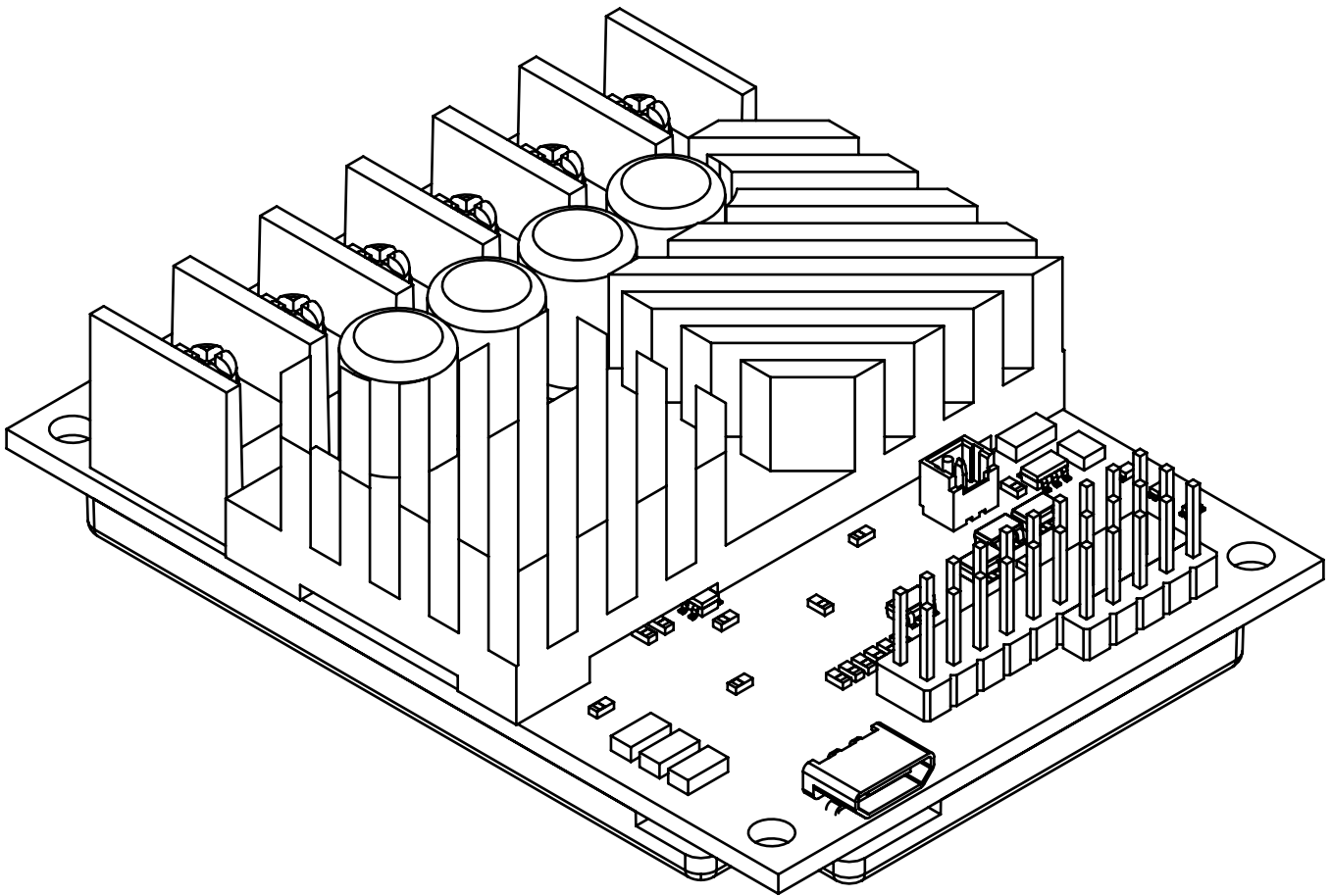




# BASICMICRO MOTION CONTROL

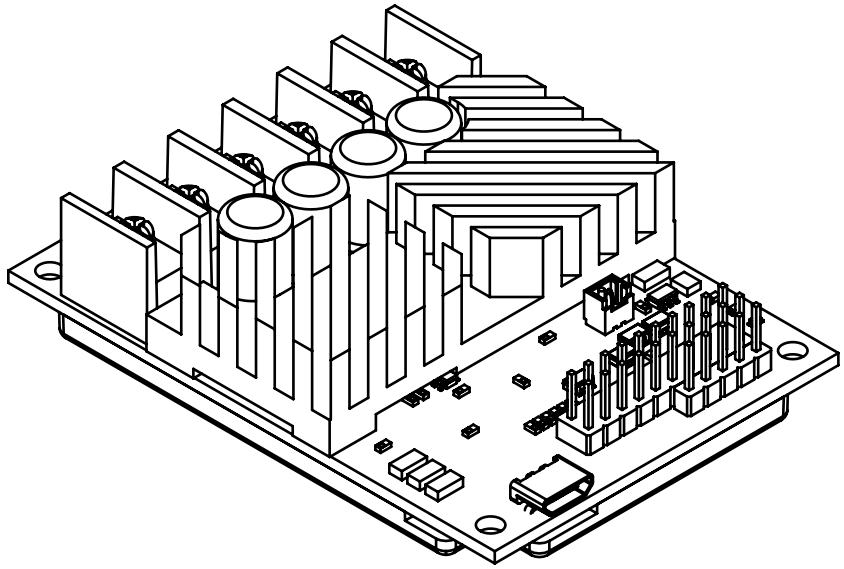


**RoboClaw 2x30A, 34VDC Dual Channel  
Brushed DC Motor Controller**

**Data Sheet Version 3.0**

## Feature Overview:

- 30 Amps Continuous Per Channel
- 60 Amps Peak Per Channel
- Dual Quadrature Decoding
- 9.8 million PPS Decoding
- Multimode Interface
- TTL Serial
- USB Port
- Analog Interface
- R/C Input Control
- Limit, Home and E-Stops
- Up to 34VDC Operation
- Air Cooled
- Cooling Fan Control
- 3.3v Compliant Control Outputs
- 5v Tolerant Control Inputs
- Programmable Current Limiting
- Programmable Voltage Clamping
- Closed and Open Loop Operation
- Auto Tuning PID Feature
- Mixed Control Modes
- Data Logging
- Diagnostic LEDs
- Field Firmware Updates
- Regulated 5VDC, 3A User Available Output
- Over Voltage and Under Voltage Protection
- Easy Tuning, Monitor and Setup with PC utility



## Device Overview

The RoboClaw is an intelligent, high performance motor controller designed to control dual brushed DC motors. It can be controlled from USB, RC radio, PWM, TTL serial, analog and microcontrollers such as an Arduino or Raspberry Pi.

