters, or both. These parameters are found under "Feedback Bandwidth Tuning" on the "Basic Setup" tab.



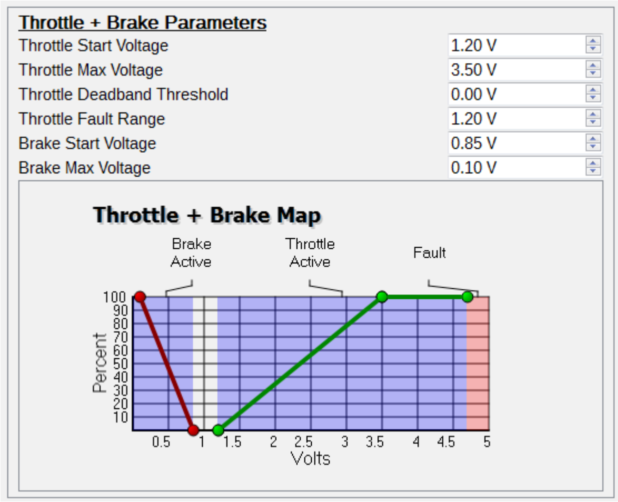
5.6 Throttle and Regen Voltage Maps

Advanced Setup tab

With most ebike controllers, the throttle signal controls the effective voltage and hence unloaded rpm of the motor. With a *Frankenrunner*, however, the throttle is directly controlling the motor torque.

If you pick the motor off the ground and give it just a tiny amount of throttle, it will still spin up to full rpm as there is no load on the motor. People sometimes mistake this behavior as an all-or-nothing throttle response. If you apply partial throttle *while riding*, you will get a proportional torque from the motor which will stay constant even as the vehicle speeds up or slows down. This is different from standard ebike controllers, where the throttle more directly controls motor speed.

By default, the *Frankenrunner* is configured so that active throttle starts at 1.2V, and full throttle is reached at 3.5V, which is broadly compatible with Hall Effect ebike throttles. The same signal for the throttle is also used to control regenerative braking (see Section 6.1 Signal Mapping).



The regen voltage is mapped by default so that regenerative braking starts at 0.8V and reaches maximum intensity at 0.0V. This way there is no overlap between the throttle region and the braking region and a single wire can control both ranges.

5.7 Field Weakening for Speed Boost

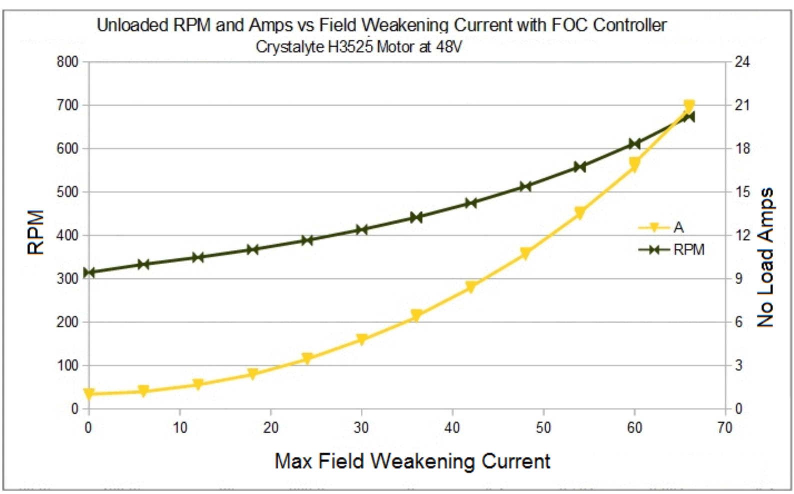
Basic Setup tab

The *Frankenrunner* can boost the top speed of your motor beyond what is normally possible from your battery voltage. This is accomplished by injecting a field weakening current in advance of the torque producing current.

The amount of speed boost received for a given field weakening current will depend on the winding characteristics of your particular motor. A trial and error approach is recommended, increasing in small increments until the desired top speed is achieved.

Increasing a motor's top speed in this way is less efficient than using a higher voltage pack or a faster motor winding, but for a speed boost of 15% to 20%, the additional losses are quite reasonable.

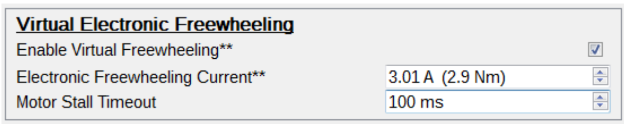
The following graph shows a large direct drive hub motor's rpm as a function of field weakening current. The upper black line is the motor's measured rpm, while the initially lower yellow line is the no-load current draw, reflecting the amount of extra power lost due to field weakening. We can see that at 20 amps of field weakening, the motor speed increases from 310 rpm to 380 rpm, while the no load current draw is still just under 3 amps.



