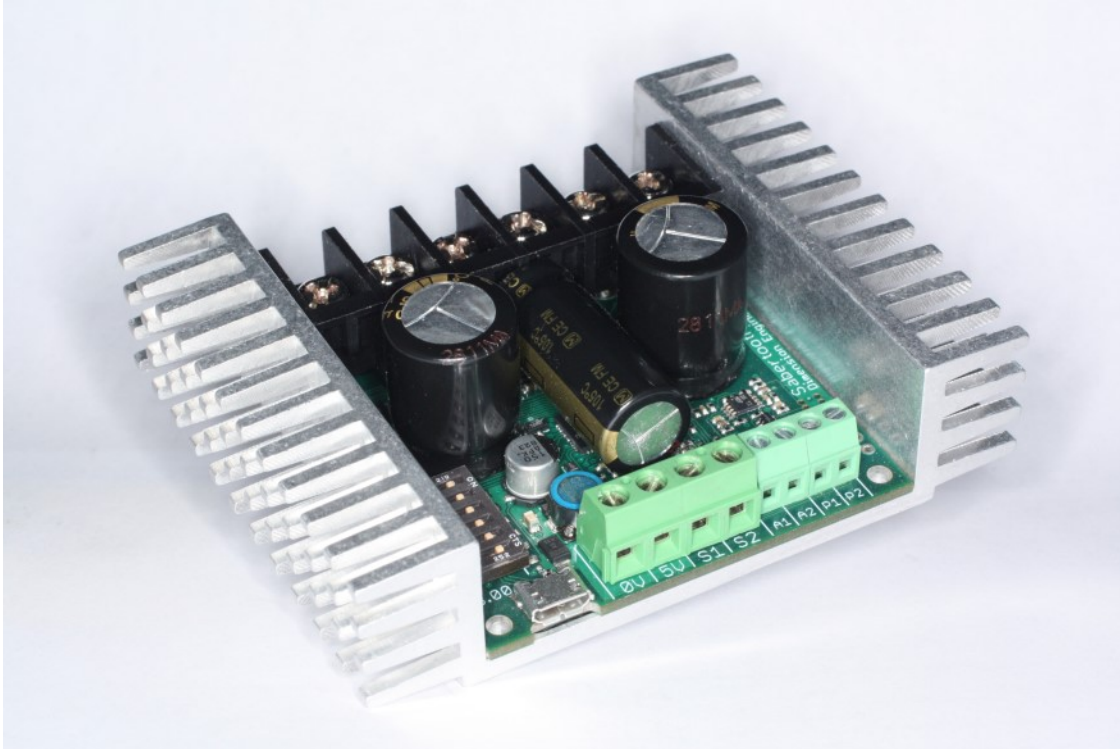


Sabertooth 2x32

Dimension Engineering



Sabertooth 2x32 is a dual channel motor driver capable of supplying 32 amps to two motors, with peak currents up to 64 amps per motor. It can be operated from radio control, analog, TTL serial or USB inputs. It uses regenerative drive and braking for efficient operation. A variety of operating modes including tank style mixing and automatic calibration allow most projects to work immediately out of the box. In addition to the standard operating modes, Sabertooth 2x32 features additional signal inputs and power outputs, as well as enhanced configuration options. User-defined operating modes allow for custom operation, such as switching between radio control and computer-driven inputs on the fly, emergency stops or front panel control overrides. The auxiliary power outputs can be configured to allow the Sabertooth 2x32 to operate from a power supply without a parallel battery, or automatic control of electromagnetic brakes. When combined with Dimension Engineering's Kangaroo x2 motion control module, the Sabertooth 2x32 can be used for closed-loop position or speed control with encoder or analog feedback. The state of the driver can be monitored in real time using the USB port in any operating mode, making debugging your project faster and easier. Sabertooth 2x32 is more flexible, robust and powerful than previous motor drivers, while also being easier to use.

Contents

Features	4
Overview	5
Specifications	6
Inputs	7
Outputs	8
Input Power.....	11
Power Supply	11
Battery.....	13
Wiring.....	14
Control Modes	16
Analog	16
Radio Control	19
Serial	22
Setup	22
Plain Text Serial.....	24
Packet Serial.....	27
Legacy Packet Serial	27
Legacy Simplified Serial.....	27
USB Mode	28
User Mode	31
DEscribe PC software.....	34
General tab	36
Analog tab	38
RC tab.....	40
Calibration.....	40
Serial and USB tab.....	42
Motor Outputs tab.....	44
Power Outputs tab.....	46
Diagnostics tab.....	48

Tools menu.....	49
User Mode	50
Setup tab.....	50
Programs.....	51
Program commands.....	52
Special considerations	54
Mounting.....	55
Drawings	56
Printable mounting diagram	57

Features

Dual motor driver with mixed and independent options

Sabertooth 2x32 will drive two motors at up to 32A continuous and 64A peak each. These can be mixed together for a tank drive type vehicle, or run independently. It can be controlled by analog voltages, R/C transmitters, TTL serial commands, USB, or a combination of these signals.

USB input

Every third generation Sabertooth motor driver comes with USB standard. This makes operating from a PC or advanced microcontroller as easy as plugging in the cable. Windows drivers are included with the DEScribe PC software, and no driver is required for Linux. USB is also used with to set options, create custom operating modes, monitor the system, and update the firmware with new features.

Auxiliary inputs and outputs

Sabertooth 2x32 has two additional 8A power outputs, which can be set up to operate electromagnetic brakes on motors, act as a voltage clamp to protect power supplies, or power other medium duty loads. Sabertooth 2x32 also has extra signal inputs and extra serial ports which enable better control options.

User-created operating modes for custom applications

We often hear that a motor driver would be perfect if it only had one more input or a minor operating change. Third generation Sabertooth motor drivers like 2x32 have user-scriptable operating modes, which allow you to mix and match analog, R/C, serial and USB inputs, create custom output functions and handle tasks automatically. You can even switch control from one input type to another. Many jobs that would have taken an additional microcontroller can now be handed with just the Sabertooth 2x32.

Self-tuning PID control using the optional Kangaroo x2 expansion board

When used with the optional Kangaroo x2 expansion board, Sabertooth 2x32 works with quadrature encoder or potentiometer feedback for speed or position control. Because it is self-tuning, you can skip the hours or days of work getting your PID coefficients dialed in.

High resolution Synchronous regenerative drive with ultrasonic switching frequency

Sabertooth 2x32 features high resolution inputs and over 4000 output speeds for the smoothest control on the market. The switching frequency is over 29 kHz, so there's no annoying motor whine. The outputs use synchronous rectification for high efficiency and low heat generation, as well as regenerative drive to save energy and extend battery life.

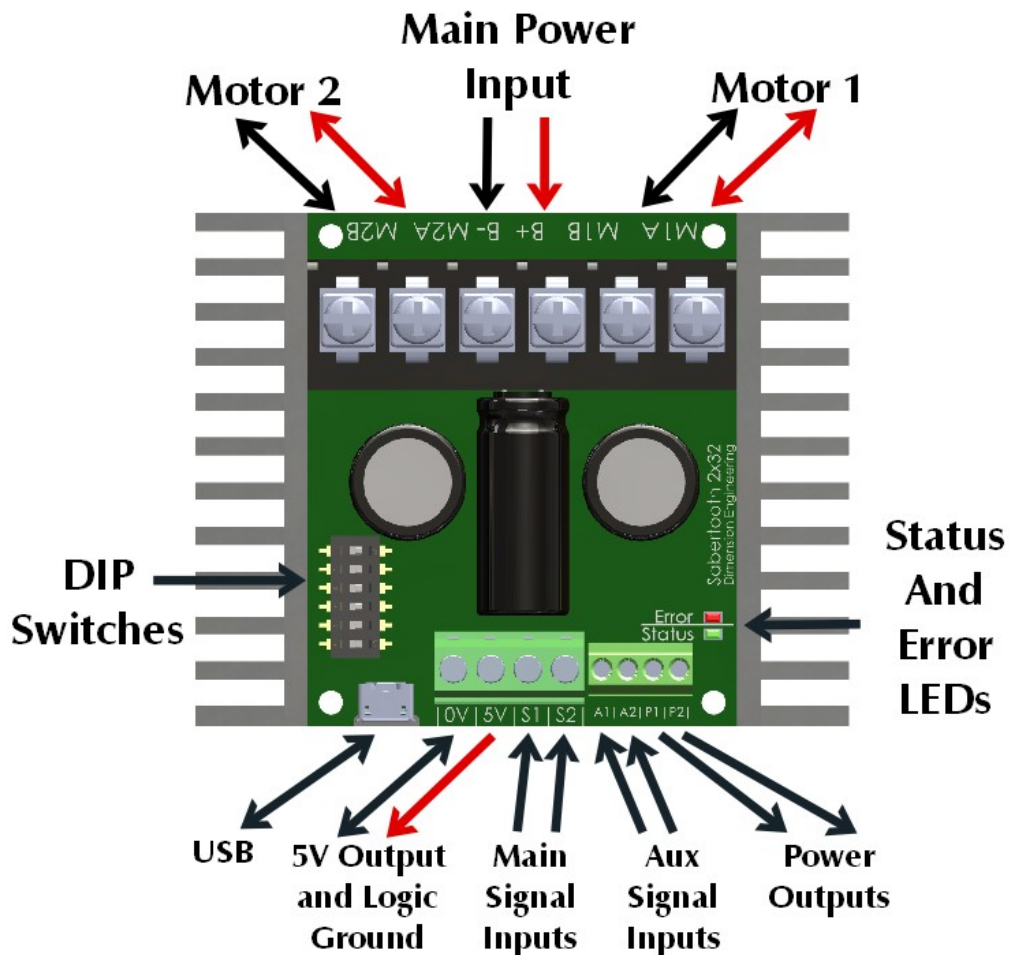
Adjustable current limit, ramp rates and thermal protection

The current and temperature limit of the Sabertooth 2x32 can be set for each motor channel. This can be used to protect the mechanism of your device, as well as protect itself.

Easy mounting and setup

Sabertooth 2x32 features a new heat sink CNC machined from a single block of aluminum. Mounting is accomplished with included 4-40 hardware. All connections are by screw terminals, so no soldering or special cables are required, other than an optional micro USB cable.

Overview



Main Power Input: Connect to a 6V-33.6V Battery or Power Supply.

Motor 1 and Motor 2: Connect Motor 1 to the M1A and M1B. Connect Motor 2 to M2A and M2B.

DIP Switches: These are used to set the operating mode and options. Can be changed while operating.

USB: A standard Micro USB port. Connect to a PC or other USB host to control, monitor or modify.

Logic Ground: The 0V logic ground is connected internally to B-.

5V Output: 5V is a regulated 5 volt output. You can use it to power additional circuitry up to 1 amp.

Main Signal Inputs: Connect your main analog, R/C or serial signals here.

Aux Signal Inputs: These may be used for additional control. Optional in most modes.

Power Outputs: These connect to voltage clamp resistors, electromagnetic brakes, field windings, or other moderate power loads. 8A max current per channel.

Status and Error LEDs: These glow and flash to indicate the status of the Sabertooth 2x32.

Specifications

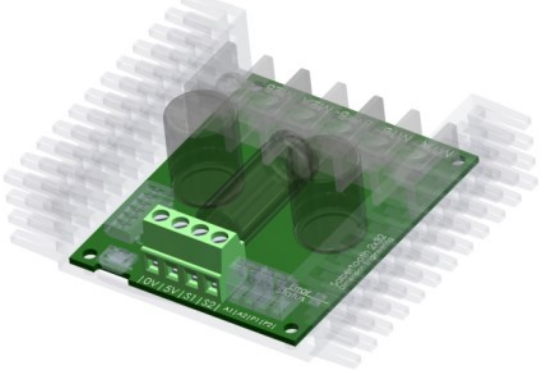
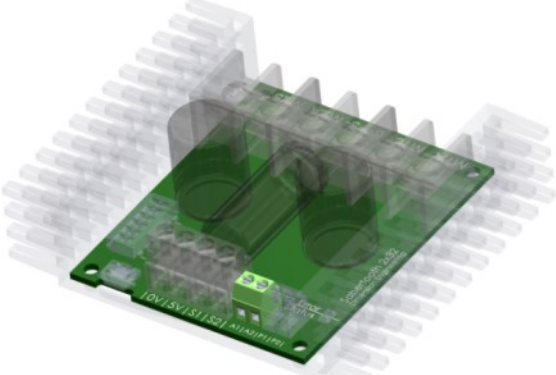
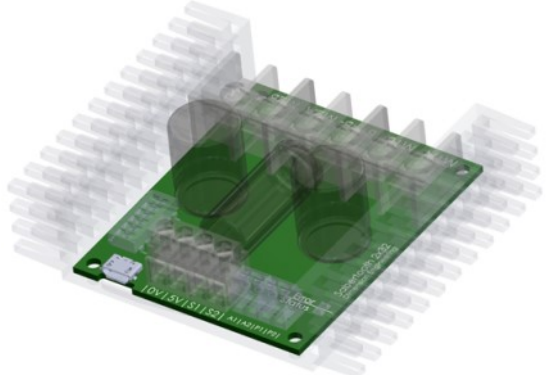
	<i>Mechanical specifications</i>		
Dimensions	2.75 x 3.5 x 1.0 inches (70mm x 90mm x 26mm)		
Weight	4.5 ounces (125 grams)		
	Minimum	Typical	Maximum
Wire size, battery	16 gauge	10 gauge	10 gauge
Wire size, motors	16 gauge	12 gauge	10 gauge
Wire size, signal	28 gauge	24 gauge	18 gauge
Operating temperature	0F (-20C)	70F (25C)	160F (70C) ¹
	<i>Electrical Characteristics</i>		
	Minimum	Typical	Maximum
Input voltage, B+ and B-	6.0 Volts	12 or 24 Volts	33.6 Volts
Continuous output current, M1 and M2	-	-	32 amps ¹
Peak output current, M1 and M2	-	-	64 amps ²
Output voltage, M1 and M2	-95% of input voltage (average)	-	95% of input voltage (average)
Voltage, P1 and P2	0V	-	Input voltage +.3V
Output current, P1 and P2			8 amps, sink only
Output voltage, 5V	4.85	5.0	5.15
Output Current, 5V	-	-	1A
Input voltage, S1 and S2	-.3V	0V to 5V	12V ³
Input voltage, A1 and A2	-.3V	0V to 5V	12V ³
Output Voltage, S2 and A2	0V		3.5V

¹ Maximum continuous output current derates linearly above 40C ambient. At 70C ambient the maximum continuous output current is 10 amps per channel.

² Can be reduced by software setting

³ Stress rating only, signals over 5 volts will be read as 5 volts by the Sabertooth 2x32

Inputs

<i>Main Signal Inputs</i> The main inputs are labeled S1 and S2. They can be set for analog, R/C or serial communications. When using serial, S1 is the RX port and S2 is the TX port.	 Main signal inputs are labeled S1 and S2
<i>Auxiliary Inputs</i> The auxiliary inputs are labeled A1 and A2. They can be set for analog, R/C or serial communications. When using serial, A1 is the RX port and A2 is the TX port.	 Auxiliary signal inputs are labeled A1 and A2
<i>USB Input</i> The Sabertooth connects via a standard micro USB plug. A windows driver installs with the DEScribe PC software. Linux and Mac will work out of the box with no driver. The USB port is also used to communicate with DEScribe to configure the Sabertooth 2x32.	 USB input. The USB input is not labeled.

Sabertooth 2x32 has four logic inputs S1, S2, A1 and A2. The analog input range is 0 to 5 volts. Digital signals can be 3, 3.3 or 5v logic. All third generation drivers such as Sabertooth 2x32 include a USB port. The USB port can be used for control from a PC or embedded processor like a Raspberry Pi. It can also be used to monitor the inputs and outputs, set up user modes and custom settings, and update the firmware.