RoboClaw supports 3.3V or 5V logic levels, travel limit switches, home switches, emergency stop switches, power supplies, braking systems and contactors. A built-in switching mode BEC supplies 5VDC at up to 3 Amp for powering user devices. In addition power supplies can be utilized by enabling the built in voltage clamping control feature.

A wide variety of feedback sensors are supported. This includes quadrature encoders, potentiometers and absolute encoders which can be easily configured using the available auto tune function. With sensors, two brushed DC motors can be controlled in closed loop mode allowing precise control over position and speed. With the ability to use potentiometers, servo systems can be created and controlled from any of RoboClaw's interface modes.

For greater control, built-in commands are available for controlling acceleration, deceleration, distance, speed, current sense, voltage limits and more. In addition, RC and analog modes can be configured by user defined settings to control acceleration and deceleration rates.

RoboClaw incorporates multiple protection features including temperature, current, over voltage and under voltage limits. The protection features are self monitoring and protect RoboClaw from damage in any operating condition. User definable settings such as maximum current limit, maximum and minimum battery voltages are provided for more refined control.

RoboClaw's regenerative capabilities will charge a supply battery during slow down or braking. It's advance circuitry can change direction during full throttle without damage! RoboClaw also incorporates a LiPo cutoff mode to prevent battery damage.

## Multimode Interface

RoboClaw's I/O are voltage protected and can handle up to 5VDC. The I/O only output a high of 3.3V. This allows RoboClaw to be interfaced to 5V or 3V logic easily with no translation circuits required. RoboClaw can be connected directly to a Raspberry Pi or Arduino. All of RoboClaw's inputs are internally pulled-up to prevent false triggers. Inputs can also be configured using the Motion Studio application.

### **User Regulated Power Output**

RoboClaw provides regulated power (BEC) for user devices. A high efficiency switching regulator supplies 5VDC at up to 3 Amps. This voltage can be used to power external sensors, encoders, MCUs and other electronics. The regulated user power is automatically current limited and thermally protected.

### **Main Battery**

The peak operational input voltage can be up to 34VDC. The models maximum input voltage can not be exceeded. If the maximum voltage is exceed damage can occur. Fully charged batteries maximum voltage must be taken into account when in use. RoboClaw is a regenerative motor controller. During regeneration, voltages can peak over the maximum rated voltage and must be managed properly. Do not disconnect the negative terminal if other ground sources are connected, such as USB cable.

### **Logic Battery**

RoboClaw accepts a logic battery. The logic battery is also known as a backup battery. The user regulated power output (BEC) is by default powered from the main battery, unless a logic battery is detected. The logic battery source is coupled to the main battery through an on board automatic switch. If the main battery voltage drops below the logic battery input level, the logic circuit and user regulated power output will be drawn from the logic battery.

### **Software**

RoboClaw can be easily configured using the Motion Studio software tool. The Windows based application enables users to quickly configure RoboClaw. The software can be used during run time to monitor and control several operational parameters. Motion studio is available from the Basicmicro.com website. It can also be found in the Downloads section of the Basicmicro website or listed under the Download tabs on the production page.

### **User Manual**

This data sheet only covers model specific information and basic wiring. To properly setup and use RoboClaw refer to the RoboClaw User Manual available for download from <http://www.basicmicro.com>.

### **Cooling**

RoboClaw will generate heat. The maximum current ratings can only be achieved and maintained with adequate heat dissipation. The motor controller should be mounted so that sufficient airflow is provided. Which will dissipate the heat away from the motor controller during operation. Some models of RoboClaw include a built-in automatic cooling fan controller, which can be used to help maintain continuous currents under extreme conditions.

### **Emergency Stop**

The motor controller should be wired using an external contactor, relay or high amperage mechanical switch to control the main power input. A second power source should be used to power the logic section in situations where the main power will be under heavy load. Voltage drops can occur from constant full load or high speed direction changes. Voltage drop can cause logic brown outs if only a main battery is used without a logic battery.

### **USB**

A dual ferrit core USB cable is required. The USB port should be used for configuration and debugging. The USB protocol is not designed for electrically noisy environments. The USB port will likely disconnect and not automatically recover during operation in electrically noisy environments. To recover from a dropped USB port, the motor controllers USB cable may require being unplugged and re-plugged in. The TTL serial control should be the preferred method of control in electrically noisy environments.