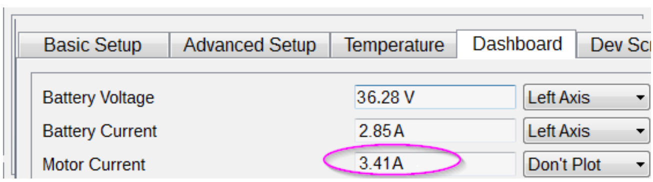
5.8 Virtual Electronic Freewheeling

Dashboard/Basic Setup tabs

The *Frankenrunner* controller can be set to inject a small amount of current into the motor, even when the throttle is off. When properly tuned, this current injection can overcome the drag torque present in hub motors capable of regenerative braking, allowing them to spin freely when pedaling without any throttle.



To setup this feature, we recommend first going to the “Dashboard” tab. With the system running under throttle with no load on the motor, note the “Motor Current” value.



Navigate back to the “Basic Setup” tab, check “Enable Virtual Freewheeling,” and set “Electronic Freewheeling Current” to a value slightly less than that of the observed unloaded motor current. The “Motor Stall Timeout” setting determines when this injection current will stop once the motor comes to a stop.

Once the values for “Virtual Electronic Freewheeling” are set, the controller will draw about 10 to 40 watts in order to overcome the motor’s drag. Regenerative braking should recapture more energy than lost due to the injection current.

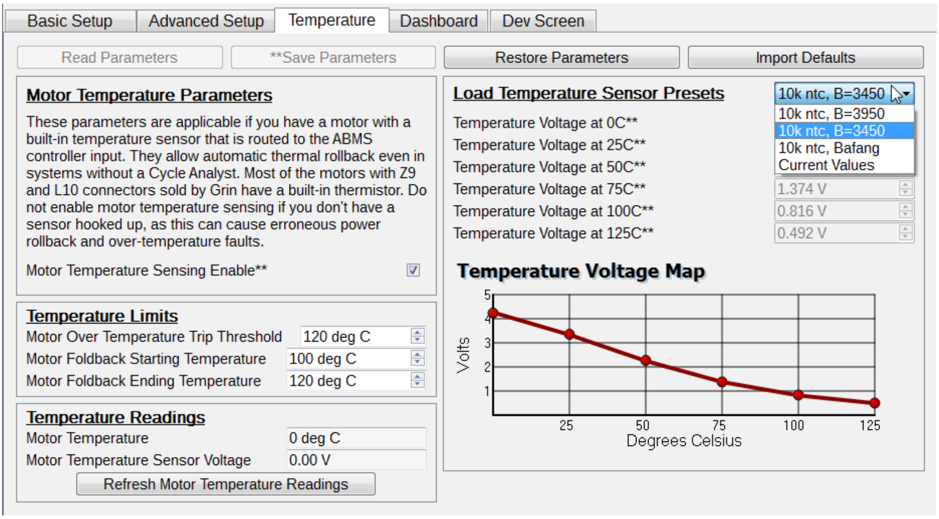
Users of mid-drive motors can also use this feature to keep the drive train always engaged, eliminating windup delay and harsh clutch engagement when throttle is applied and the motor comes up to speed.

5.9 Motor Temperature Sensing

Temperature Setup tab

The *Frankenrunner* has the ability to sense motor temperature and prevent the motor from overheating via thermal rollback. An entire tab in the software suite is dedicated to the motor thermal rollback settings. A six point table maps the voltage that corresponds to the temperatures of 0°, 25°, 50°, 75°, 100°, and

125°C. For convenience we have provided a drop-down selection of the three thermistor types commonly used on hub motors. For other temperature devices, these values can also be input manually instead.



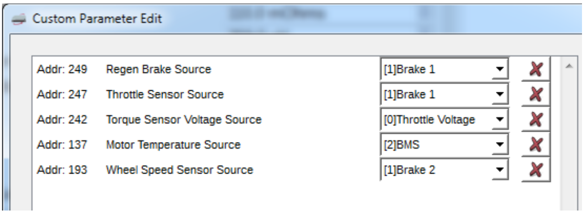
This feature allows for safe operation of motors even without a V3 Cycle Analyst. If you have a Cycle Analyst we recommend using the CA3's thermal rollback features and leaving the controller's rollback unchecked.

6 Additional Details:

6.1 Signal Mapping

The *Frankenrunner* uses the analog Brake 1 signal input as both the throttle and regenerative brake signal source. This differs from the earlier Phaserunner controllers which used separate signals for "Throttle Sensor Source" and "Regen Brake Source", which were then shorted externally in the connector wiring.