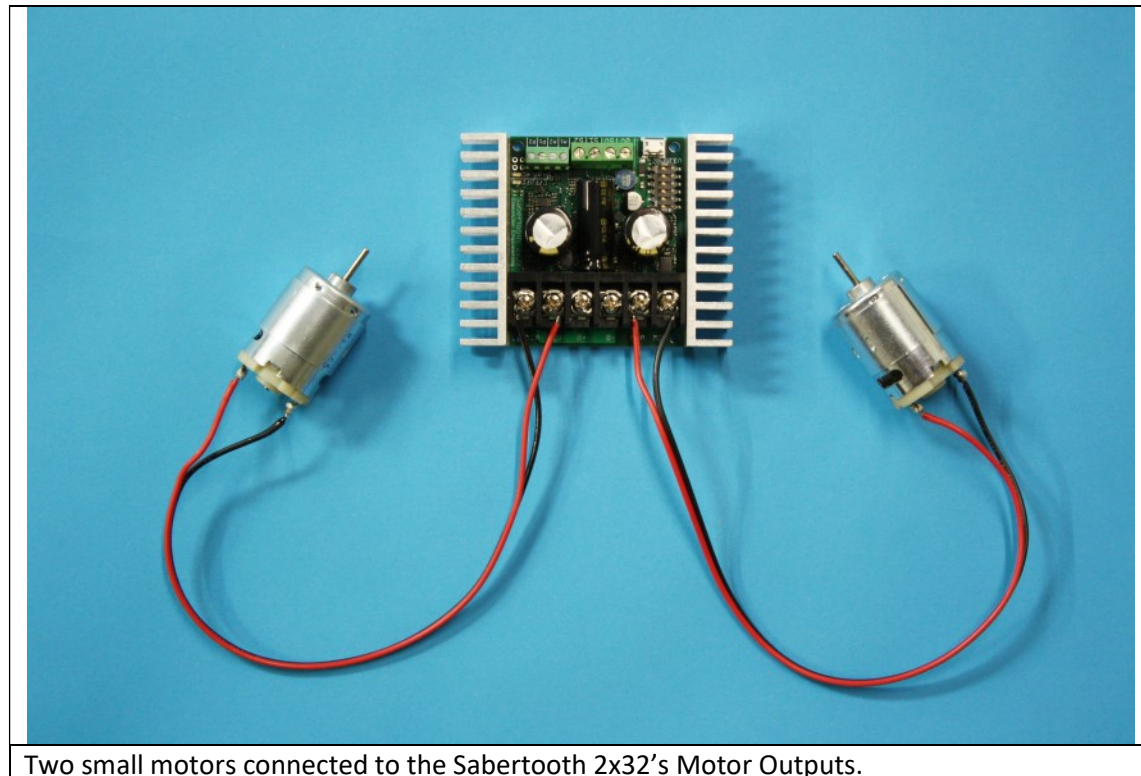
Outputs

Sabertooth 2x32 is a dual motor driver, and can drive two motors at up to 32 amps continuous and 64 amps peak current per channel. In addition, there are two 8 amp auxiliary outputs and two 20 milliamp indicator outputs.

Motor Outputs

Sabertooth 2x32's M1 and M2 motor outputs are 12 bit, synchronous regenerative motor drives. They have a switching frequency of 30 kHz for silent operation. Each channel has a programmable current limit. It is also possible to disable regeneration to drive loads like lamps or anodizing tanks, or completely disable the outputs and braking to allow motors to freewheel.

Typical motors used with a Sabertooth 2x32 include fractional horsepower permanent magnet motors, wheelchair type motors, scooter type motors and brush type cordless tool motors. Most two wire, permanent magnet motors can be made to work. Sabertooth motor drives have also been used to drive large solenoids, lamps, heaters, coolers, electromagnets, shakers and transducers. To minimize heating and losses, use as large a wire to your motors as is practical. 12 gauge is typical for short wiring runs.



Two small motors connected to the Sabertooth 2x32's Motor Outputs.

Indicator Outputs

Two of the signal inputs, S2 and A2, can be used as 20 milliamp, 3.3 volt outputs. This is typically used to drive LEDs for remote control panels or signal back to microcontrollers. The indicator outputs are

secondary functions of the S2 and A2 inputs. When in indicator mode, these pins have a built-in 220 ohm series resistor, so they can drive loads directly.

Power Outputs

These open collector outputs can sink up to 8 amps of current each. The power outputs can be configured as Voltage Clamps, Brakes or Controllable Outputs.

Voltage Clamp

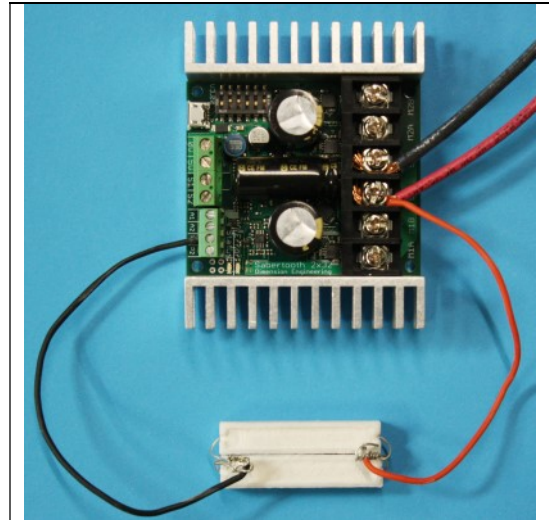
With a typical regenerative driver, it is difficult to operate from a power supply, because while braking there is nowhere for the regenerated energy to dissipate. This can lead to power supply shutdown or damage. By connecting the power outputs to a resistor pack (sold separately, or construct your own) the Sabertooth 2x32 can operate from a power supply without an additional battery or circuitry. The resistor's value should be calculated to provide the typical motor current or 8 amps, whichever is less. Voltage clamp mode is selected using the Power

Outputs tab in the DEScribe software. Voltage Clamp is the default behavior for both the P1 and P2 outputs.

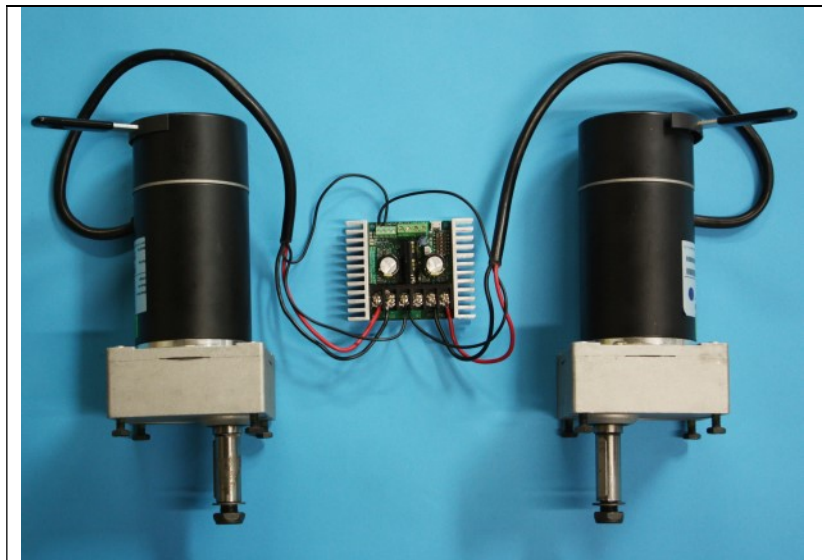
Brakes

The power outputs can also be used to operate electromagnetic brakes. Systems such as wheelchairs often have brakes to prevent rolling away when power is removed. Brakes are also used in CNC machines and automation to hold alignment when power is removed or reduce power consumption. In brakes mode, the

brakes automatically disengage when motion is commanded, and engage when the motor stops. When using Brakes mode, the P1 output is linked to the M1 motor output, and the P2 output is linked to the M2 motor output. With the brake timing changed, brakes mode can also be used to control the field in a shunt wound or separately excited motor. Brakes mode must be selected and configured from the Power Outputs tab using the DEScribe software.



Power output connected to a resistor and configured as a voltage clamp



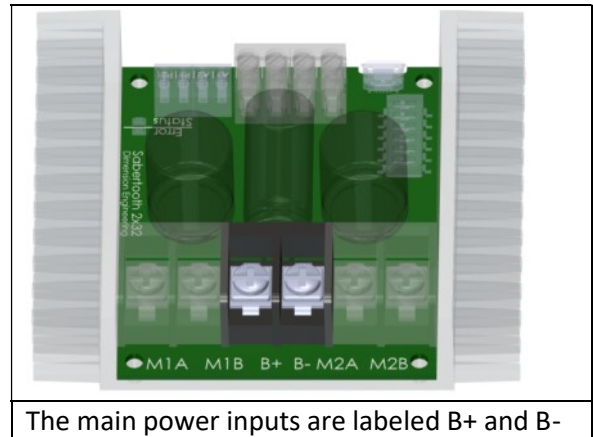
Power outputs connected to electromagnetic brakes.

Controllable Output

Finally, the power outputs can be used as additional variable power outputs. These could be used to drive fans, solenoids, valves, heaters or similar medium power devices. This is primarily used with serial or USB inputs, or custom User Mode programs. Controllable Output must be selected in DEScribe.

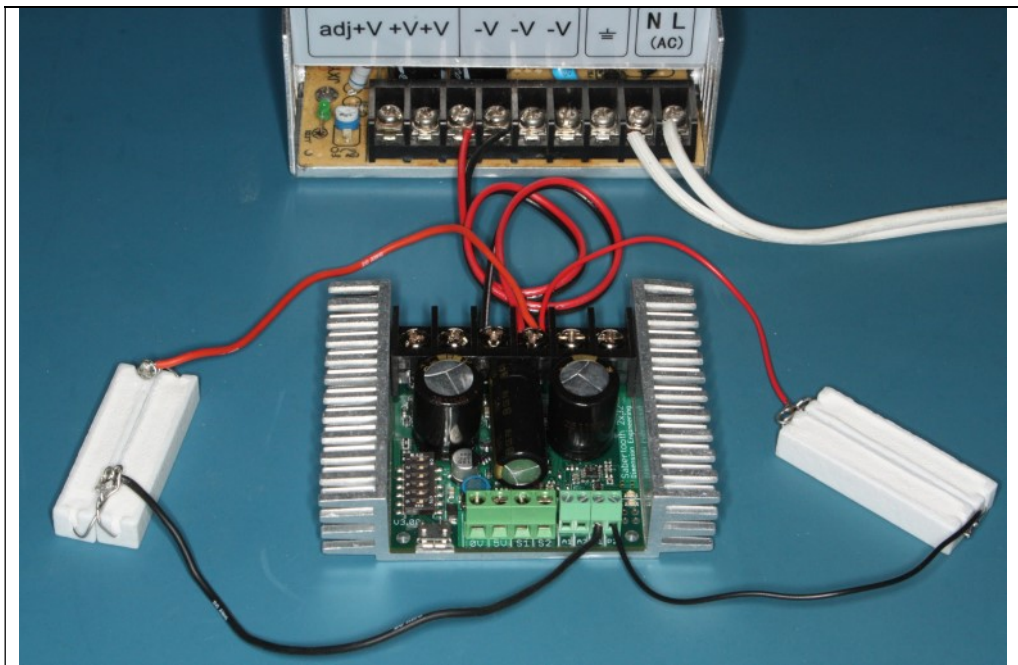
Input Power

Sabertooth 2x32 can be used with power supplies or batteries. Input power is connected to the center power terminals labeled B+ and B-. The input voltage range of the Sabertooth 2x32 is 6.0V to 33.6V. The input current is dependent on the motors being used and the load placed upon them. The input current can be limited by reducing the current limit of one of both motor channels.



The main power inputs are labeled B+ and B-

Power Supply



Sabertooth 2x32 connected to a power supply and voltage clamp resistors.

Sabertooth 2x32 can be used with a suitably sized power supply. Power supplies used with a Sabertooth 2x32 should have output voltages between 7V and 30V. It is important to make sure the power supply being used can produce enough current for the load. Power Supply mode is selected by setting DIP switch 3 to the ON position. One major improvement in the third generation Sabertooth motor drivers is the addition of the Power outputs P1 and P2. These can be used with an appropriate resistor to

dissipate the regenerative current generated when the device slows down or stops. This allows the use of power supplies without batteries or capacitors to absorb the regenerated energy. A calculator on the Dimension Engineering website will assist with choosing appropriate resistor packs. At least one of the power outputs should be configured as a voltage clamp when using a power supply.

Battery

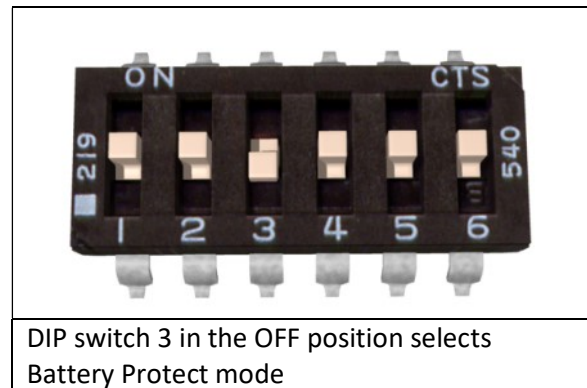
Sabertooth 2x32 can be used with a variety of battery types and voltages. When running from a battery that can be damaged by deep discharge, such as a lithium polymer or lead acid battery, you should set DIP switch 3 to the battery protect position, which is OFF. When in this position, the Status LED will blink out the number of detected cells, followed by a pause, in a repeated pattern. This is useful to verify that the Sabertooth has detected the correct number of cells. When the battery is depleted, the Sabertooth will stop driving the

motors, and the Status and Error LEDs will both blink in sync. When using a battery, please ensure that it can handle the current draw the motors will supply. Because Sabertooth is a regenerative motor driver and will return energy to the battery when braking is commanded, only rechargeable batteries are recommended.

With the DEScribe PC software, you can change what type of battery the Sabertooth is protecting. The default setting for battery protect is Lithium Polymer. The options for batteries are laid out in the following table. For each type, you can modify the minimum cell voltage.

Battery Type	Number of series cells in pack	Nominal Voltage
Lithium Polymer or Lithium Ion	2 to 8	7.4V to 29.6V
Lithium FerroPhosphate (LiFePO ₄)	3 to 9	9.9V to 29.7V
Lead Acid	1 to 4 6V batteries or 1 to 2 12V	6V to 24V
Nickel Metal Hydride (NiMH)	6 to 20	7.2V to 24V
Nickel Cadmium (NiCd)	6 to 20	7.2V to 24V
Custom	1 to 20	-

By default, the number of cells is automatically detected. The Sabertooth assumes the battery is fully charged when it powers up. With high cell count battery packs that are partially discharged, this can be inaccurate. If you are not planning to use multiple voltages, you can specify the exact battery that will be used to avoid this problem.



Wiring

Power connections

As a general rule of thumb, you should use the thickest wire that is practical to make power connections, especially on the battery leads. Using undersized wire will lead to the wire getting hot, and can lead to elevated temperatures on the Sabertooth 2x32 as well.

The main power connections to the Sabertooth 2x32 are on the rear edge of the board. Connections are made to large black screw terminals. These terminals will accept 10 to 24 gauge wire. Using stranded wire it is possible to run twinned 10 gauge wire connections to the battery terminals. This is often a good idea if your design will be running both motors near or above the 32 amp continuous limit. For the motor connections, single 10 gauge wires should be sufficient for all applications.

Motor 1 connects to the M1A and M1B terminals. Motor 2 connects to the M2A and M2B terminals.

If you are using the power outputs as regenerative clamps, one side of the resistor pack connects to the P1 terminal, and the other side connects to the B+ terminal. You may find it easier to connect the positive side of the power input through a bus bar or in the wiring harness instead of at the Sabertooth itself. This is acceptable. By default the power outputs are set up for regenerative clamp.

If you are using the power outputs for electromagnetic brakes, the positive side of both brakes connect to the B+ terminal. The negative side of the M1 brake connects to P1, and the negative side of the M2 brake connects to P2. Remember that to use brakes you must set the power outputs to brakes mode using the DEScribe PC software.

If you are using the power outputs at a voltage other than the main system voltage, such as to run a 5v cooling fan, connect the negative side of the device being powered to the P1 or P2 auxiliary power output, and the positive side to that supply voltage. If you are using an inductive load such as a motor from a different voltage, you will need to install a flyback diode across the device to prevent problems.

Signal connections

The signal connections, as well as the auxiliary power outputs, connect to the smaller green screw terminals on the front edge of the board. All the signal inputs can accept signals between 0V and 5V. In digital input modes, logic high can be between 2.7V and 5V. This allows for interface to boards using 3.3 or 2.7 volt microcontrollers without the need for level translators.

USB: The USB port is used for connection to a PC, tablet or advanced microcontroller. If only the USB port is connected, the internal logic circuitry of the Sabertooth will be active, but neither input nor output terminals will operate. This is useful to enable changing the operating options or updating the firmware without powering on the entire device. In a production setting, it is also useful to set up the Sabertooth 2x32 before installation.

0V: This is the main logic ground of the Sabertooth 2x32. It is internally connected to B-.

5V: This is a 5V output, and can supply up to 1A of current to devices such as receivers, potentiometers, microcontrollers or servos.

S1: This is a main signal input. Its functionality depends on the operating mode. Input voltages into S1 should be between 0V and 5V.

S2: This is a main signal input. Its functionality depends on the operating mode. Input voltages into S2 should be between 0V and 5V. S2 can also be set up as an Indicator type output.

A1: This is an auxiliary signal input. Its functionality depends on the operating mode. . Input voltages into A1 should be between 0V and 5V.

A2: This is an auxiliary signal input. Input voltages into A2 should be between 0V and 5V. A2 can also be set up as an Indicator type output.