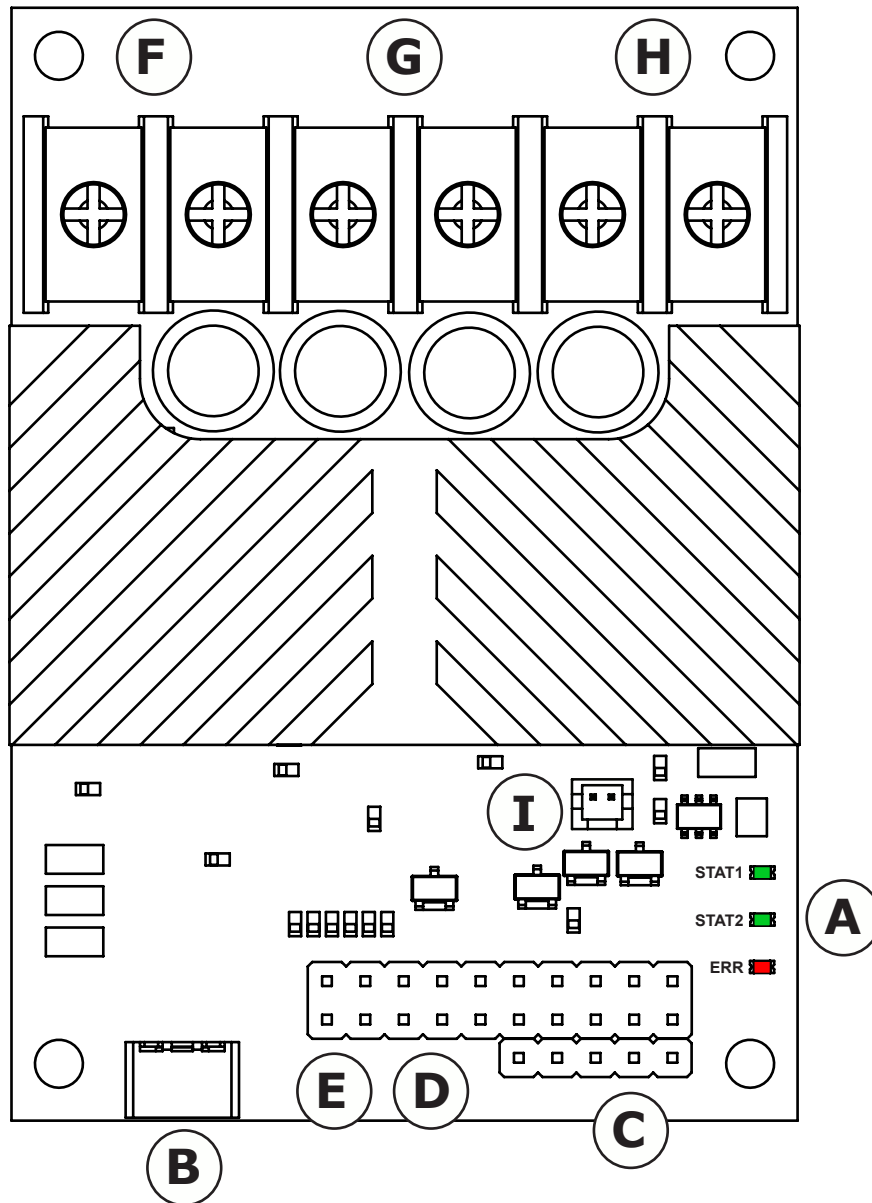
### **Firmware Updates**

Firmware updates will be made available from time to time. Updates may add new features or resolve technical issue. It is important before using RoboClaw for the first time, to update to the latest firmware. Download and install Motion Studio with the USB driver. Motion Studio will check for firmware updates. Refer to the RoboClaw User Manual or Application Notes for additional information on updating the RoboClaw firmware. RoboClaw must be extrnally power. It wil not power from the USB cable.

### **Software Libraries**

Software libraries are available for controlling the RoboClaw with external harware such as a micrcontroller. Libraries are available for Arduino (C++) , Raspberri Pi (Python) and C#. The libraries and example code can be downloaded at: <https://www.basicmicro.com> and <https://resources.basicmicro.com/>

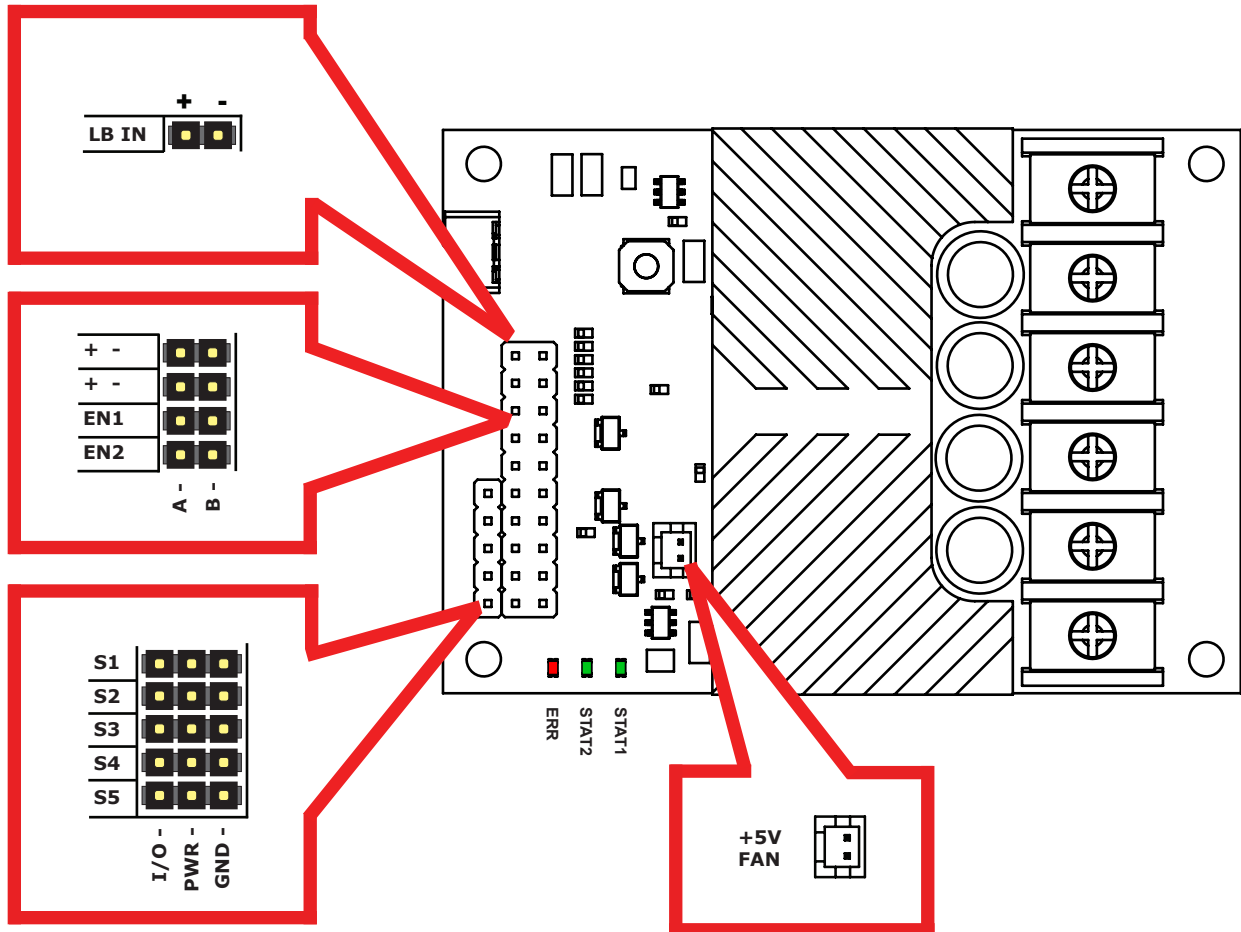
Hardware Overview:



ID	Function	DESCRIPTION
A	Status LEDs	Provides RoboClaw status information.
B	USB Port	Communicate with RoboClaw via USB.
C	Control Inputs	S1,S2,S3,S4 and S5 control inputs.
D	Encoder Inputs	Dual encoder input and power pins.
E	Logic Battery	Logic battery jumper setup and logic battery power input.
F	Motor Channel 1	Motor driver output screw terminals for channel 1.
G	Main Battery	Main battery screw terminal input.
H	Motor Channel 2	Motor driver output screw terminals for channel 2.
I	Fan Control	Fan Controller output. 5VDC. Fan On at 45°C. Fan Off at 35°C

**Control Interface**

The RoboClaw uses standard male pin headers with 0.100" (2.54mm) spacing. The pin headers are ideal for use with standard servo cables and other popular interface connectors. The table below list the pins and their respective functions. All pins are 5V tolerant and output 3.3V for compatibility with processor such as Raspberry Pi and Arduino. R/C pulse input, Analog and TTL can be generated from any microcontroller such as a Arduino or Raspberry Pi. The R/C Pulse input pins can also be driven by any standard R/C radio receiver. There are several user configurable options available. To configure RoboClaw, install Motion Studio and connect it to an available USB port.