The nominal 'throttle' input of the controller is now mapped as a torque sensor input, while inputs that were previously unused are now being used.



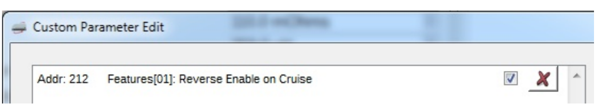
These settings can be seen by going edit->edit parameters as shown above. The difference between the V3 Phaserunner devices and *Frankenrunner* are summarized in the table below:

Input Source	**Phaserunner (V3)**	**Frankenrunner**
Throttle Source	Analog Input 1 (Thrt)	Analog Input 2 (Brk1)
Regen Brake Source	Analog Input 3 (Brk2)	Analog Input 2 (Brk1)
Torque Sensor Source	N/A	Analog Input 1 (Thrt)
Motor Temperature	N/A	Analog Input 4 (BMS)
PAS	N/A	Digital Input 2 (PFS)
Wheel Speed Sensor	Hall A	Analog Input 3 (Brk2)
Fwd/Rev	Digital Input 1 (Cruise)	Digital Input 1 (Cruise)

These changes were necessary to fully support the independent use of torque sensors and throttles on the same controller. As a result if full parameter settings saved from a Phaserunner device are then imported to the *Frankenrunner* or vice versa, unexpected throttle and braking behavior can occur.

6.2 Reverse Mode

The signal *PAS 2* used in the 6 pin PAS plug is electrically equivalent to the *FWD/REV* pin in the Main plug. This input is by default configured as a reverse switch input by enabling "Reverse on Cruise" for applications that require direction reversal. When using quadrature (2 wire) PAS sensors plugged into the PAS plug, this must be unchecked.

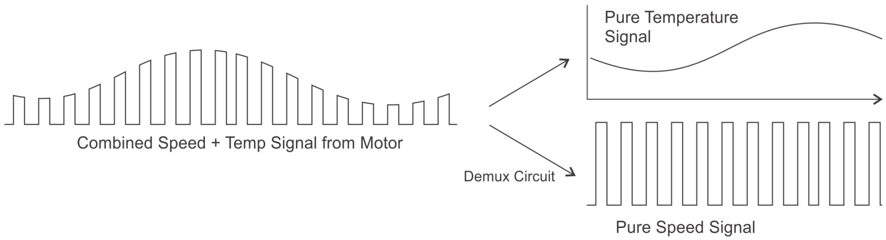


6.3 Wheel Speed Sensing

The *Frankenrunner* will automatically select the source of the wheel speed signal for vehicle speed measurement. If there are speed pulses present on the wheel speed sensor pin then these will be mapped automatically to the Cycle Analyst plug. If no speed pulses are detected even after the motor is spinning, then the motor Hall signals will get fed to the speed signal input instead.

6.4 Combined Temp / Speed Signal

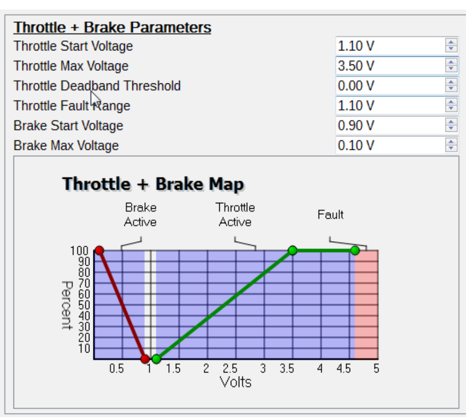
The *Frankenrunner* can also work with combined temperature and speed signals that are present on the temperature input pin. If there are no speed pulses present on the Wheel Speed signal, and the temperature signal periodically drops to 0V, then the *Frankenrunner* will treat those 0V pulses as speed signals to track wheel rotation.



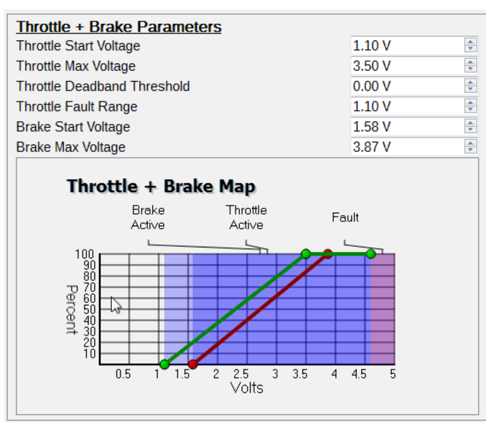
6.5 Independent Regenerative Braking

The *Frankenrunner* by default has both the throttle signal source and the regen brake signal source coming from the same line, Analog Input 2, which floats at 1V when disconnected. This allows for a single wire to control power and braking. The signal can be shorted to ground via an ebrake lever to achieve maximum regen braking.