Wiring Summary

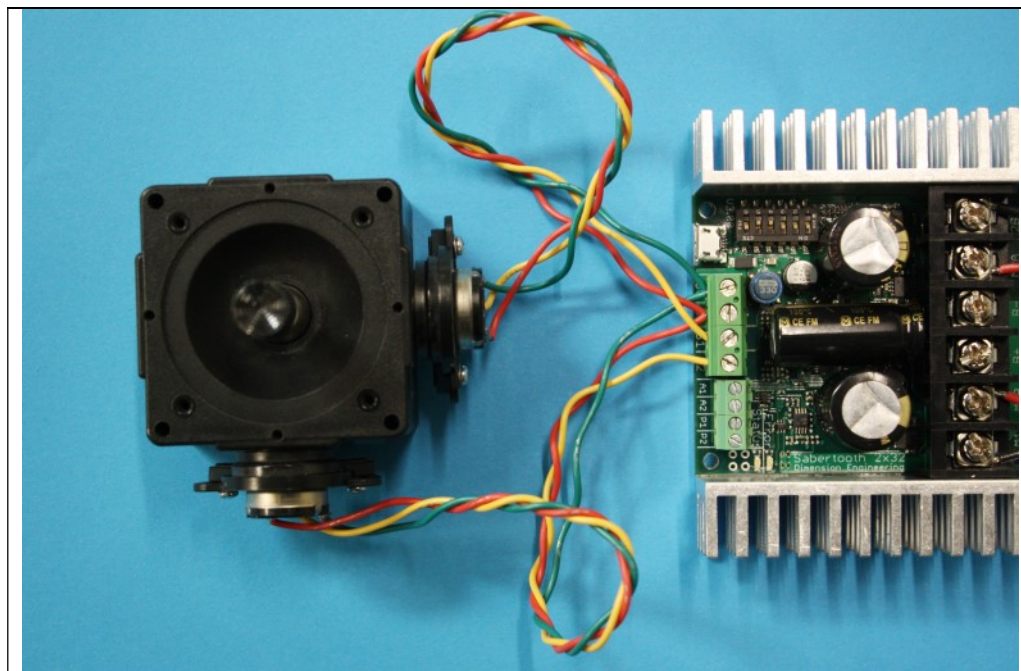
Name	Description	Voltage	Current
B+	Power input positive	6V – 33.6V	64A /128A peak
B-	Power input negative	0V only	64A /128A peak
M1A	Motor 1 output	0V – 33.6V	32A/64A peak
M1B	Motor 1 output	0V – 33.6V	32A/64A peak
M2A	Motor 2 output	0V – 33.6V	32A/64A peak
M2B	Motor 2 output	0V – 33.6V	32A/64A peak
0V	Logic ground	0V only. Tied internally to B-	2A
5V	5V power output	5V	1A
S1	Signal input 1	0V – 5V	1mA
S2	Signal input 2, serial TX	0V – 5V	1mA in, 20mA out
A1	Auxiliary input 1	0V – 5V	1mA
A2	Auxiliary input 2, serial TX	0V – 5V	1mA in, 20mA out
P1	Power output 1. Open collector current sink only.	0V – 33.6V	8A
P2	Power output 2. Open collector current sink only.	0V – 33.6V	8A
USB	Micro USB port		100mA max

Control Modes

Sabertooth 2x32 has four main control modes, plus a special User mode that can be used to create custom control modes. Control Modes are selected via the six DIP switches. DIP switches 1 and 2 select the control mode, and DIP switches 4, 5 and 6 select the options within each control mode.

Analog

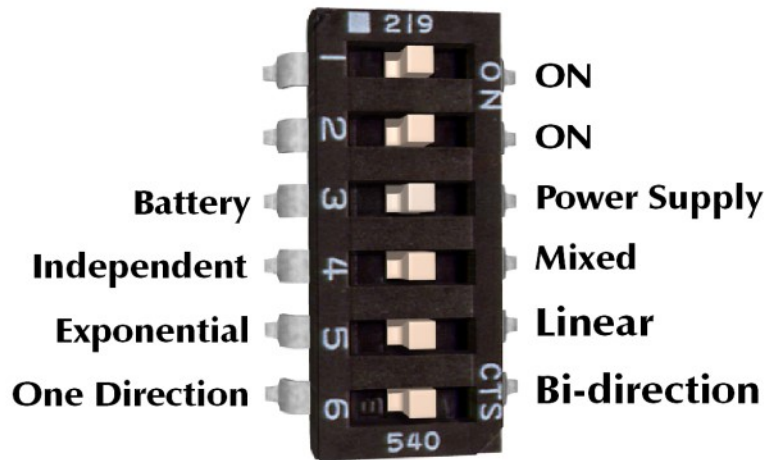
Analog control uses analog voltages to send commands to the Sabertooth 2x32. This is the simplest way to control a Sabertooth. These voltages can be generated by potentiometers, switches, joysticks or digital to analog converters. The input voltage range is 0 volts to 5 volts. Custom ranges can be set using the DEScribe PC software. In Analog the S1, S2, A1 and A2 terminals are analog inputs. They are internally pulled down to 0V if not connected to a signal.



An analog joystick connected to a Sabertooth 2x32

To use analog mode with a potentiometer or joystick, connect the negative terminals to 0V. Connect both positive terminals to 5V. Feed the signals into S1 and S2. If used, A1 and A2 are connected the same way.

Analog Mode DIP Switch Settings



DIP switch 4 selects between mixed and independent mode.

Mixed mode is selected by setting DIP switch 4 to the ON position. Mixed mode is primarily used for differential drive mobile vehicles. The signal going to S1 controls the forward/backwards speed of both motors. The signal going to S2 controls the right/left turning of both motors.

Independent mode is selected by setting DIP switch 4 to the OFF position as shown. In Independent mode, the speed of the motor connected to the M1 motor outputs is controlled by the analog signal sent to the S1 input, and the speed of the motor connected to the M2 output is controlled by the analog signal sent to the S2 input.

DIP switch 5 selects between linear and exponential control.

Linear control makes the speed of the output motor directly proportional to the input voltage. This is best for control systems and speed adjustment dials. Linear control is selected by setting DIP switch 5 to the ON position.

Exponential control makes the motors less responsive around the zero speed point. This is useful for fine control of small motions, while not losing the ability to go full speed. The exponential mapping is saved in EEPROM, and can be modified with the DEScribe PC software. Although this mode is called exponential, you can use the software to create other input to output functions as well.

DIP switch 6 selects between bidirection and single direction modes.

Bidirection control is selected by setting DIP switch 6 to the ON position. In this mode, an input signal of 2.5 volts is stopped. Voltages higher than 2.5 volts cause the motor to go forward, and voltages less than 2.5 volts cause the motor to go in reverse.

Single direction mode is selected by setting DIP switch 6 to the OFF position. In this mode, a command of 0V stops the motor and a command of 5V is full speed. The direction the motor runs can be controlled by the A1 and A2 inputs. If you connect a switch between 5V and A1, it will act as a forward/reverse

switch for the signal being fed into S1. If only a single direction is needed, only S1 and S2 need to be connected.

Speed ramping

In all analog modes except Single Direction, an analog signal sent to the A1 auxiliary input controls the ramp rate for both channels of the motor driver. This is useful to make gentle motions or limit peak currents due to acceleration. For example, a ride-on electric skateboard that started at full power immediately will throw its rider off, but one with a several second ramp rate is easy to ride. If this input is not connected, the input's internal pull-down will automatically set the fastest response. If adjustable ramping is not desired, leave the A1 input disconnected. The ramp rate can also be set using the DEScribe PC software.

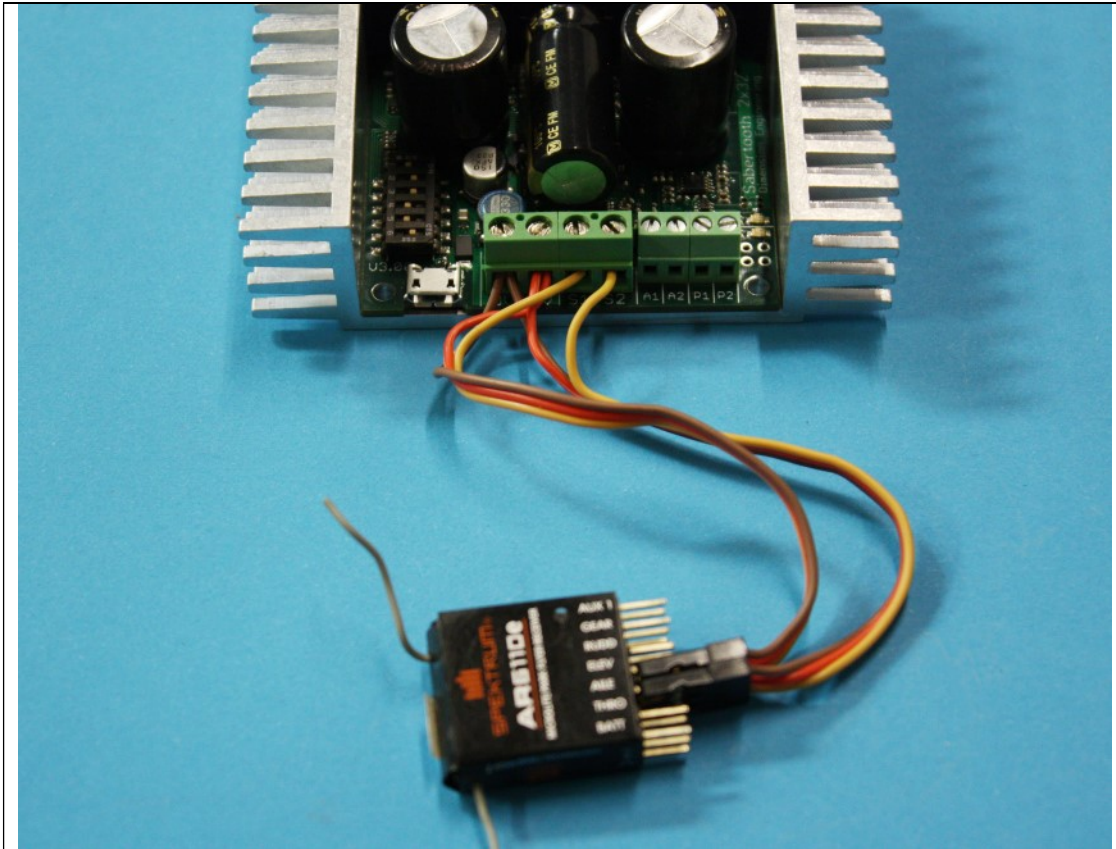
Maximum speed

In all analog modes except Single Direction Mode, a signal sent to the A2 auxiliary input sets the maximum speed for both motor outputs. This can be used for fine control, because as the maximum speed scales down, so do all other inputs. You might want to reduce the maximum speed for very precise control while inspecting a specimen, then turn the speed all the way back up to jog over to the next. If this input is not connected, the input's internal pull-down will automatically set the fastest speed, so if adjustable max speed is not desired, you should leave the A2 input disconnected.

Other analog control mode options, such as automatic calibration, are available by using the DEScribe PC software. You can also use DEScribe to create custom analog control modes.

Radio Control

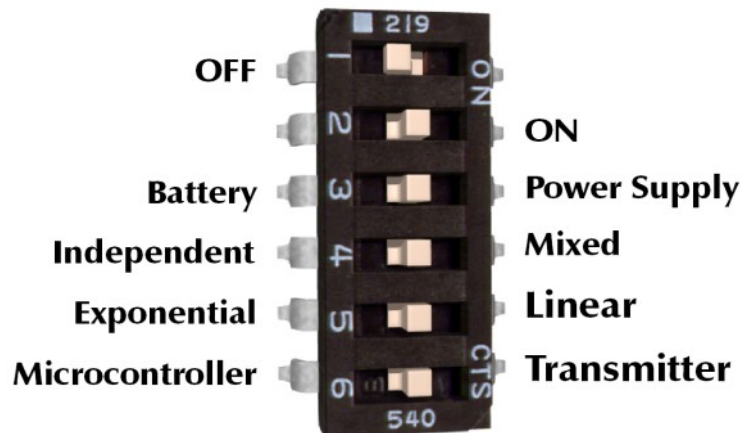
Radio Control uses R/C (servo) pulses to send commands to the Sabertooth 2x32. These signals are generated by R/C transmitters and receivers, or by microcontrollers. Anything that can generate servo signals can be used to drive a Sabertooth in Radio Control mode. In Radio Control mode the S1, S2, and A2 inputs are set up to read R/C pulses. The A1 input is set up as an analog input and can be used with a potentiometer for an adjustable ramp rate.



A radio receiver being used to control a Sabertooth 2x32

R/C signals are typically carried on servo pigtails, which are three wires. The brown wire is ground, and connects to 0V. The red wire connects to 5V. The orange or white wire carries the signal, and connects to S1, S2 or A2. Sabertooth 2x32 will power a receiver, so no separate receiver battery is required.

Radio Control Mode DIP Switch Settings



DIP switch 4 selects between mixed and independent mode.

Mixed mode is selected by setting DIP switch 4 to the ON position as shown. Mixed mode is primarily used for differential drive mobile robots. The signal going to S1 controls the forward/backwards speed of both motors. The signal going to S2 controls the right/left turning of both motors.

Independent mode is selected by setting DIP switch 4 to the OFF position as shown. In Independent mode, the speed of the motor connected to the M1 motor outputs is controlled by the R/C signal sent to the S1 input, and the speed of the motor connected to the M2 output is controlled by the R/C signal sent to the S2 input.

DIP switch 5 selects between linear and exponential control.

Linear control makes the speed of the output motor directly proportional to the input signal. This is best for control systems, as well as microcontrollers. Linear control is selected by setting DIP switch 5 to the ON position.

Exponential control makes the motors less responsive around zero speed. This is useful for fine control of small motions, while not losing the ability to go full speed. It is especially useful for differential drive robots and vehicles. The exponential mapping is saved in EEPROM, and can be modified with the DEScribe PC software.

DIP switch 6 selects between Transmitter and Microcontroller mode.

Transmitter mode is selected by setting DIP switch 6 to the ON position. Because all R/C transmitters use slightly different timing, transmitter mode automatically calibrates the stopped position at startup. The maximum and minimum signals are constantly recomputed. In this way, the Sabertooth learns the transmitter's signals every time it is powered up. In Transmitter mode, loss of signal from the receiver will stop the motor driver. Be advised that some receivers, especially 2.4 GHz receivers, will continue to put out servo signals even if communication with the transmitter is lost, so check the manual for your receiver's failsafe behavior.

Microcontroller mode is selected by turning DIP switch 6 OFF. In microcontroller mode, the input pulse ranges are fixed. A 1500us signal is stopped, 1000us is full reverse and 2000us is full forward. To allow for slower microcontrollers like Basic Stamps, by default there is no timeout in Microcontroller mode.

DEscribe can be used to override these settings and use other input ranges, as well as change the timeout behavior and timing. Using DEscribe, there is also a Saved calibration mode which learns the transmitter settings one time only, and a Joystick mode which only automatically calibrates the center position.

Speed ramping

In all R/C modes, an analog signal sent to the A1 auxiliary input controls the ramp rate for both channels of the motor driver. This is useful to make gentle motions or limit peak currents due to acceleration. For example, an R/C lawnmower that started at full power immediately might tear up soft grass. If this input is not connected, the input's internal pull-down will automatically set the fastest response. If adjustable ramping is not desired, leave the A1 input disconnected. The ramp rate can also be set using the DEscribe PC software.

Flip input

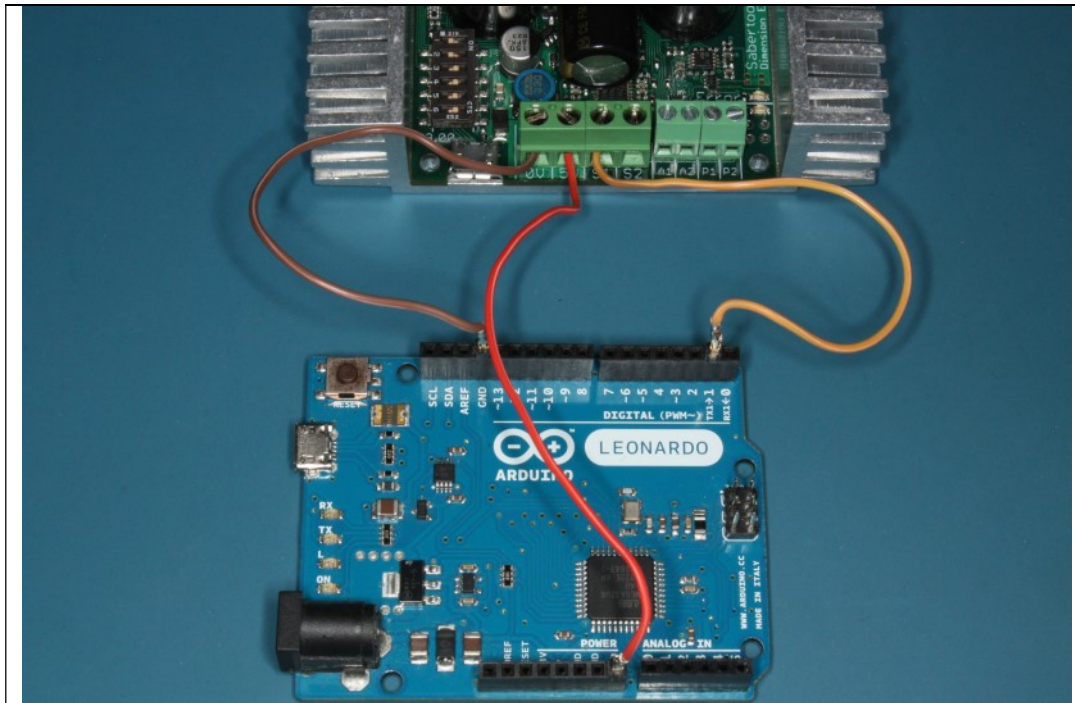
The auxiliary A2 input is also an R/C input. It is an optional Flip channel. In mixed mode, an R/C signal coming from a switch (such as the channel 5 gear switch on the transmitter) is used to select between normal and inverted mode. This is used for mobile robots that can run both right side up and upside down. If the robot is flipped upside down, flipping the gear switch will correct the steering.