NAME	UART TTL	ANALOG	R/C PULSE	FLIP SWITCH	E-STOP	HOME	LIMIT	V-CLAMP	Encoder
S1	RX	Motor 1	Motor 1						
S2	TX	Motor 2	Motor 2						
S3				X	X			X	
S4					X	Motor 1	Motor 1	X	
S5					X	Motor 2	Motor 2	X	
EN1									Motor 1
EN2									Motor 2
FAN									

### Logic Battery (LB IN)

The logic circuit of RoboClaw can be powered from a secondary battery wired to LB IN. A logic battery will prevent brownouts when the main battery is low or under heavy load. The positive (+) terminal is located at the board edge and ground (-) is the inside pin closest to the heatsink.

### Encoder Power (+ / -)

The pins labeled + and - are the source power pins for encoders. The positive (+) is located at the board edge and supplies +5VDC. The ground (-) pin is near the heatsink. On RoboClaws with screw terminals, power for the encoders can be supplied by the 5VDC and GND on the main screw terminal.

### Encoder Inputs (1A / 1B / 2A / 2B)

The encoders inputs are labeled EN1 and EN2. EN1 is for encoder 1 and EN2 is for encoder 2 which also correspond to motor channel 1 and motor channel 2. Quadrature encoder inputs are typically labeled 1A, 1B, 2A and 2B. Channel A of both EN1 and EN2 are located at the board edge on the pin header. Channel B pins are located near the heatsink on the pin header. Quadrature encoders are directional. When connecting encoders make sure the leading channel for the direction of rotation is connected to A. If one encoder is backwards to the other you will have one internal counter counting up and the other counting down. Use Motion Studio to determine the encoders direction relative to the motors rotation. Encoder channels A and B can be swapped in software using Motion Studio to avoid re-wiring the encoder or motor.

### Control Inputs (S1 / S2 / S3 / S4 / S5)

S1, S2, S3, S4 and S5 are configured for standard servo style headers I/O (except on ST models), +5V and GND. S1 and S2 are the control inputs for serial, analog and RC modes. S3 can be used as a flip switch input, when in RC or Analog modes. In serial mode S3, S4 and S5 can be used as emergency stops inputs or as voltage clamping control outputs. When configured as E-Stop inputs, they are active when pulled low. All I/O have internal pull-ups to prevent accidental triggers when left floating. S4 and S5 can be configured as home switch and limit switch inputs. The pins closest to the board edge are the I/Os, center pin is the +5V and the inside pins are ground. Some RC receivers have their own supply and will conflict with the RoboClaw's 5v logic supply. It may be necessary to remove the +5V pin from the RC receivers cable in those situations.

### Cooling Fan Control

The fan control will automatically turn on and off a fan based on RoboClaws temperature. The fan will turn on when the board temperature reaches 45°C and will automatically turn off when the board temperature falls below 35°C. The fan control circuit can power a 5VDC fan at up to 230mA. A wide range of fans can be used. The CFM rating of the fan will determine how effective the fan is at cooling. A tested fan is available from DigiKey under part number: 259-1577-ND. However any fan can be used provided it meets the electrical specifications outlined above.

### Main Battery Screw Terminals

The main power input can be from 6VDC to 34VDC on a standard RoboClaw. The connections are marked + and - on the main screw terminal. The plus (+) symbol marks the positive terminal and the negative (-) marks the negative terminal. The main battery wires should be as short as possible. Do not disconnect the negative terminal if other ground sources are connected, such as USB cable.



***Do not reverse main battery wires or damage will occur.***



***Do not disconnect the negative wire if other ground sources are connected, such as USB cable. Damage will occur.***

### Disconnect

The main battery should include a quick disconnect of the positive terminal, in case of a run away situation and power needs to be cut. The switch must be rated to handle the maximum current and voltage from the battery. Total current will vary depending on the type of motors used. A common solution would be an inexpensive contactor which can be sourced from sites like Ebay. A power diode rated for approximately 2 to 10 Amps should be placed across the switch/contactator to provide a return to the battery when power is disconnected. The diode will provide the regenerative power a place to go even if the switch is open.

### Motor Screw Terminals

The motor screw terminals are marked with M1A / M1B for channel 1 and M2A / M2B for channel 2. For a typical differential drive robot, the wiring of one motor, should be reversed from the other. The motor and battery wires should be as short as possible.

### Wire Length

Wire diameter and length are important to consider when designing a robot or motor control system. As wire length increases in length, its inductance is increased. This can cause potentially harmful voltage spikes. The wires from the controller to battery and motors should be short as possible with the largest gauge available.

### Power Supplies

RoboClaw is a regenerative motor controller. When a motor is braking, slowing or coasting it will generate energy. When a battery is used the surplus energy is dump back into the battery. When a power supply is used the regenerative energy has no where to go. If the excess energy is not controlled when using a power supply it will result in damage to the motor controller or power supply or potentially both. There are a few ways to manage the regenerative energy. The two most common ways are a battery in parallel with the power supply or clamping the excessive energy and converting it to heat through a high watt resistor. See Vclamp at Basicmicro.com



***Regenerative energy can result in damage to the motor controller if left unmanaged.***

### ESD Sensitivity

RoboClaw utilizes a high performance processor that can be susceptible to electrostatic discharge. I/O and control signals are protected. However, handling of the PCB, static can be introduced into areas that will bypass this protection. Ensure a discharge before handling of RoboClaw.



***Static Sensitive Device. Handle with care!***

## Control Modes

RoboClaw has 4 main functional control modes explained below. Each mode has several configuration options. The modes can be configured using Motion Studio or the built-in buttons. Refer to the RoboClaw User Manual for installation and setup instructions.

## RC

Using RC mode RoboClaw can be controlled from any hobby RC radio system. RC input mode also allows low powered microcontrollers such as a Basic Stamp to control RoboClaw. Servo pulse inputs are used to control the direction and speed. Very similar to how a regular servo is controlled. Encoders are supported in RC mode, refer to the RoboClaw user manual for setup instructions.