If completely independent throttle and regen brake signals are desired, then the throttle signal source can be changed to Analog Input 1. Separate wires can be used to control braking versus throttling, which then excludes the possibility of having a torque sensor input.



Throttle + Brake Map for Combined Signal



Example for Independent Throttle and Brake

7 Cycle Analyst Settings

Current Sensing [Cal->RShunt]

The *Frankenrunner* uses a 1.00 mΩ +- 0.02 mΩ shunt resistor for current sensing. The exact calibrated value is laser engraved on the controller heatsink. In order to have the most accurate readout of battery current, ensure that the *Cycle Analyst's* "RShunt" value is set to match this.

Throttle Out [ThrO->Up/Down Rate] [SLim->Int,D,PSGain]

Because the *Frankenrunner* uses a torque throttle rather than a voltage throttle, the entire throttle voltage range is always active. Optimal settings for the throttle output on a V3 Cycle Analyst will differ than that for generic ebike controllers.

The ramp up and ramp down rates as well as the feedback gain settings (AGain, WGain, IntSGain, DSGain, PSGain) can be set much higher than with a conventional controller with a voltage throttle. These will need to be tweaked for your particular motor, but good starting values are:

- Ramp Up: 6 V/sec
- WGain: 50-100
- AGain: 300-400
- IntSGain: 100-150
- DSGain: 600-800
- PSGain: 2-3 V/kph

8 LED Flash Codes

The embedded red LED inside the controller provides a useful status indicator. It will flash according to the following table if the controller detects any faults. Some faults will clear automatically once the condition clears, such as "Throttle Voltage Outside of Range," while other faults may require turning the controller off and on.

Table 2: LED Fault Flash Codes

1-1	Controller Over Voltage
1-2	Phase Over Current
1-3	Current Sensor Calibration
1-4	Current Sensor Over Current
1-5	Controller Over Temperature
1-6	Motor Hall Sensor Fault
1-7	Controller Under Voltage
1-8	POST Static Gate Test Outside Range
2-1	Network Communications Timeout
2-2	Instantaneous Phase Over Current
2-3	Motor Over Temperature
2-4	Throttle Voltage Outside of Range
2-5	Instantaneous Controller Over Voltage
2-6	Internal Error
2-7	POST Dynamic Gate Test Outside Range
2-8	Instantaneous Controller Under Voltage
3-1	Parameter CRC Error
3-2	Current Scaling Error
3-3	Voltage Scaling Error
3-4	Headlight Under Voltage
3-5	Torque Sensor
3-6	CAN Bus
3-7	Hall Stall
4-1	Parameter2CRC

The LED may also flash several different warning codes. These warnings do not stop the controller from running and will appear as various limits are reached in normal operation, they are not usually a cause for any concern.

Table 3: LED Warning Flash Codes

5-1	Communication Timeout
5-2	Hall Sensor
5-3	Hall Stall
5-4	Wheel Speed Sensor
5-5	CAN Bus
5-6	Hall Illegal Sector
5-7	Hall Illegal Transition
5-8	Low Voltage Rollback Active
6-1	Max Regen Voltage Rollback Active
6-2	Motor Overtemperature Rollback
6-3	Controller Overtemperature Rollback
6-4	Low SOC Foldback
6-5	Hi SOC Foldback
6-6	I2tFLDBK
6-7	Reserved
6-8	Throttle fault converted to warning

9 Specifications

9.1.1 Electrical

Peak Battery Current	Programmable up to 96A*
Peak Phase Current	Programmable up to 96A*
Peak Regen Phase Current	Programmable up to 96A*
Continuous Phase Current	Approximately 55A at thermal rollback, varies with air flow and heat sinking
Phase Current Rollback Temp	90°C Internal Temp (casing ~70°C)
Max Battery Voltage	87V (20s Lithium, 24s LiFePO4)
Min Battery Voltage	19V (6s Lithium, 7s LiFePO4)
eRPM Limit	Not recommended above 60,000 eRPM, though it will continue to function beyond this.
RShunt for Cycle Analyst	1.00 mΩ

* Thermal rollback will typically kick in after 1 to 2 minutes of peak phase current, and then phase current will automatically reduce to maintain controller rollback temperature.

9.1.2 Mechanical

Dimensions LxWxH	106 x 56 x 39 mm
Weight	0.20 / 0.25kg (MT / L10)
Signal Cable Lengh	11cm to Connector End
Motor Cable Length	40cm to Connector End
Waterproofing	Fully Potted Circuitry, IP rated signal plugs