Serial

Serial mode is used to control the Sabertooth 2x32 from a PC or microcontroller.

Setup

There are a variety of serial communication protocols that can be used with Sabertooth 2x32. By default, all of these modes use 9600 baud, 8N1 TTL serial levels. The baud rate can be changed using the DEScribe PC software.

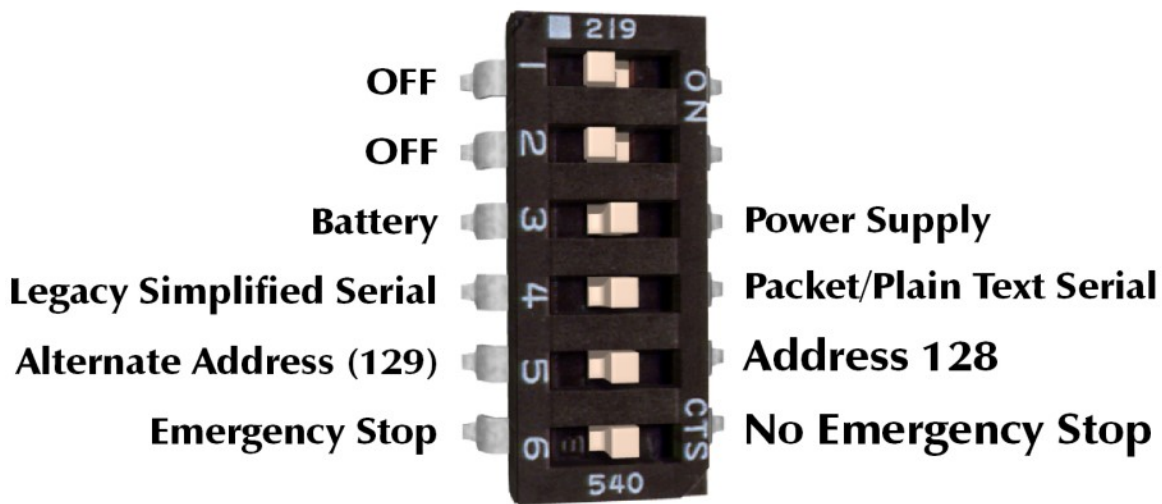


A Sabertooth 2x32 receiving serial commands from an Arduino microcontroller

Wiring

Wire the S1 connection on the Sabertooth 2x32 to the serial TX pin of your microcontroller. To use any command that reports data back from the Sabertooth 2x32, connect the S2 terminal to the Serial RX pin of your microcontroller. Connect the 0V terminal to the microcontroller's ground. The Sabertooth 2x32's 5V output may be used to provide power to the microcontroller.

Serial Mode DIP Switch Settings



DIP Switch 4 selects between packet/plain text serial and legacy simplified serial.

If DIP switch 4 is in the ON position, the Sabertooth 2x32 is listening for packet serial or plain text serial commands. It will automatically respond to either type of command.

If DIP switch 4 is in the OFF position, the Sabertooth 2x32 is listening for legacy simplified serial commands. This mode uses single-byte commands to command each motor, and is included primarily for compatibility. If you are developing a new design, we recommend using one of the newer command protocols.

DIP Switch 5 selects the packet serial address.

In packet serial mode, DIP switch 5 sets the packet serial address. The ON position sets address 128, and the OFF position sets the address to a user-definable alternate. By default, this is address 129, but it can be changed in DEScribe. Packet serial addresses are used to run multiple Sabertooth motor drivers on the same serial line.

If you are sending plain text serial commands, DIP switch 5 has no effect.

In legacy simplified serial, DIP switch 5 selects between 9600 baud (switch 5 ON) and a user-defined setting, which defaults to 2400 (switch 5 OFF)

DIP Switch 6 enables or disables emergency stops.

If DIP switch 6 is in the OFF position, emergency stops are enabled. This emergency stop is active low and internally pulled down. To enable the M1 output, connect the A1 terminal to 5V. To enable the M2 output, connect the A2 terminal to 5V. If these connections are broken and emergency stops are enabled, the motors will stop immediately. This might be used for a safety E-stop in a machine, to only allow motion while a dead-man switch is held, or to detect a disconnected control cable.

Plain Text Serial

Plain text serial mode uses ASCII formatted plain text strings to control the Sabertooth 2x32. This makes serial communications especially easy, as one can open a terminal window on a PC and start typing commands, or use a printf command on a microcontroller. Using Plain Text Serial mode allows full control of the Sabertooth 2x32. The tradeoff is plain text takes more data to send the same information, so the maximum update rate may be less. Plain text serial has an optional checksum, but by default does not require one. Data integrity is not assured unless DEScribe is used to make the checksum required.

All commands follow the same format. All commands consist of a two letter destination, followed by a colon, followed by the command and a newline (Enter key). For example, **M1: -532(Enter)**

Basic commands

Outputs are controlled by sending the destination address, followed by a colon, followed by the power setting, followed by a newline. Most settings take commands from -2047 to 2047. Commands outside this range will be ignored. If you are using a terminal program to command the Sabertooth, remember that the command only takes effect when you press the Enter key.

Destination address	Description
M1	Motor 1
M2	Motor 2
MD	Drive channel. Both motors .Forward/Backwards in Mixed Mode
MT	Turn channel. Both motors. Right/Left in Mixed Mode
P1	Power output 1
P2	Power output 2
R1	Ramp rate motor 1
R2	Ramp rate motor 2
Q1	Auxiliary variable 1
Q2	Auxiliary variable 2

Example	Result
M1: 2047\r\n	Motor 1 will go forward full speed.
M2: -1023\r\n	Motor 2 will go backwards at half speed.
M1: 0\r\n M2: 0\r\n	Motors 1 and 2 will stop.
MD: 0\r\n MT: -512\r\n	The robot will turn left at ¼ speed. Please note that to control using the mixed commands MD and MT, you must have sent both a turn and a drive command at some point. Once you have sent them both the first time, you only have to update the turn or drive commands as you desire, it is not necessary to always send them both.
P1: 2047\r\n	Power output 1 will be set to full power. Note that the power outputs are only controllable if they are set as additional output using the DEScribe software.

	Otherwise they will perform their normal voltage clamp or brake behavior and ignore serial input.
R2: 2047\r\n	This sets the speed ramping to the maximum amount. At this setting, it will take the Sabertooth approximately 8 seconds to go from a stop to full speed.
Q1: 1\r\n	By default the Q parameters control motor freewheeling. Setting a positive value will disable the motor channel. Freewheeling and shutdown are different. In a freewheel state the motor will act as though there is no power applied and the motor leads are floating. This makes the motor shaft easier to turn manually. In a shutdown state, the motor will act as if there is no power and the motor leads are connected together. This makes the motor shaft harder to turn manually. In user modes, the Q parameters are often used to change the operation of the program or activate special modes.

Note: \r\n is not visible in a terminal program, but must be included when using a function such as printf to designate the end of the message.

Get Commands

Values can be read back from the inputs and outputs using the get command. Note that when reading from the inputs, it will read according to the current pin configuration. The Sabertooth 2x32 will respond back in the same format: the address, followed by a colon and a space, followed by the value and a newline. Remember that when connected using USB, the get commands are always available, even if you are using R/C or analog mode. This can be helpful for debugging.

Command	Result	Examples
M1: get\r\n M2: get\r\n	Returns the duty cycle of the M1 or M2 output, from -2047 for full reverse to 2047 for full forward.	M1: get\r\n might return M1: -875\r\n
P1: get\r\n P2: get\r\n	Returns the duty cycle of the P1 output. -2047 is no output, and 2047 is full power. If the power output is set up for voltage clamp or brakes, this can use used to determine if the brakes or output clamp is engaged.	P1: get\r\n might return P1: -2047\r\n P2: get\r\n might return P2: 1776\r\n
S1: get\r\n S2: get\r\n	Returns the input value for the S1 or S2 inputs. In USB mode, by default these are set as analog inputs. In packet serial mode, they are serial RX and TX. By creating a user mode	S1: get\r\n might return S1: 0\r\n S2: get\r\n

	program, they can be set to any of the input types.	might return S2: 2012\r\n
A1: get\r\n A2: get\r\n	Returns the input value for the S1 or S2 inputs. By default these are analog in USB and serial modes.	A1: get\r\n might return S1: -2047\r\n

Extended Get commands

In addition to these values, outputs of the Sabertooth such as motor current and battery voltage can be read by using the extended get commands. Because each motor has its own current and temperature, the structure is slightly different. The destination address should be M1 or M2, and then the command letter is added. On the reply, the type of reading is added before the value. Some values, such as the battery voltage, are shared between both motors, so polling either M1 or M2 will return the same value.

Command	Result	Examples
M1: getb\r\n M2: getb\r\n	Returns the battery voltage in tenths of a volt. A battery reading of 12.5 volts will report as B125	M1: getb\r\n might return M1:B240\r\n
M1: getc\r\n M2: getc\r\n	Returns the motor current in tenths of an amp. Please note this is a noisy signal and may vary by up to several amps. This is normal. Positive current values mean energy is being drawn from the battery, and negative values indicate energy is being regenerated into the battery.	M1: getc\r\n might return M1:C320\r\n M2: getc\r\n might return M2:C-20\r\n
M1: gett\r\n M2: gett\r\n	Returns the temperature of the output transistors for this channel, in degrees C.	M1: gett\r\n might return M1:T30\r\n M2: gett\r\n might return M2:T85\r\n

Note: \r\n is not visible in a terminal program, but must be included when using a function such as scanf to designate the end of the message.

Additional commands

Command	Result	Examples
M1:shutdown\r\n M2:shutdown\r\n	Shuts off the motor output. Using the shutdown command will put the motor in a hard brake state.	M1: shutdown\r\n M2: shutdown\r\n
M1:startup\r\n M2:startup\r\n	Returns the motor channel from a shutdown state to normal operation.	M1: startup\r\n M2: startup\r\n

Packet Serial

Packet serial mode uses much the same commands as plain text serial mode, but does so in a more compact binary communication protocol with a reliable checksum or CRC. There are open source libraries for Arduino and C# and sample code for both on the Dimension Engineering website, as well as documentation on the protocol itself.

Legacy Packet Serial

For compatibility with existing programs, Sabertooth 2x32 supports the same 4 character packet serial commands as previous generation Sabertooth motor drivers. Using these commands only allow for 8 bits of motor output precision, and does not have a robust CRC as the new packet serial commands do, but they work acceptably in most applications. The Sabertooth 2x32 will automatically differentiate between the legacy and new commands, so the setup is the same as for Packet Serial. Using the legacy commands does not allow motor driver information such as battery voltage, temperatures or currents to be read back from the Sabertooth.

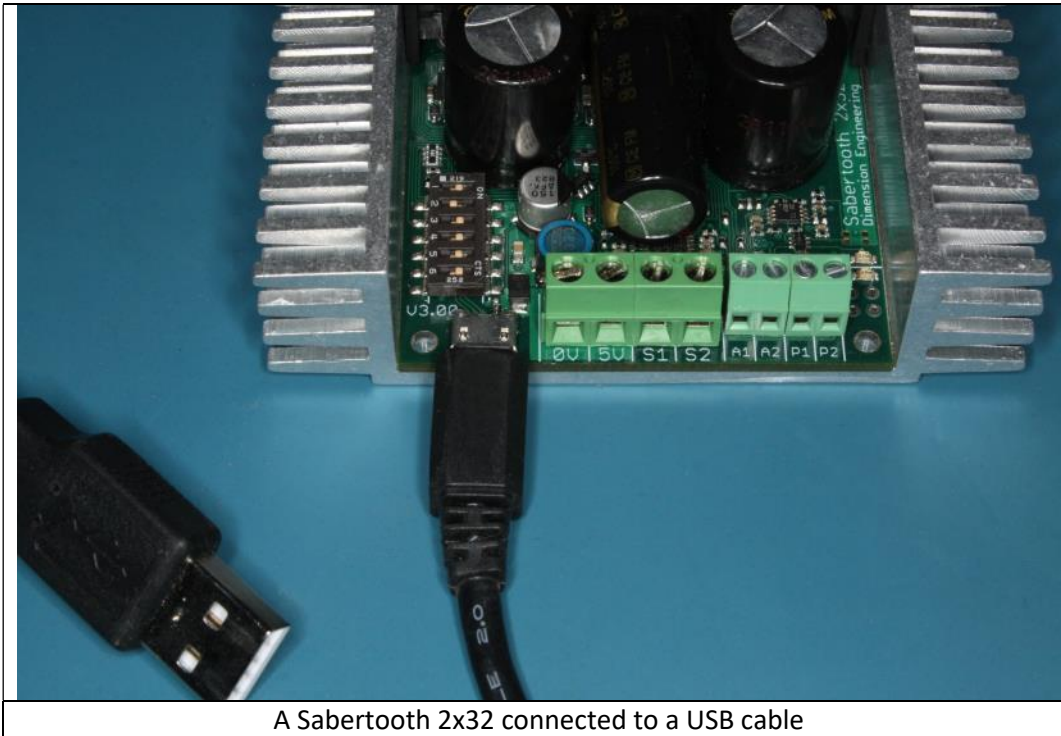
The legacy packet serial commands are documented in a separate application note on Dimension Engineering's website.

Legacy Simplified Serial

For compatibility with existing programs, Sabertooth 2x32 supports Legacy Simplified Serial mode, which is the same as Simplified Serial mode on previous generation motor drivers. The DIP switch settings are different, and the auxiliary inputs are not used.

USB Mode

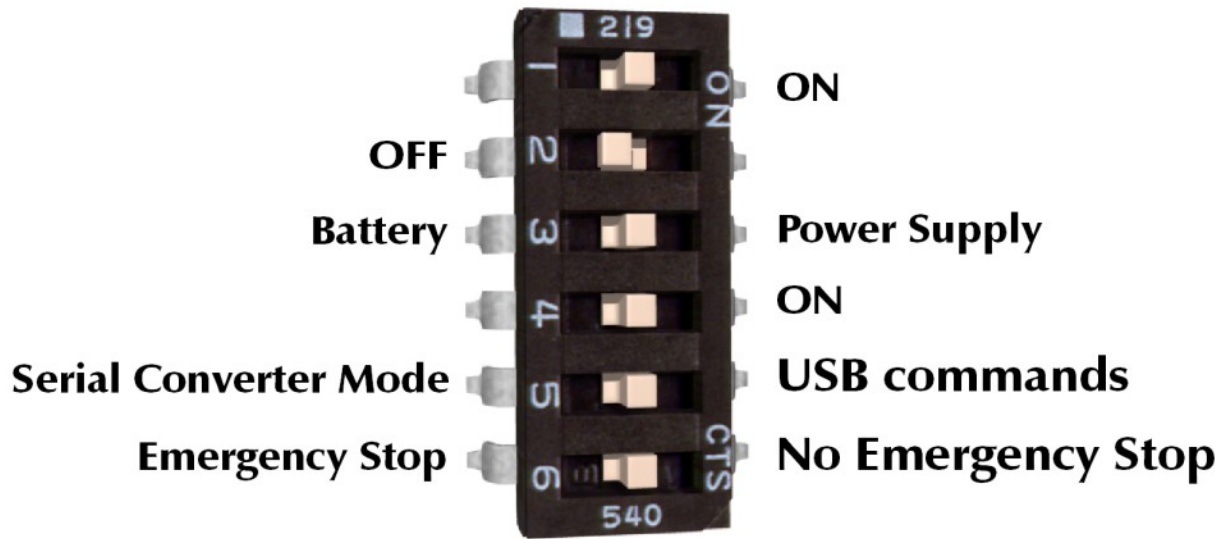
USB mode takes primary input from the USB port, using the same commands as the serial modes. USB mode can also use the Sabertooth 2x32 as a USB to serial converter, to relay commands from the PC to other serial devices, such as another Sabertooth motor driver or a Kangaroo x2 motion control card.



A Sabertooth 2x32 connected to a USB cable

Connect a USB Micro B cable to the USB port on the Sabertooth 2x32. Connect the USB cable to a PC, tablet or single board computer. Even though the signal inputs are not required for control in most USB Modes, they are read as analog inputs. This can be used as a general-purpose analog to digital converter for your system.

USB Mode DIP Switch Settings



DIP switch 4 in the ON position selects USB mode.

USB control uses the same commands as the serial modes. A Sabertooth 2x32 will appear to the operating system both as a serial port and as an HID device. Usually, the serial port is how you will send and receive commands from the Sabertooth. Dimension Engineering supplies an open-source C# library, as well as example code for other programming languages and platforms. For a description of the serial protocol and commands, see the Serial Mode section. USB is the easiest way to interface to PCs or advanced microcontrollers like a Raspberry Pi. Additionally, USB handles its own addressing, collision detection, connection and disconnection detection and checksums, so data corruption is less of a worry.

DIP switch 5 enables or disables USB serial converter mode.

In serial converter mode, you can use the USB port to both send commands to the Sabertooth 2x32 and also relay commands to other Dimension Engineering devices. In this mode, the host Sabertooth 2x32 becomes Packet Serial address 135, to be out of the way of the other devices.