## Analog

Analog mode uses an analog signal from 0V to 2V to control the speed and direction of each motor. RoboClaw can be controlled using a potentiometer or filtered PWM from a microcontroller. Analog mode is ideal for interfacing RoboClaw with joystick positioning systems or other non microcontroller interfacing hardware. Encoders are supported in Analog mode, refer to the RoboClaw user manual for setup instructions.

## Simple Serial

In simple serial mode RoboClaw expects TTL level RS-232 serial data to control direction and speed of each motor. Simple serial is typically used to control RoboClaw from a microcontroller or PC. If using a PC, a MAX232 or an equivalent level converter circuit must be used since RoboClaw only works with TTL level inputs. Simple serial includes a slave select mode which allows multiple RoboClaws to be controlled from a signal RS-232 port (PC or microcontroller). Simple serial is a one way format, RoboClaw can only receive data. Encoders are not supported in Simple Serial mode.

## Packet Serial

In packet serial mode RoboClaw expects TTL level RS-232 serial data to control direction and speed of each motor. Packet serial is typically used to control RoboClaw from a microcontroller or PC. If using a PC a MAX232 or an equivalent level converter circuit must be used since RoboClaw only works with TTL level input. In packet serial mode each RoboClaw is assigned a unique address. There are 8 addresses available. This means up to 8 RoboClaws can be on the same serial port. Encoders are supported in Packet Serial mode, refer to the RoboClaw user manual for setup instructions.

## USB Control

USB can be used in any mode. When RoboClaw is in packet serial mode and another device, such as an Arduino, is connected commands from the USB and Arduino will be executed and can potentially override one another. However if RoboClaw is not in packet serial mode, motor movement commands will be overridden by Analog or RC pulse input. USB packet serial commands can then only be used to read status information and set configuration settings.



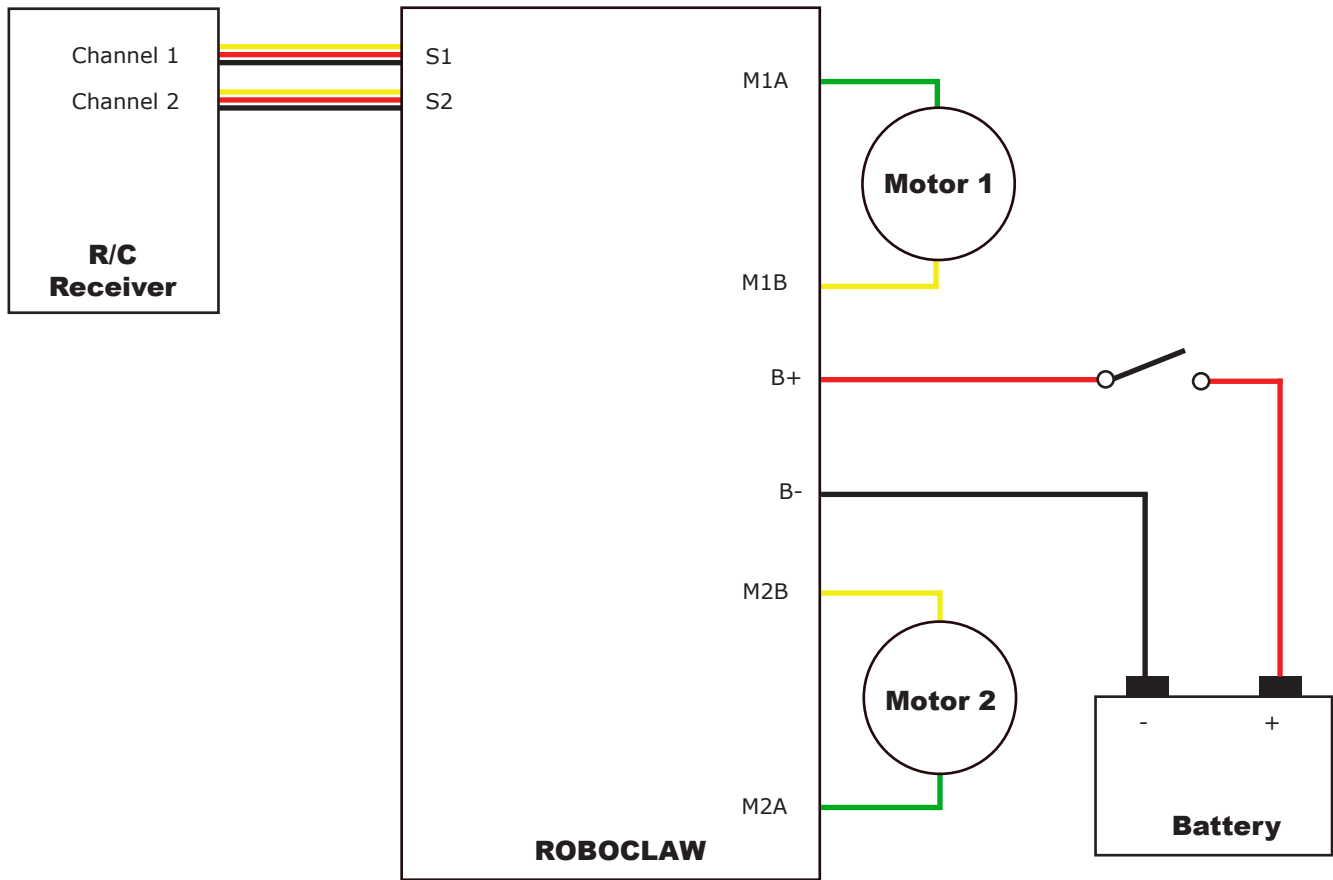
**Wiring Basics**

There are several wiring configurations for RoboClaw. Each configuration will have unique wiring requirements to ensure safe and reliable operation. The diagram below illustrates a very basic wiring configuration with a maximum voltage of 12VDC and with small motor system, where safety concerns are minimal. This is the most basic wiring configuration possible. All uses of RoboClaw should include some kind of main battery cut off switch, even when safety concerns are minimal. Never underestimate a system with movement when an uncontrolled situation arises.

In addition, RoboClaw is a regenerative motor controller. If the motors are moved when the system is off, it could cause potential erratic behavior due to the regenerative voltages powering the system. The regenerative voltages can cause problems if a power supply is used for main power. A voltage clamping circuit is recommended to dump the excessive voltages. See the RoboClaw user manual or Application Notes for voltage clamping setup and wiring diagrams.

**R/C Mode**

The below wiring diagram is very basic and for use with R/C mode. R/C mode can be used when pairing RoboClaw with a standard R/C receiver. R/C mode can also be used with a microcontroller and using servo pulses to control RoboClaw. The RoboClaw supplies power to the R/C system. If the R/C receiver used, has its own power the 5V pin on the 3 pin connector must be removed, otherwise it will interfere with RoboClaw's BEC.



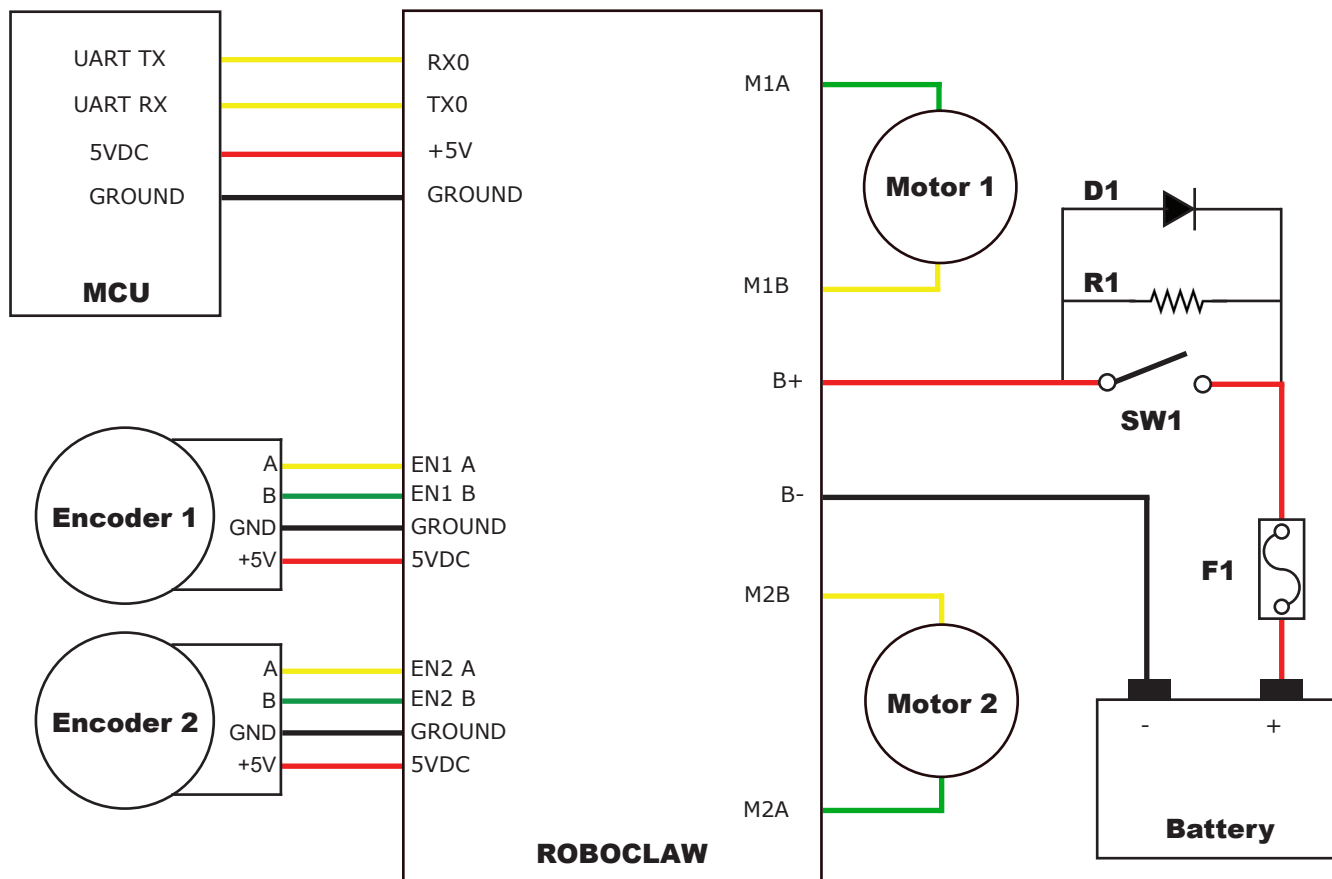
**Wiring Safety**

In system with movement, safety is a concern. This concern is amplified when dealing with higher voltages. The wiring diagram below illustrates a properly wired system. An external main power cut off is required (SW1). The external cut off can consist of a high amperage mechanical switch or a contactor.

When the RoboClaw is switched off or a fuse is blown, a high current diode (D1) is required to create a return path to the battery for potential regenerative voltages. In addition, a pre-charge resistor (R1) is required to reduce the high inrush currents to charge the on board capacitors. A pre-charge resistor (R1) should be around 1K, 1/2Watt for a 60VDC motor controller which will give a pre-charge time of about 15 seconds. A lower resistances can be used with lower voltages to decrease the pre-charge time.

**Closed Loop Mode**

A wide range of sensors are supported for closed loop operation. RoboClaw supports dual quadrature encoders (up to 9.8 million PPS), absolute encoders, potentiometers and hall effect sensors. The wiring diagram below is an example of closed loop mode using quadrature encoders. Quadrature encoders are directional. RoboClaw’s internal counters will increment for clockwise rotation (CW) and decrement for counter clockwise rotation (CCW). When wiring encoders A and B channels it is important they are wired to match the direction of the motor. If the encoder is wired in reverse it can cause a run away condition. All motor and encoder combinations will need to be tuned (see the RoboClaw user manual).