The most common use of this mode is to control a Kangaroo x2 motion control module connected to the Sabertooth 2x32. Closed loop motion commands are sent to the Kangaroo x2. Closed loop positions and velocities are read from the Kangaroo x2 through the Sabertooth. Plug the Kangaroo x2 into the main terminal of the Sabertooth as shown in the Kangaroo X2 manual. Set the DIP switch settings on the Sabertooth to USB Serial Converter mode. Connect a wire from the A1 terminal of the Sabertooth 2x32 to the S2 terminal of the Kangaroo x2. Connect a second wire from the A2 terminal of the Sabertooth 2x32 to the S1 terminal of the Kangaroo x2.

Serial Converter mode can also be used to control additional Sabertooth or SyRen motor controllers. These should be set for packet serial mode, with appropriate packet serial addresses. Connect the A2

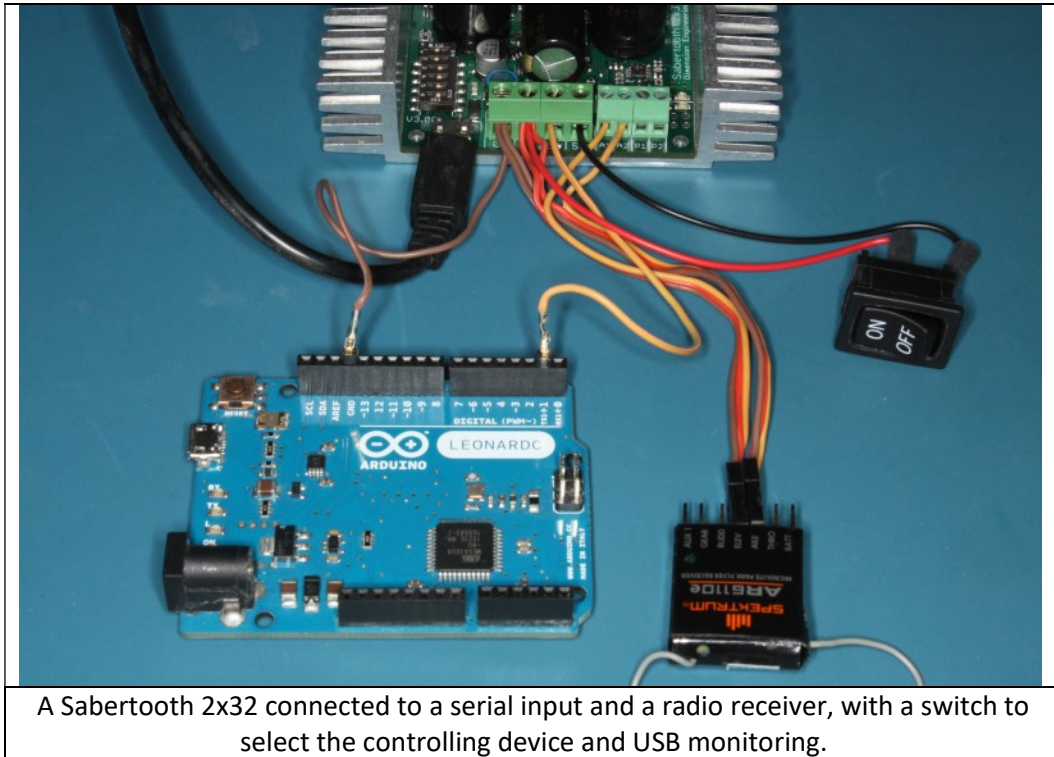
terminal of the Sabertooth 2x32 to the S1 terminal of the additional SyRen or Sabertooth drivers. Connect the A1 terminal to the S2 terminals to enable reading back data, if the device supports it.

DIP Switch 6 enables or disables emergency stops.

If DIP switch 6 is in the OFF position, emergency stops are enabled. This emergency stop is active low and internally pulled down. To enable the M1 output, connect the A1 terminal to 5V. To enable the M2 output, connect the A2 terminal to 5V. If these connections are broken and emergency stops are enabled, the motors will stop immediately. This might be used for a safety E-stop in a machine, to only allow motion while a dead man's switch is held, or to detect a disconnected control cable. Emergency stops provide a second way to shut down the machine if the host computer crashes or locks up, as well as reacting more quickly than a PC might. Hardware emergency stops are required for some safety regulations. Emergency stop is not available when using serial converter mode.

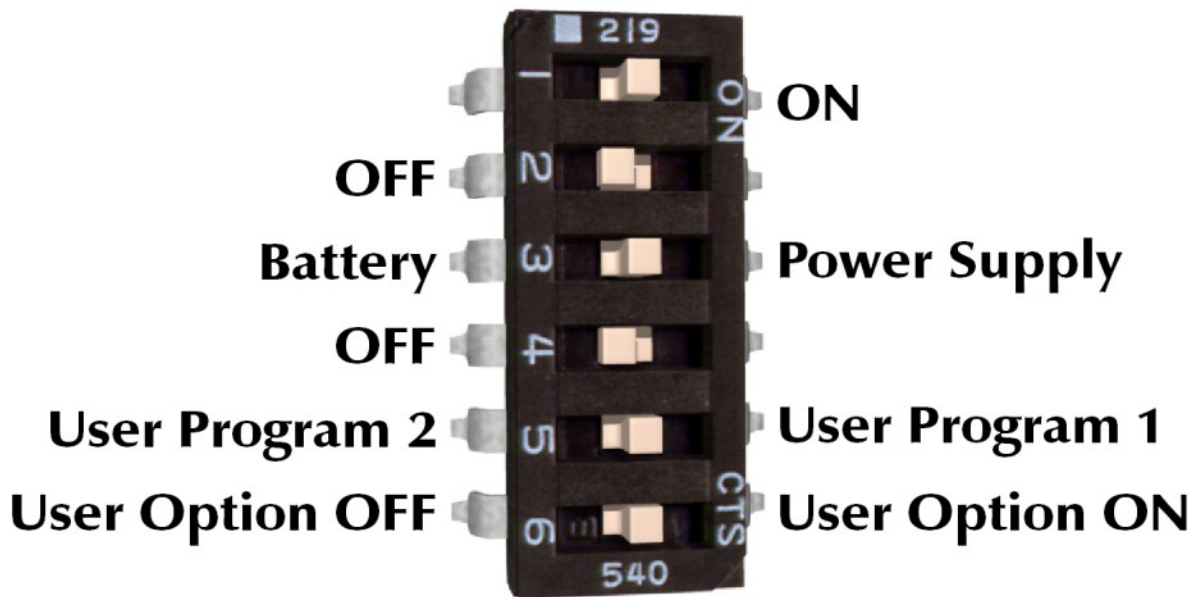
User Mode

User modes are custom operating modes that are created inside DEScribe. They can combine input types and functions to create a Sabertooth tailored specifically for an application. This can cut down on the number of components needed for a product and enable more sophisticated functionality.



In User Mode any of the inputs can be set as any of the input types. Each input should be connected according to its own type.

User Mode DIP Switch Settings



DIP switch 4 in the OFF position selects User Mode.

User modes are custom modes that can be reprogrammed with the DEScribe PC software.

DIP switch 5 selects between User Mode program 1 and User Mode program 2.

Each Sabertooth 2x32 can have two separate User Mode programs. If DIP switch 5 is in the ON position, User Mode program 1 is selected. If DIP switch 5 is in the OFF position, User Mode program 2 is selected.

DIP switch 6 is used inside the User Mode programs.

DIP switch 6 is left available to the creator of the User Mode program, to allow for in-program settings. For example, in a program that controls a conveyor belt, this might be used to select whether the belt runs to the right or to the left.

Default User mode programs

Sabertooth 2x32 ships with two default user mode programs. This is for additional out-of-the-box functionality, and to give programming examples that can be modified.

Default program 1: Serial Autopilot with R/C takeover

Default program 1 would be used with both a microcontroller and an R/C transmitter. Often, you want a robot to run autonomously, except for having the option of manual control for parking, maneuvering in tight spaces, or if the autopilot malfunctions.

S1 is set up as a serial receiver and is connected to an Arduino or similar.

S2 is set up as an R/C input with fixed calibration.

The R/C signal connected to S2 controls whether the motor driver responds to the serial input or the R/C inputs. This is connected to channel 5 on the transmitter, which is a two position switch. In one position, the Sabertooth will respond to the serial commands, in the other it will respond to the R/C signals.

A1 and A2 are set up as R/C inputs with automatic calibration and differential drive mixing.

A1 is the forward/back input and is connected to the elevator channel of the transmitter.

A2 is the right/left input and is connected to the aileron channel of the transmitter.

Default program 2: Analog with remote Indicator LEDs and current display

Default program 2 would be used when it is necessary to have the error indicator on a control panel. Often when building machinery the Sabertooth 2x32 is tucked deep inside a control cabinet, so the onboard LEDs are not visible. Also, for some applications it is useful to display the load current, to prevent loading a mechanism. The power outputs P1 and P2 are used to drive analog panel meters to display load current on each motor.

In this example, the M1 motor is set up bi-direction, with a potentiometer connected to S1 controlling speed and direction.

The M2 motor is set up for separate speed and direction controls. A potentiometer going to A1 controls the output speed, and a switch going between the 5V output and A2 controls the direction.

S2 is set to Indicator output and mirrors the error LED. If the error LED illuminates, an LED connected between S2 and 0V will also illuminate.

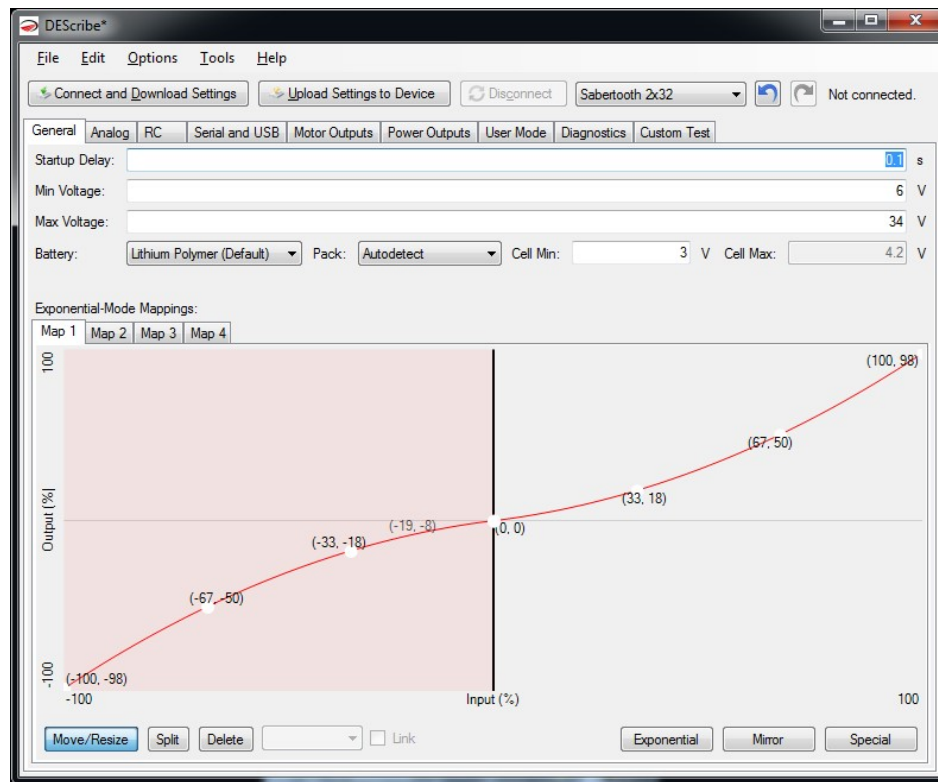
P1 is connected with a 1k pull-up resistor to the 5V output and set as a controllable output, driven by the internal M1 current reading. A 32 amp current reading will drive a full scale 5 volt output. Connect a 5V analog meter between P1 and 0V.

P2 is connected with a 1k pull-up resistor to the 5V output and set as a controllable output, driven by the internal M2 current reading. A 32 amp current reading will drive a full scale 5 volt output. Connect a 5V analog meter between P2 and 0V.

In addition to the example programs, User Mode equivalents for the other operating modes can also be found in the Examples folder.

DEscribe PC software

DEscribe is Dimension Engineering's PC software for adjusting, modifying, monitoring and updating motor drivers, as well as certain other products. It is compatible with Windows XP and newer operating systems. Third generation motor drivers like Sabertooth 2x32 are highly configurable. DEscribe is necessary to access many of the advanced features of these products. DEscribe is a free download, and can be found on Dimension Engineering's website.



Getting Started

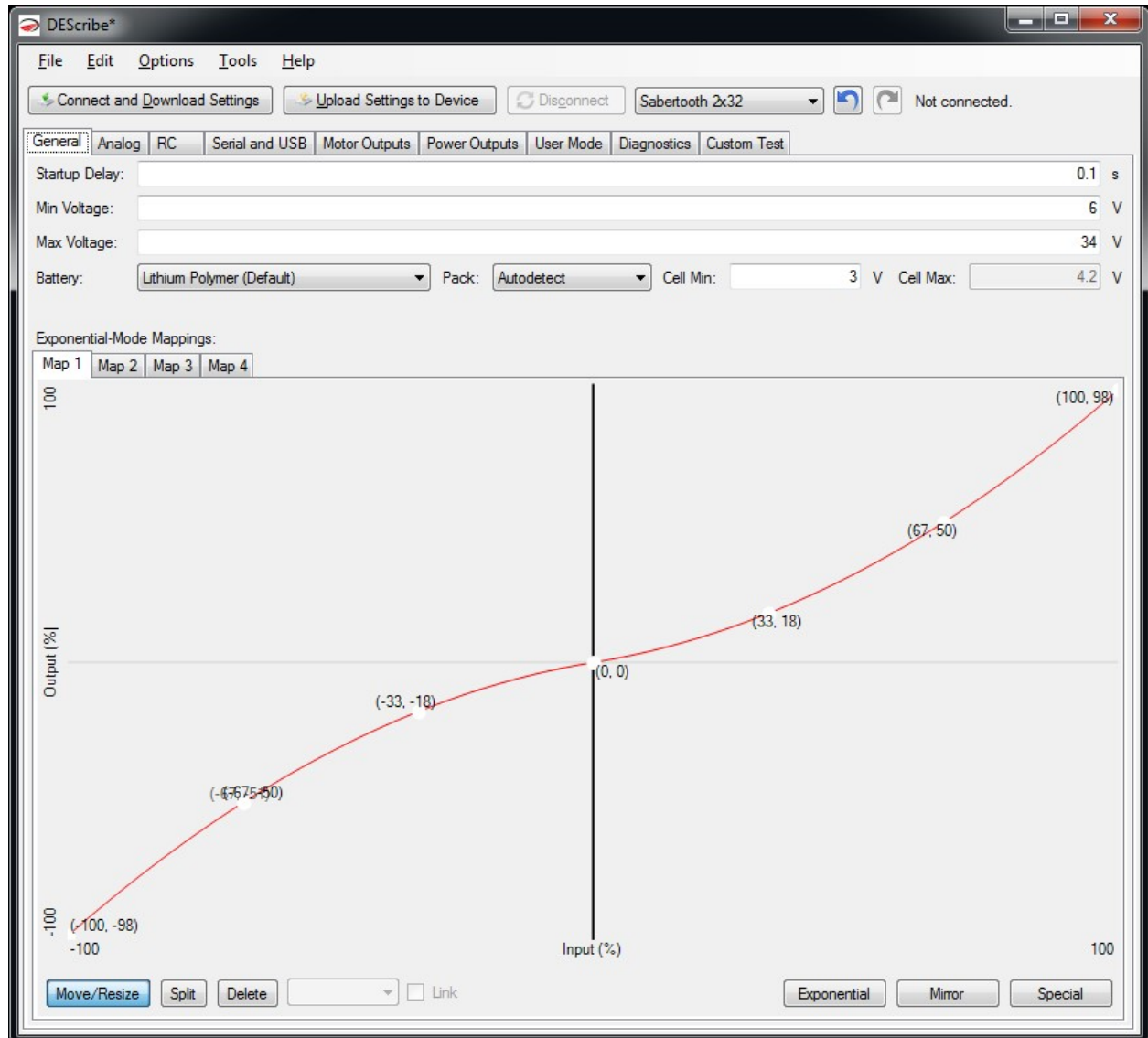
Sabertooth 2x32 is compatible with a micro USB cable. Once you have installed DEscribe, connect one side to the Sabertooth 2x32 and the other side to the PC. You do not have to have a battery, motors or inputs connected, but you may if this is more convenient. Without a battery connected, the green Status LED and red Error LED will flash together when connected to USB. This is normal, and is indicating a low battery voltage. You may communicate with your Sabertooth 2x32 over USB in any operating mode, so you do not need to change the DIP switches.

With the Sabertooth 2x32 connected to USB, launch DEscribe. A quick start screen will appear. Click *Connect and Download Settings* to connect to the Sabertooth 2x32. Generally, you will want to start a DEscribe session by connecting to the Sabertooth. If you do not, then changes you already made could be lost. The *Connect* dialog box will appear. Click the button next to the Serial Port label, and select Sabertooth 2x32. Because Sabertooth 2x32 is a native USB device, no other options are necessary. Click

OK. DEScribe will download the current settings. Any time you want to save the modified settings to the Sabertooth, click the *Upload Settings to Device* button.