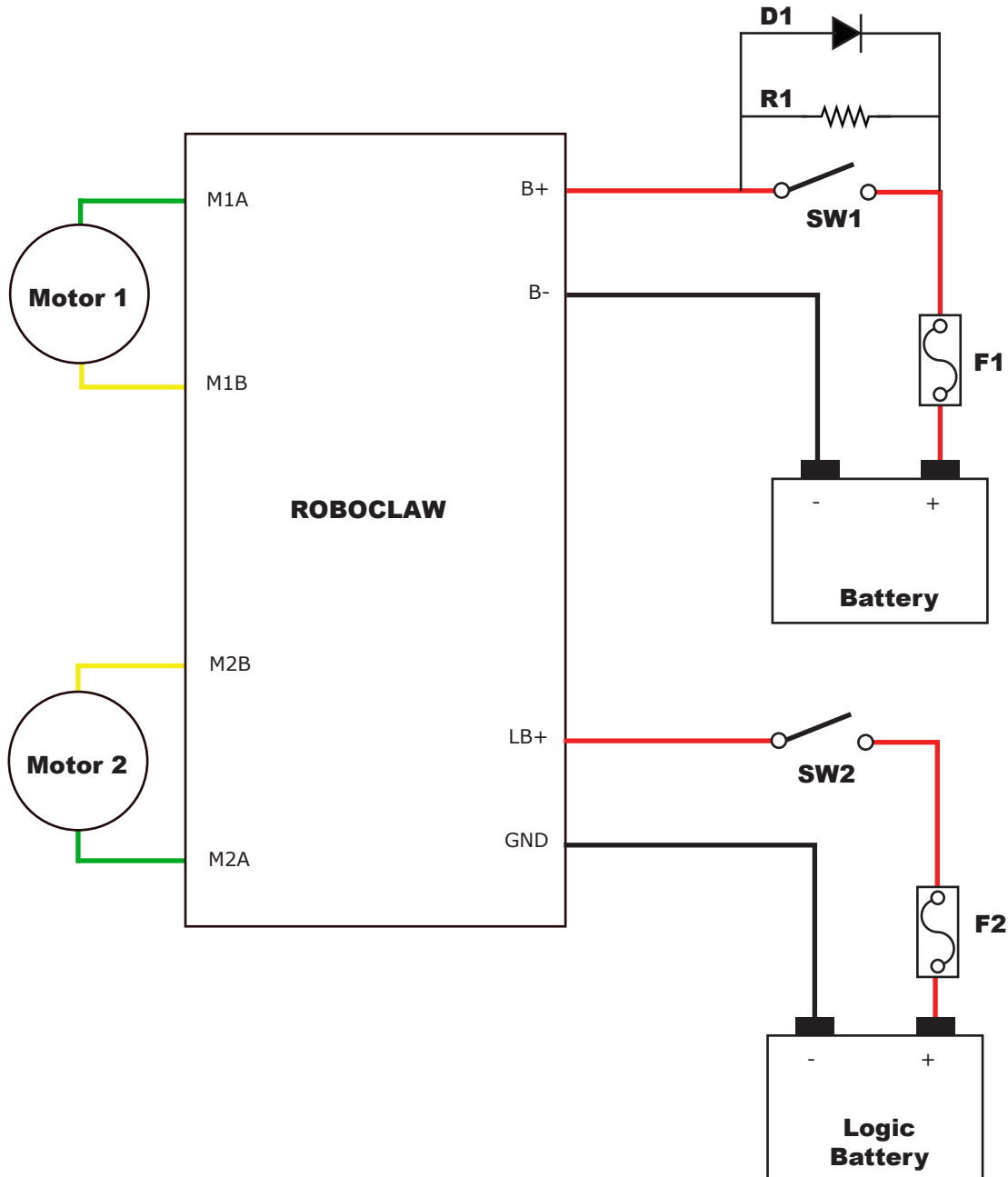


**Logic Battery**

An optional logic battery is supported. Under heavy loads the main power can suffer voltage drops, causing potential logic brown outs which may result in uncontrolled behavior. A separate power source for the motor controllers logic circuit, will remedy potential problems from main power voltage drops. The logic battery maximum input voltage is 34VDC with a minimum input voltage of 6VDC. The 5V regulated user output is supplied by the secondary logic battery if supplied. The mAh of the logic battery should be determined based on the load of attached devices powered by the regulated 5V user output.

**Logic Battery Jumper**

The configuration below utilizes a logic battery. Older models of RoboClaw have a logic battery jumper. On models where the LB-MB header is present the jumper must be removed when using a separate logic battery. If the header for LB-MB is not present, then the RoboClaw will automatically set the logic battery power source.



## RoboClaw Setup

The RoboClaw features several control modes such as packet serial, simple serial, RC and analog. Documentation for wiring and software configuration can be found in the User Manual as well as RoboClaw Application Notes.

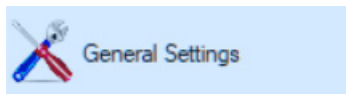
The User Manual can be found here: <https://www.basicmicro.com/downloads>  
 Application Notes can be found on our website: <https://resources.basicmicro.com/>

Motion Studio is required to setup RoboClaw. Motion Studio is a free application that can be downloaded from [Basicmicro.com](http://Basicmicro.com).

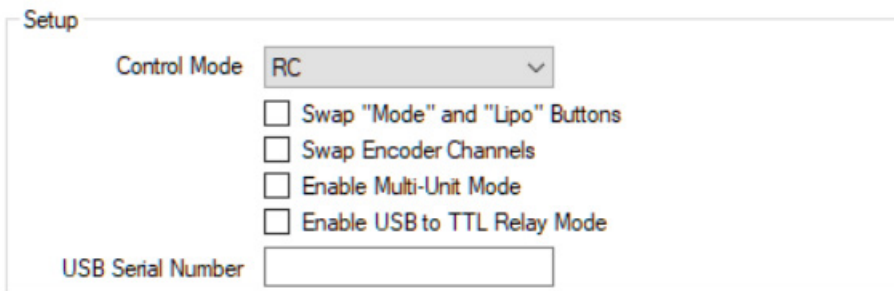
## RC Control Setup

This quick guide will outline setting up a RoboClaw for RC control.

1. Download and install Motion Studio.
2. Wire the RoboClaw, power supply, motor and RC hardware as shown in the RC wiring diagram in this datasheet.
3. Ensure the RoboClaw is wired using the safety precautions illustrated in this datasheet.
2. Connect a micro USB cable between the RoboClaw and a computer with Motion Studio installed. The USB connection does not power the RoboClaw, a power supply must be connected. Open Motion Studio and on the top left under "Attached Devices" select and click the motor controller shown.
3. Next, click on General Setting on the left-hand side of the application.