General tab

The General tab contains settings that affect the Sabertooth 2x32 in all modes and for all outputs.



Startup Delay

Startup Delay sets how long the Sabertooth 2x32 waits after power is applied to begin operation. This delay is to allow host microcontrollers to boot, receivers to start and analog voltages to stabilize.

Min Voltage and Max Voltage

Min Voltage and Max Voltage set the input voltages within which the Sabertooth 2x32 will operate normally. If the input voltage is outside this range, all motor and power outputs will be off.

Battery

Battery selects the type of battery used and the number of cells. It is also possible to change the low voltage cutoff settings here. These settings go into effect when DIP switch 3 is in the OFF position. See the Battery section for more details.

Exponential-Mode Mappings

Sabertooth 2x32 can have up to four maps which modify the ratio of input signal to output signal. These can be used to add a deadband, change the responsiveness of the system, make the Sabertooth only turn the motors one way, or other uses. Each exponential map is defined by a series of curves. Each curve is defined by control points. By clicking and dragging the control points, you can change the shape of the curve. Which exponential map is used for each output is selected with DIP switches and/or settings in that operating mode's tab in DEScribe.

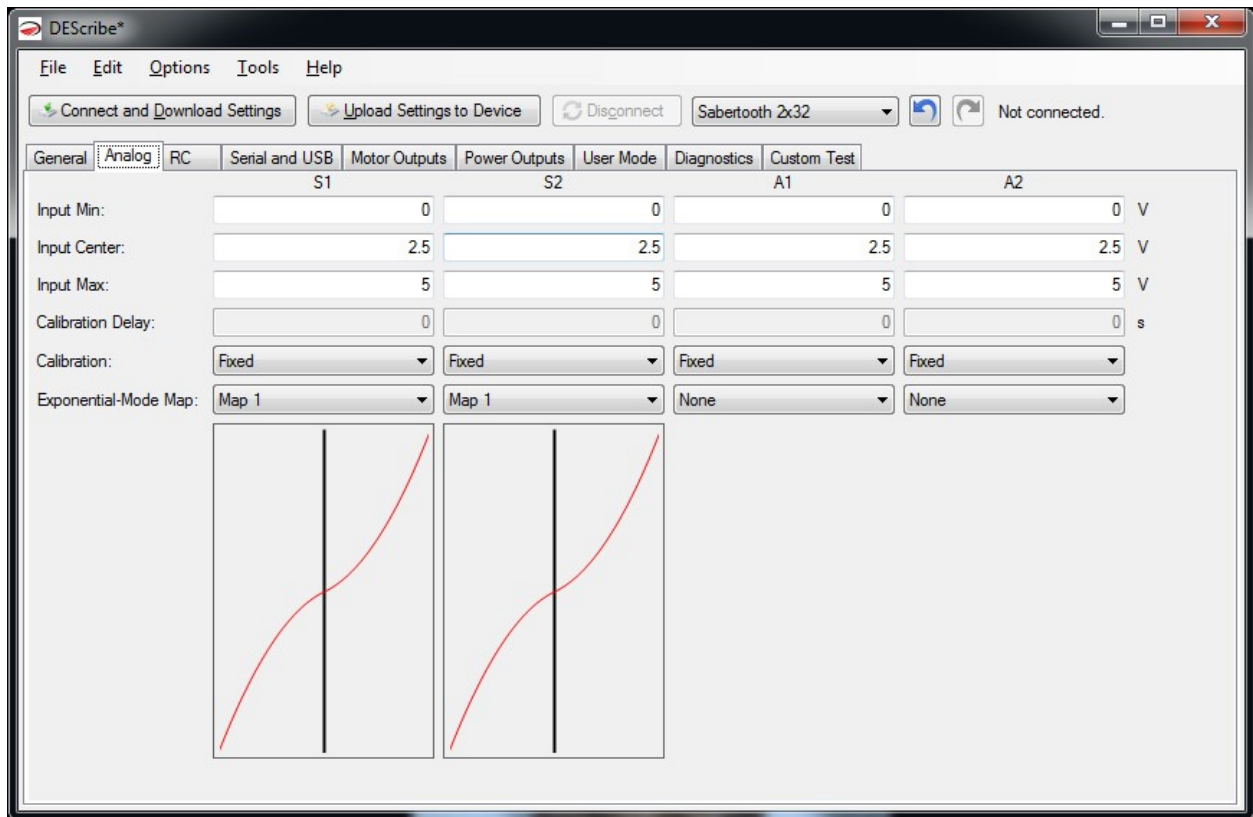
Most of the time, what you want is a certain percentage of exponential and a certain amount of deadband where the motor output is off. These can be handled automatically. Clicking the Exponential button in the bottom right corner of the window will bring up the Exponential Curve window. By dragging the sliders, the currently tabbed exponential map can be modified with little effort.

For more customized maps, you may have to add additional segments. To break a segment into two smaller segments, select the Split tool at the bottom, then click on the curve you wish to divide. The start and end of the curves can be changed by clicking the black vertical divider bars and dragging right or left. To remove a segment, select the Delete tool and click on the segment you wish to remove. To change the curve type, select the segment and change the dropdown from Curve to Linear or Constant. If you have created a custom exponential map, it is often a good idea to save your settings file for re-use later. To save a settings file, click File... Save and create a file name.

There are four maps. By default, Exponential mode uses Map 1 for both channels, so it is usually the only one that needs to be modified. Maps can be modified even if they are not currently being used.

Analog tab

The Analog tab contains settings that are specific to operation in Analog mode. Each of the four analog inputs has the same options. Also, if in a different mode a pin is set for analog input, it is configured here.



Calibration

Depending on what is generating the input signal, different calibration types are most appropriate. Which other settings are active depends on the calibration type selected for the input.

Fixed Calibration

With Fixed calibration, the voltage levels for minimum speed, maximum speed and zero speed are pre-defined and do not change during operation. This would be used with a potentiometer, a DAC or other inputs that are well known and repeatable. Most of the time in Analog mode you will use Fixed calibration. When using Fixed calibration, you can change the Input Min, Input Max and Input Center values. For example, if you are controlling the Sabertooth from a DAC which has a maximum output voltage of 3.0 volts, you would want to change Input Max to 3.0 and Input Center to 1.5.

Automatic Calibration

With Automatic calibration, the voltage levels are re-learned every time the motor driver is powered up. This should only be used with input signals that self-center, such as joysticks. When powered up, after a

delay (the Calibration Delay setting) the Sabertooth will take whatever voltage level is presented at the input and use that as its zero speed setting. It will then constantly look for the largest and smallest voltage that it has ever seen, and use these values as full speed forward and reverse. Automatic calibration is very helpful for signals such as low-cost joysticks that have significant variability in their center and end values. The downside to using Automatic calibration is that if the joystick is not in its center position when the Sabertooth is powered up, or after it loses power for any reason, control will be incorrect.

Joystick Calibration

With Joystick calibration, only the stopped position is re-learned each time the Sabertooth powers up. You can manually change the Input Min and Input Max values. This can prevent slow speed creeping if the joystick does not return exactly to center due to mechanical hysteresis or sticking. Higher quality joysticks will often have a defined output voltage range, usually 0V to 5V, so automatic calibration of the endpoints is unnecessary.

Saved Calibration

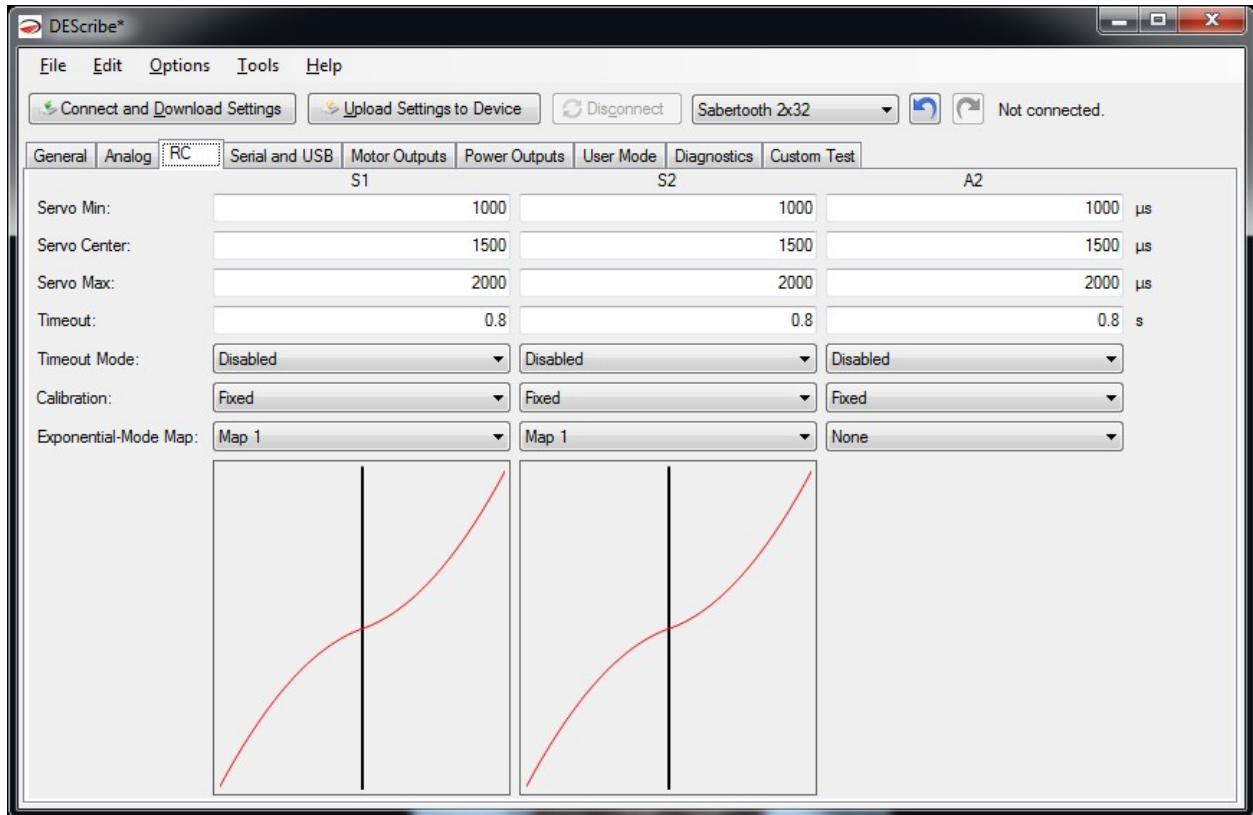
Saved calibration acts exactly like Automatic calibration the first time the driver is powered up after selecting this option, but it then reverts to fixed calibration using the settings it has discovered from that point forward. This is primarily used in production settings, to allow the Sabertooth to learn the specifics of the joystick or control it is mated to, without having the limitations of automatic calibration.

Exponential-Mode Map

To use different maps, select one of the four maps using the drop-down list. A graphical representation of the selected map will appear below the channel. Please note that these maps are only used in Analog mode if DIP switch 5 is set to the OFF position.

RC tab

The R/C tab contains settings that are specific to operation in R/C mode. Because every manufacturer uses slightly different timing for their R/C signals, it is not unusual to have to change the settings to work with a specific transmitter and receiver combination, or a specific microcontroller.



Calibration

Depending on what is generating the input signal, different calibration types are most appropriate. Which other settings are active depends on the calibration type selected for the input. In R/C mode, these settings will only take effect if DIP switch 6 is set to the OFF position. With DIP switch 6 in the ON position, the inputs will always be set for Automatic Calibration.

Fixed Calibration

With Fixed calibration, the R/C pulses for minimum speed, maximum speed and zero speed are pre-defined and do not change during operation. If you are using a microcontroller or other well-defined signal source, you should always use fixed calibration. When using fixed calibration, you can change the Servo Min, Servo Max and Servo Center values. While an Arduino can be programmed to send 1000 to 2000us pulses, a Spektrum DX7 transmitter typically sends pulses ranging from 1200 to 1800us. The Sabertooth 2x32 can be used to read the servo timing of an unknown source by using the Diagnostics tab.

Automatic Calibration

With Automatic calibration, the signal pulses are re-learned every time the motor driver is powered up. This is the normal mode to use with spring-return joystick channels on an R/C transmitter. When powered up, after a delay Sabertooth will take the servo timing and use that as its zero speed setting. It is important to have the joystick centered and the transmitter on when the Sabertooth is powered up. The Sabertooth 2x32 will then constantly look for the longest and shortest signals that it has ever seen, and use these values as full speed forward and reverse. Automatic calibration allows the Sabertooth 2x32 to get acceptable performance from a wide variety of transmitters and receivers. The downside to using Automatic calibration is that if the joystick is not in its center position when the Sabertooth is powered up, or after it loses power for any reason, control will be incorrect.

Joystick Calibration

With Joystick calibration, only the stopped position is re-learned each time the Sabertooth powers up. This can prevent slow speed creeping if the joystick does not return exactly to center due to mechanical hysteresis or sticking. Joystick calibration is rarely used in R/C mode.

Saved Calibration

Saved calibration acts exactly like Automatic calibration the first time the driver is powered up after selecting this option, but it then reverts to Saved calibration using the settings it has discovered from that point forward. This is primarily used in production settings, to allow the Sabertooth to learn the specifics of the transmitter it is mated to, without the limitations of Automatic calibration.

Timeout

This setting controls how long after the receiver stops sending data the Sabertooth 2x32 will shut off the motor outputs.

A note on certain receivers

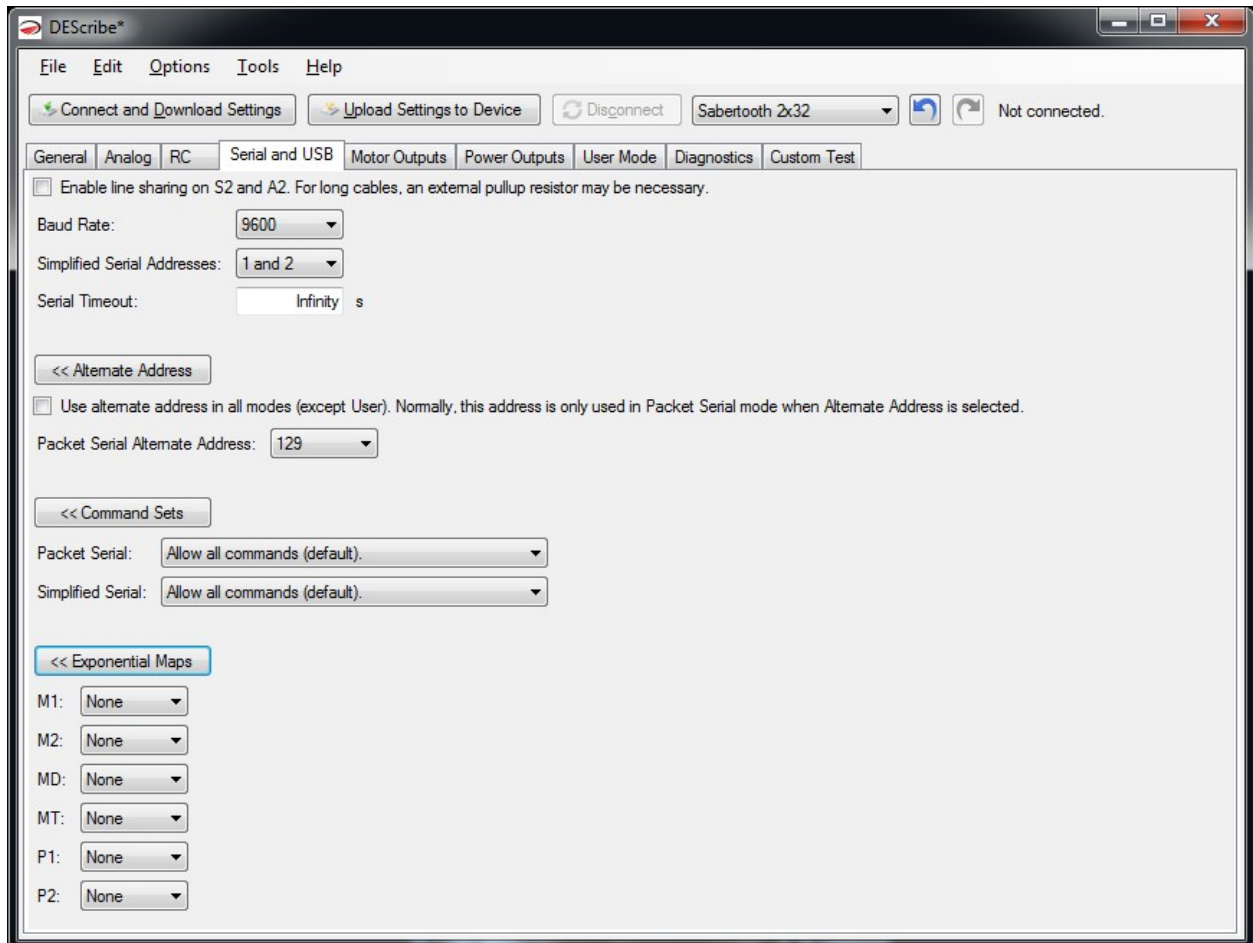
Some receivers will send a “bind” or “failsafe” position before they have gotten any data from the transmitter. The Sabertooth 2x32 will center on the first signal it sees, so if the trim position of the transmitter does not match the bind or failsafe position, your robot may creep forwards or backwards. Typically this can be remedied by re-binding the receiver to the transmitter. Some receivers will continue to put out the last good signal even if communication with the transmitter is lost. This is undesirable with a Sabertooth, because your receiver could continue to send a “drive forward” signal and the rover could run away. Consult your radio’s manual to find out how to disable this “feature.”

Exponential-Mode Map

To use different maps, select one of the four maps using the drop-down list. A graphical representation of the selected map will appear below the channel. Please note that these maps are only used in R/C mode if DIP switch 5 is set to the OFF position.

Serial and USB tab

The Serial and USB tab contains settings that affect both Serial mode and USB Mode. USB and Serial mode use the same communication protocols.



Enable Line Sharing

This setting allows multiple Sabertooth motor drivers to be used with the same serial port. Instead of always being driven, the transmit lines back to the host processor are only driven while data is being sent.

Baud Rate

This changes the baud rate for the serial modes. Options are 2400, 9600, 19200, 38400 and 115200 baud. The default is 9600. The baud rate setting of the Sabertooth must match the setting of the host microcontroller.

Simplified Serial Addresses

Because Plain Text Serial does not have an address byte, this is used to select which motor driver is being addressed. For example, a first Sabertooth could be set to 1 and 2, and a second Sabertooth set to use addresses 3 and 4. In this configuration, a command of M1: 500 will command only the first

Sabertooth's M1 output to turn. A command of M3: 500 would control the M1 output of the second Sabertooth.

Serial Timeout

If the Sabertooth has not gotten a new command within this amount of time, it will assume the host program has locked up and shut down the motors. Sending a new valid command will cause the motors to restart. A setting of **Infinity** means the motors will run at the last commanded speed forever.

Alternate Address

This option is used to set the packet serial address of the Sabertooth. This is used in conjunction with DIP switch 5 in Serial mode. You can also force this address in all modes with the checkbox.

Command sets

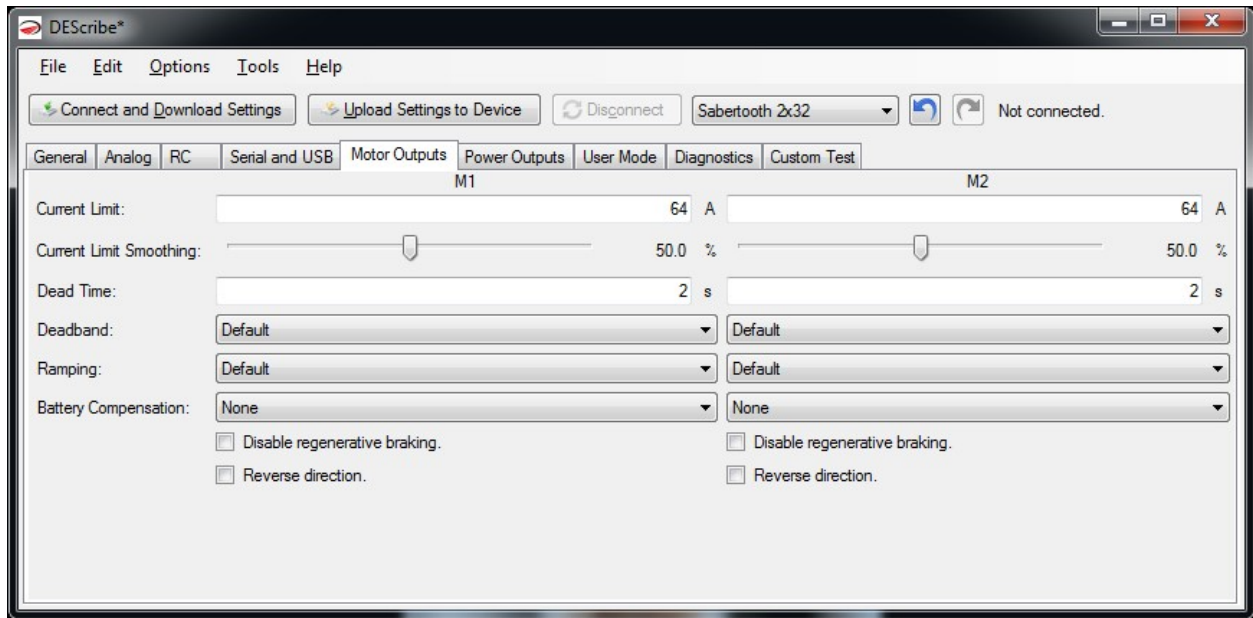
These options tell the Sabertooth to use or ignore certain command sets. These would typically be set for robustness if you are using CRC protected packets.

Exponential

These cause selected serial commands to be processed through the exponential maps. This can reduce the load on the host processor.

Motor Outputs tab

These settings affect the operation of the motor outputs in all modes.



Current Limit

This setting sets the maximum current for the motor output. If the system tries to draw more current than the setting, the output duty cycle will be reduced. For Sabertooth 2x32, each motor output can be limited individually in a range between 0A and 64A. This is a soft current limit, and will slow acceleration but allow motor motion to continue. There is also a fixed hard current limit above approximately 70 amps which protects the Sabertooth from a shorted motor.

Current Limit Smoothing

This setting changes how quickly the current limit responds. More smoothing will cause the Sabertooth to ignore small overcurrent spikes for smoother operation.

Dead time

This is the length of time that the motor will stay in a regenerative braking mode after a stop command is sent. After this delay, if the motor is still being commanded to stop, the motor output will shut down to conserve power. Any command that would move the motors will immediately wake the Sabertooth from this shutdown state.

Deadband

This is the range of input signals that the Sabertooth considers to be a zero speed input. If the setting is left at **Default** the setting varies depending on the operating mode. For example, in Analog mode the default deadband is from 2.375 to 2.625 volts. If the setting is changed to **Custom**, text boxes will appear. Enter your desired deadband in these fields and this deadband setting will be used in all modes.

Ramping

This setting is the amount of time the controller will take to go from full reverse to full forward. If the setting is left as **Default**, then the value depends on the operating mode. The default in Analog and R/C mode is for the ramping to be controlled by an analog signal sent to A1. The default in Serial and USB mode is for the ramp timing to be controlled by a serial command. Changing the selection to **Custom** allows for a pre-defined, fixed ramp speed. The Ramping time is the time it would take to go from a full reverse command to a full forward command.

Battery Compensation

This is a new feature in the third generation Sabertooth motor drivers. As batteries discharge, the voltage they supply declines. A 12v lead acid battery will supply 13.5 volts fully charged, and 9 volts fully discharged. What this means is your system will run faster on a full battery than a depleted one. Sabertooth 2x32 is always measuring the input voltage, so it is able to compensate for this change, resulting in more consistent operation. This is especially helpful with closed loop control.

Battery compensation can only decrease the output, so the Compensated Voltage must be less than or equal to the minimum expected input voltage. Battery compensation is less efficient than using the proper input voltage directly. A Sabertooth with a 24V input compensated to 12v will run hotter than if it was running from a 12V input. The motor will also run less efficiently.

Fixed Battery Compensation

When fixed battery compensation is selected, the duty cycle will be scaled such that full speed forward or reverse puts out the Compensated Voltage, regardless of the input.

Automatic Battery Compensation

Automatic compensation uses the battery cutoff voltage defined in the General tab as the compensated voltage. For example, a 3s lithium polymer battery with a cutoff of 9V will cause the system to respond as if it is always being supplied 9 volts.

Disable Regenerative Braking

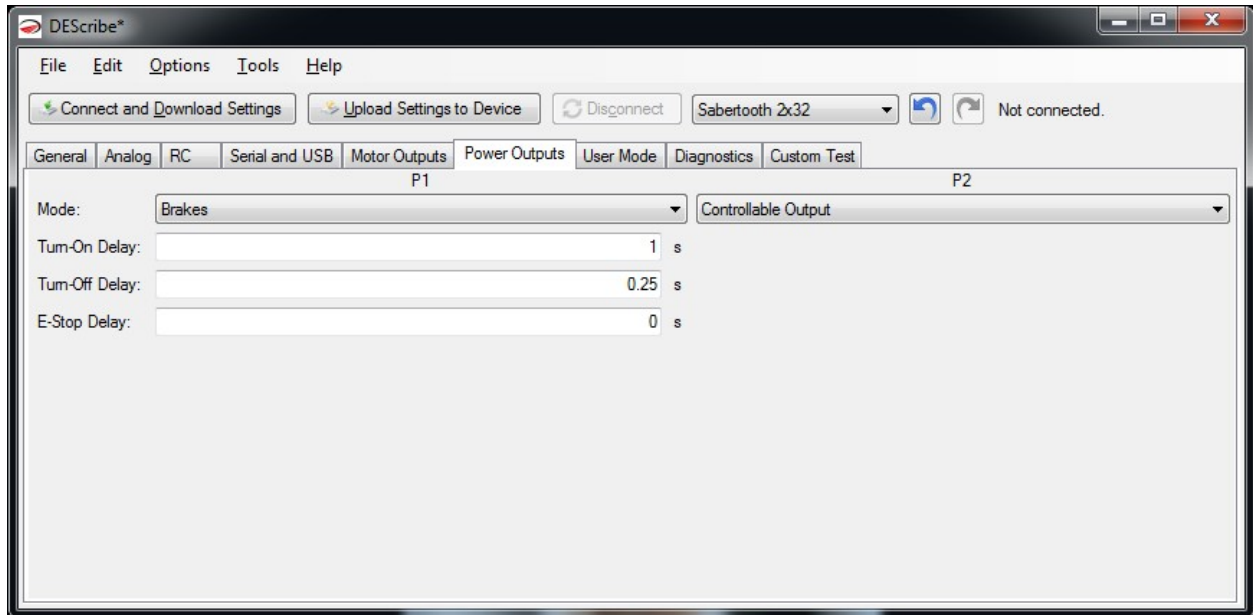
This option disables regenerative braking. This is typically used to drive plating tanks, Peltiers or other loads that are not motors. A motor driven this way may run much less efficiently. Motors driven this way will also stop less rapidly. If you are running motors, but the system runs from a power supply, it is usually better to leave regeneration enabled and use the power outputs P1 or P2 as regenerative voltage clamps.

Reverse Direction

This check box will reverse the motor output for this channel. Now instead of assembling a complicated project only to find one motor is turning backwards and having to change the wiring, you can solve the problem in software.

Power Outputs tab

The power outputs tab controls the behavior of the power outputs P1 and P2.



Mode

The power outputs must be put in the correct mode in order to have the expected function.

Voltage Clamp

In this mode, the power output P1 or P2 will turn on if the input voltage rises. This rise is due to regenerated energy being fed back into a power supply that is not able to accept it. Connecting a power resistor between the power output and the positive supply voltage and using this mode allows for operation from a power supply. **Max Voltage** is the voltage above which the output will turn on. Automatic compares the present input voltage to the average input voltage, and works well in most cases. Moving the slider allows for a custom turn-on voltage.

Brakes

In this mode, the power outputs are used to operate electromagnetic brakes. P1 corresponds to motor 1 and P2 corresponds to motor 2. When the motor stops, the brakes activate after the **Turn-On Delay**. When the motor is commanded to move again, the brakes deactivate immediately, but the motor does not turn until the **Turn-Off Delay** has elapsed. If an emergency stop is commanded, the brakes will activate after the **E-Stop delay** has elapsed.

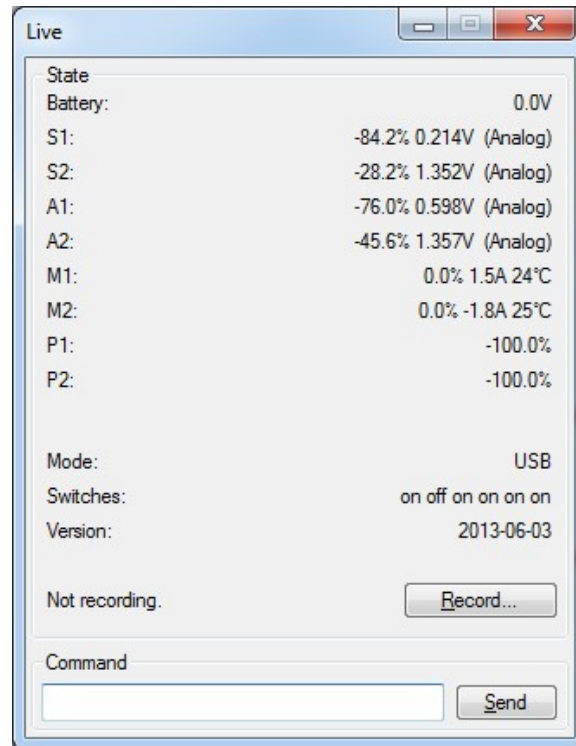
Brakes mode can also be used to drive the field of a shunt or compound wound motor. To do this, connect one side of the field to B+ and the other side of the field to the P1 or P2 power output. Turn all the timings to 0 seconds.

Controllable Output

In this mode the power output is directly controllable. A control setting of 2047 corresponds to full power and a control setting of -2047 corresponds to zero power.

Diagnostics tab

DEscribe can be used to monitor the inputs and outputs of the Sabertooth 2x32 during operation. Click on the **Show Inputs/Outputs** button to bring up the **Live** monitor window.



Battery shows the current input voltage.

The signal inputs **S1**, **S2**, **A1** and **A2** show both the raw input value and the processed percentage that input value corresponds to. This is useful for setting or debugging input calibration.

The motor outputs **M1** and **M2** show the output duty cycle, output current and motor driver temperature.

The power outputs **P1** and **P2** show their current output percentage. Remember that -100% corresponds to an off state, and 100% is full on.

Mode will read the DIP switches and display the current operating mode and options.

Switches reads the DIP switches, which is useful to diagnose if something is set incorrectly.

Version shows which firmware revision is currently loaded onto the Sabertooth 2x32.

Recording will save the monitor values into a .csv file, which can be opened as a spreadsheet. This is useful to catch transient errors or collect long-term data automatically.

If you are in USB mode, **Command** can be used to enter Plain Text Serial commands for debugging purposes. Type the command into the text box, then click the **Send** button to send the command.

Tools menu

The Tools menu at the top of the window contains an assortment of tools to make developing with your Sabertooth 2x32 easier.

Calculator is a basic calculator program.

Serial Terminal is automatically connected to the Sabertooth 2x32. It can be used to send serial commands and read responses.

System Information is used to determine information about your operating system and PC. This is useful if there are compatibility issues with either the Sabertooth 2x32 or DEScribe

Update Firmware is used to update the firmware on your Sabertooth 2x32. This will allow new features to be added or bugs to be fixed without having to send it back to Dimension Engineering.

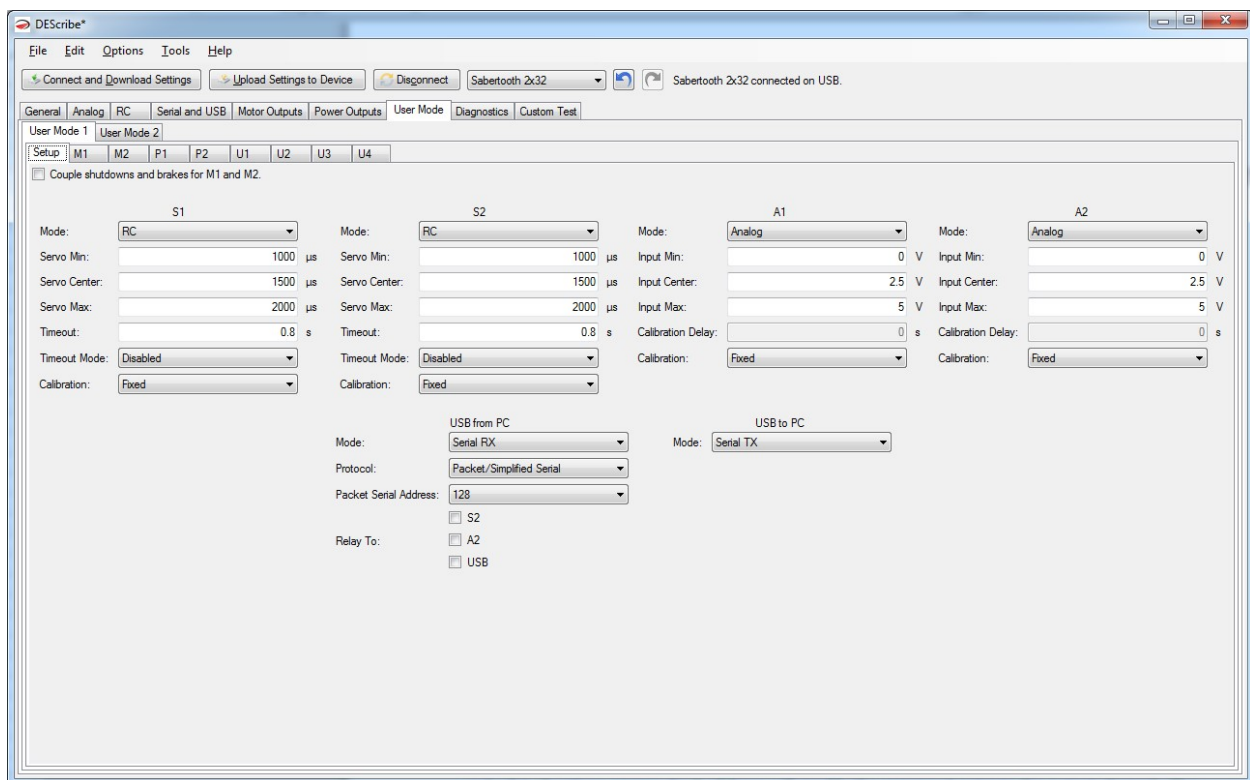
Encoder Calculator is used with quadrature encoders to calculate distance travelled per encoder count. This is most useful with the Kangaroo x2 motion control add-on.

User Mode

A major advance in the third generation Sabertooth motor drivers is the ability to have user-defined operating modes. These are set up using a graphical scripting language in the DEScribe PC software and uploaded over USB. In many cases, this removes the need for an additional microcontroller entirely. For example, you could have main control be from a PC over USB, with a radio control override, or an analog mode that only operates when the A2 input is between 2.2 and 2.5 volts.

Setup tab

To create a custom mode, the first thing that must be done is set up the inputs. On the Setup tab, select how you would like each of the six inputs configured. If you will not be using that input, you may leave it set as the default.



Signal Inputs

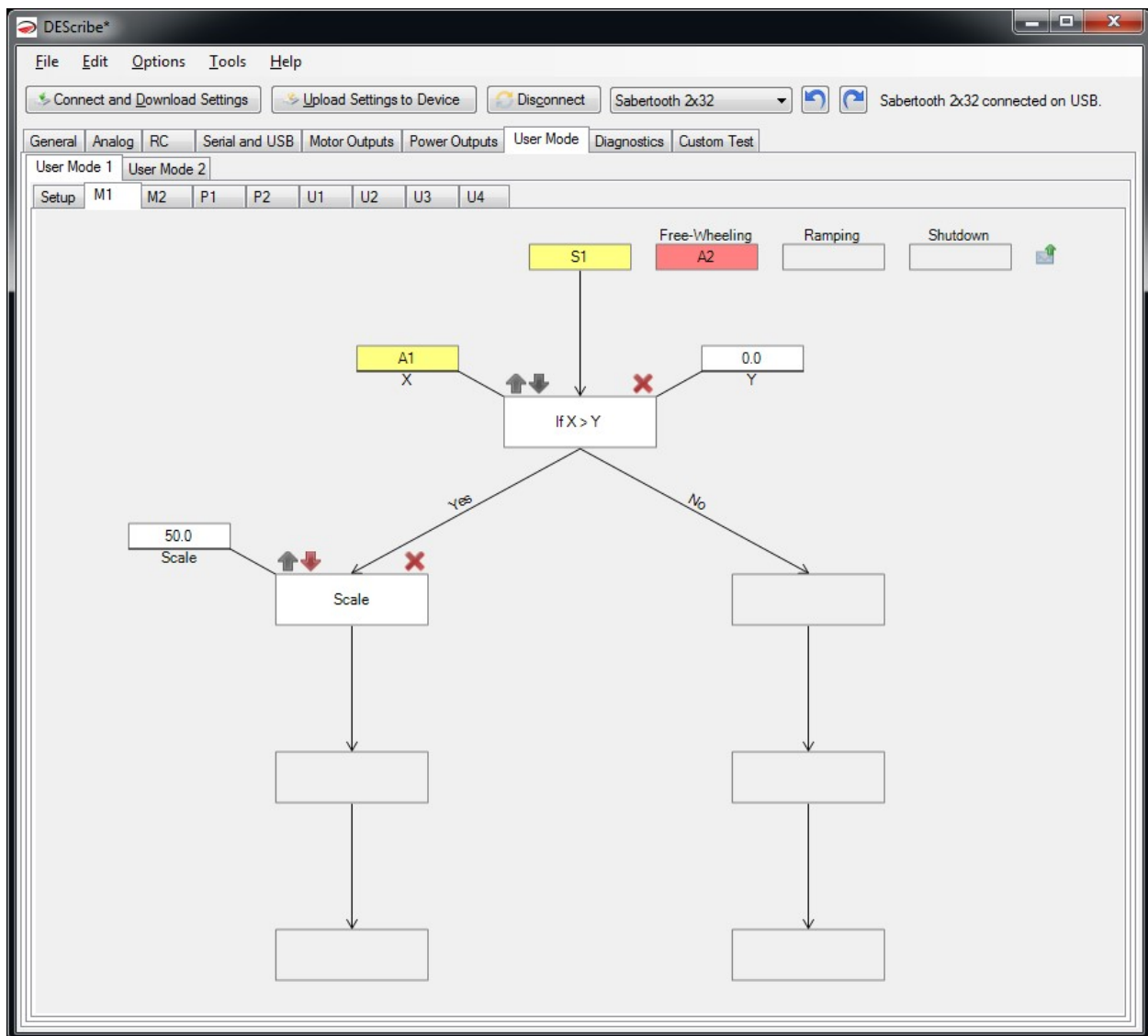
The options for each input will change depending on the Mode selected. These options are the same as for that mode's tab (Analog, R/C or Serial).

Indicator Outputs

Additionally, S2 and A2 can be configured as on/off 20 milliamp logic outputs. This is used to drive external LEDs or signal to a host processor. If the mode is set to Indicator, you must select the source that controls it.

Program tabs

Now that the inputs are set up, a program is created for each output. For example, the M1 tab holds the program that controls the M1 motor output. Each program starts with an initial value read either from an input or from the output of another program. Each program can have up to four commands. Click on the box below the initial input and a list of options appear. Each of these will modify the command going to the output channel. The commands operate in succession, following the arrows. After the last operation, the value is sent to the output.



The program shown would set up an R/C dual rate switch. The program starts with an R/C signal on the S1 input. If the R/C A1 input is positive, then the output is reduced by 50%. If not, then the signal passes through unmodified and directly controls the M1 output.

Program commands

There are ten types of operations available in user mode. Each changes the output or changes program flow.

Each operation takes one of the four block spaces in the program. To add a command, click on a blank box. To remove a block, click the red X above the block. To change or modify a block, click on the block itself and a menu will appear in front of the block. To move a block up or down in the program, click the up or down arrows.

If

The If operation is used to make the Sabertooth do one thing or another, depending on state. For example, you could make it either use USB control or an R/C signal coming into s1, depending on the state of a radio control input into S2.

The If operation takes two additional inputs, which are labeled X and Y. By clicking on the X or Y block, you can select any if the input or internal signals available.

The If operation can perform standard logical comparisons. To change which comparison it is performing, click on the center If block and then select Condition from the list that pops up.

If the comparison is true, the blocks below the Yes arrow will be run. If the comparison is false, the blocks below the No arrow will be run.

Set

The Set operation sets a new value for the output channel, ignoring any values coming into it. This is often used after an If operation. For example, if the temperature is above 10 degrees, operate. Otherwise, **set** the motor outputs to zero.

The Set operation takes one additional input, labeled Value.

Calibrate

The Calibrate operation is similar to the calibration applied to analog and R/C signals. It is used to set a new input to output mapping. Calibrate is computationally intensive and can cause your program to run slowly. For this reason, it is better to calibrate your input signals in the Setup tab and use the Calibrate block only where necessary.

The Calibrate operation takes five additional inputs, labeled Input Min, Input Center, Input Max, Output Min and Output Max.

Flip

The Flip operation reverses the polarity of the signal. For example, if the motor is running forward without the Flip operation, it will run at the same speed backwards with Flip applied.

The Flip operation takes no additional inputs.

In order to save blocks, you can also specify any signal as flipped when it is set. To do this, click on the input box and select Flipped from the menu that pops up.

Map

The Map operation applies one of the custom input-to-output mappings, which are set up in the general tab, to the output channel. This is used for exponential response, or to have a custom range, or to have a user-defined response.

The Map operation takes no additional inputs. To change which of the maps (1 through 4) are being applied, click on the Map block, click Map > at the top, and select the desired map number.

Mix

The Mix operation is used to create differential drive outputs. Mix can also be used to add or subtract signals.

The Mix operation takes two additional inputs. The program flow input (arrow into the top of the block) is the forward/reverse channel and the Mix With input on the left of the block is the right/left channel. The Direction input selects whether this is the left or right wheel.

Scale

The Scale operation is used to scale the motor speed. Typically this would be used for a max speed setting, or reduced sensitivity. It can also be used to multiply one signal by another. You can also make signals bigger by scaling with a fixed value greater than 100%.

The Scale operation takes one additional input, labeled Scale. This is the value by which the original value is scaled. Remember that signals are percentages, so if one signal is scaled by another, the result will end up smaller than the input signals.

Driver Scale

The Driver Scale operation will convert an arbitrary input range into an output of -100% to 100%. This is useful to normalize signals.

The Driver Scale operation takes one input, labeled Source.

Set Bounds

The Set Bounds operation limits a signal to between a user-defined minimum and maximum value. This is different from Scale in that until the bounds are exceeded, the input signal is not modified.

The Set Bounds operation takes two additional inputs, labeled Min and Max.

Set Range

The Set Range operation is used to set a new input or output range for the signal. For example, to make a single direction output, you would use Set Range to change the output range from [-100, 100] to [0, 100].

The Set Range operation takes four additional outputs: Input Min, Input Max, Output Min, and Output Max.

Special considerations

User Scripts U1 through U4

Sometimes while creating a user mode, you will find that you either need more than 4 blocks, or that you need to store an intermediate value that will be used by more than one output. This is what the user scripts U1 through U4 are for. There are a few important points to remember.

Scripts are processed in the following order: U1, U2, U3, U4, M1, M2, P1, P2.

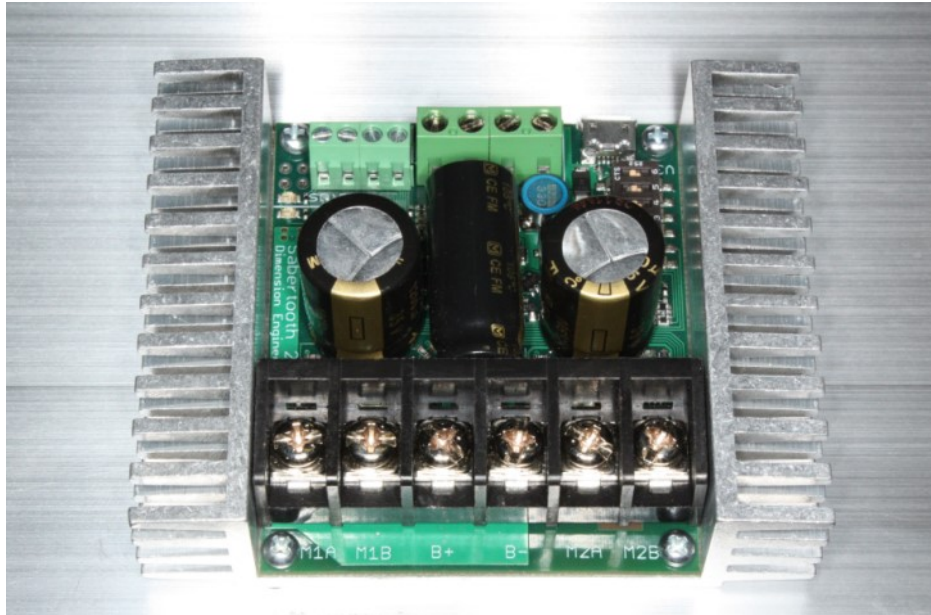
User scripts do not have free-wheeling or shutdown inputs, as those are specific to the motor and/or power outputs. They can have ramp rates.

Output Scripts P1 and P2

It is important to remember that in order to use the P1 and P2 scripts as outputs, the Mode must be set to Controllable Output in the Power Outputs tab of DEScribe. Otherwise the default operating modes such as Voltage Clamp or Brakes take precedence.

Mounting

Sabertooth 2x32 has four mounting holes, which are sized for 4-40 machine screws. 5/8" Phillips machine screws are provided, but other lengths can be used as well. Sabertooth 2x32 has a single-piece machined heat sink. The heat sink features a pin-fin design, so it can be mounted in any orientation with acceptable thermal performance.



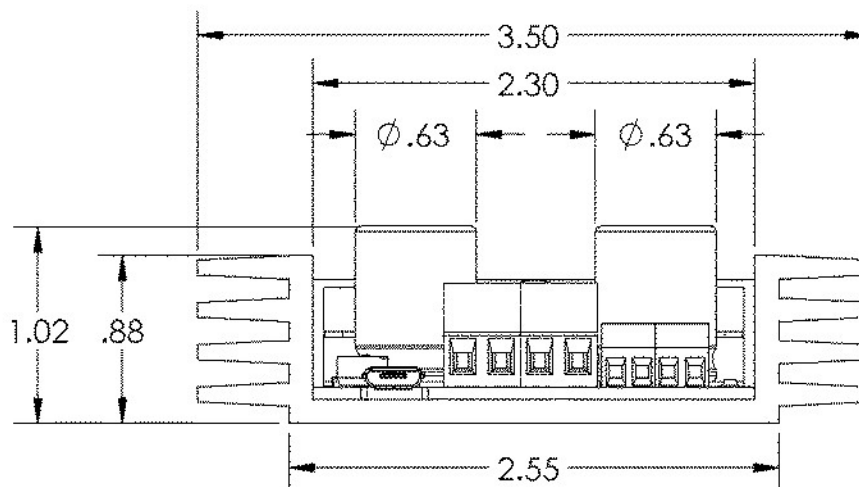
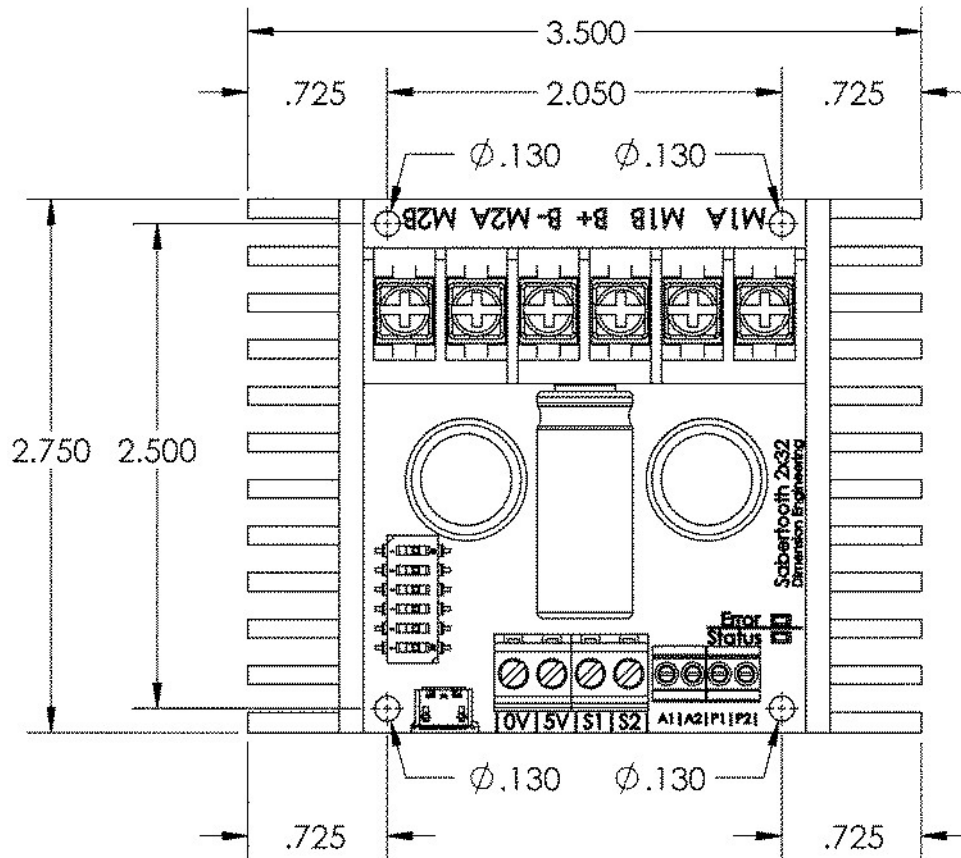
The heat sink is not electrically connected to the electronics of the Sabertooth 2x32. If there is a metal chassis available the best thermal performance will come from mounting the flat bottom of the heat sink directly to the chassis as shown in the photograph above. If the driver is mounted to an insulating material such as wood or plastic, it may be better to use standoffs to allow better cooling.

Because Sabertooth 3x32 is a high power component, capable of supplying up to two kilowatts of continuous output power, it will get warm in use. If there is airflow available in the design, having the Sabertooth 2x32 in that airflow will lead to lower board temperatures. At a very minimum, allow some space for air movement between the edges of the heat sink and the enclosure.

There is a printable full size mounting diagram at the end of this manual which can be used to mark the location of the mounting holes. There are also CAD models of the Sabertooth 3x32 available for download on Dimension Engineering's website.

Drawings

3D models in IGES, STL and Solidworks formats are also available on Dimension Engineering's web site.



Printable mounting diagram

The following image is the same size as the Sabertooth 2x32. If this page is printed, it can be cut out and used as a template to drill the mounting holes.

Remember when mounting to leave space for the input and output wires, and especially for a micro USB cable, if used. These are included for reference in the mounting diagram.