4. In the panel labeled Setup, click on the Control Mode drop down and select RC.

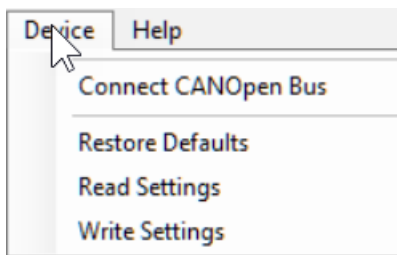


5. Next, RC options can be set in the panel labeled RC/Analog Options.



- **Mixing** - This setting is for use with a differential drive configuration in which only two motors are used for forwards/backwards movement as well as left/right steering. Tank robots are a classic example of where this setting is applicable.
- **Exponential** - The exponential setting affects the sensitivity of the controls, whether that be with a radio transmitter or microcontroller. When exponential is enabled the controls are less sensitive at the beginning of their range and more sensitive at the outside of their range.
- **MCU** - This setting allows a microcontroller to send RC-like servo pulses in place of using a typical RC receiver. RoboClaw does not need a constant pulse train in this mode. The RoboClaw will use the value of the last pulse received and execute that until a new pulse is sent. RoboClaw does not auto-calibrate the pulses sent to it in this mode.
- **RC Flip / Mode** - This setting allows for an input on S3 to flip the control direction of a motor attached to the RoboClaw. When the checkbox is selected an RC receiver output is the control signal that flips the control scheme of the motor.
- **Enable Encoder 1 in RC/Analog Mode** - This setting allows you to enable an encoder on channel one. The RoboClaw uses encoders for closed-loop control of position and velocity. This means that when a speed or position is set the RoboClaw will do the work of maintaining that value. For example, with encoders, a robot climbing a hill would be able to maintain a constant speed despite the change in load on the motors. Prior tuning of the encoders, is necessary, to use them properly.
- **Enable Encoder 2 in RC/Analog Mode** - This setting allows you to enable an encoder on channel two. The RoboClaw uses encoders for closed-loop control of position and velocity. This means that when a speed or position is set the RoboClaw will do the work of maintaining that value. For example, with encoders, a robot climbing a hill would be able to maintain a constant speed despite the change in load on the motors. Prior tuning of the encoders, is necessary, to use them properly.

5. To complete the setup click on the "Device" menu and select "Write Settings". This will update and save the settings to the attached RoboClaw.



6. Disconnect the RoboClaw from Motion Studio by clicking "Disconnect Selected Unit" in the upper left-hand side of the application. The USB connection can then be removed from the attached controller.

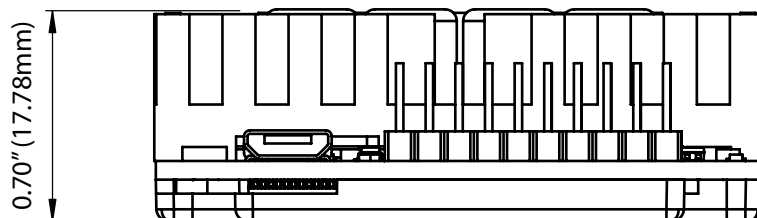
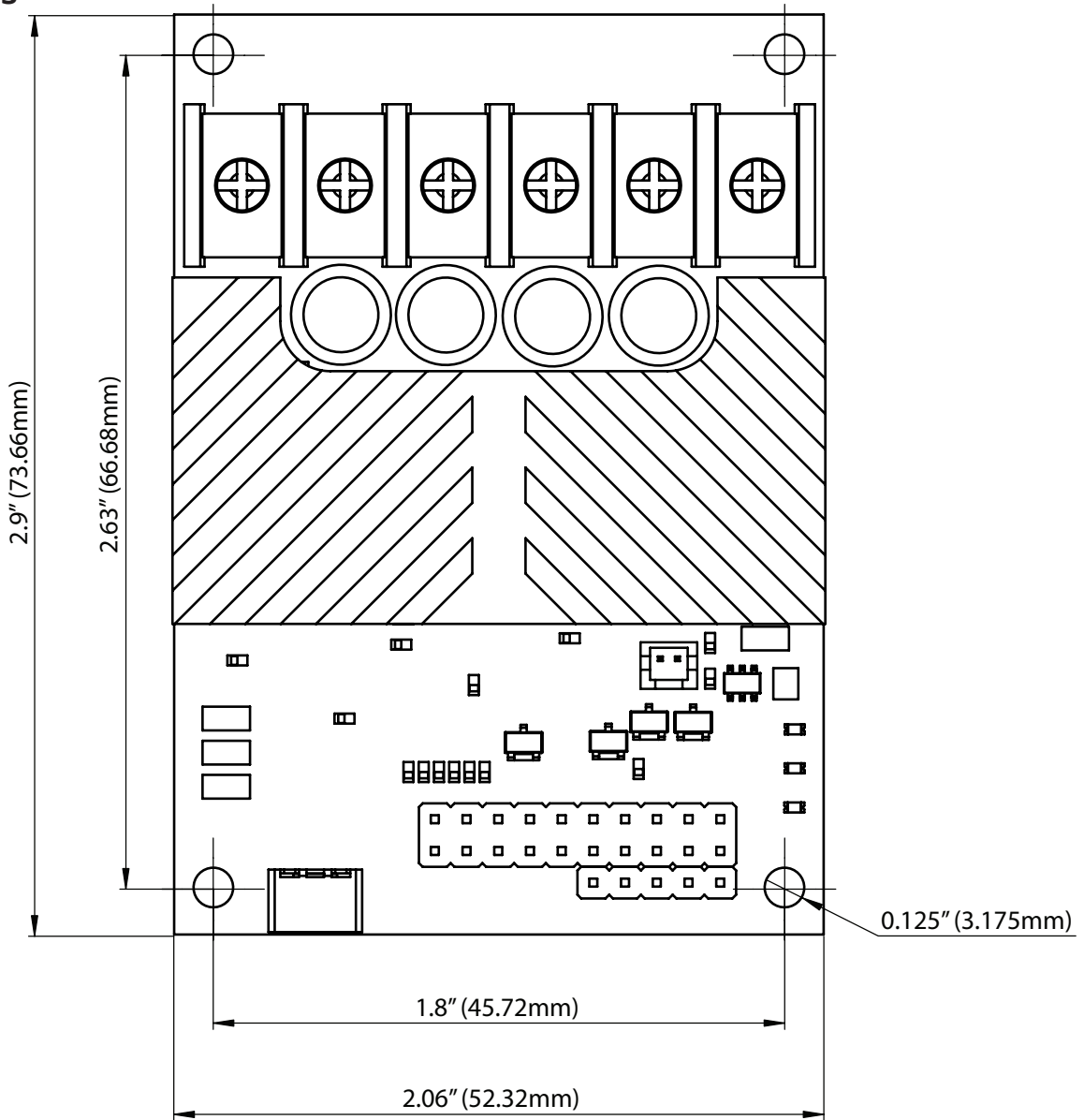
7. The robot or other system can now be operated remotely with a RC transmitter.

**Mechanical Specifications**

Characteristic	Model	Min	Typ	Max	Rating
Weight	2X30A		2.2 (62)		Oz (g)

**Dimensions**

Inches [MM]



## Electrical Specifications

Characteristic	Min	Typ	Max	Rating
Main Battery	6		34	VDC
Logic Battery	6	12	34	VDC
Maximum External Current Draw (BEC)			3	A
Motor Current Per Channel		30 <sup>(2)</sup>	60 <sup>(1,2)</sup>	A
On Resistance		4.3		mOhm
Logic Circuit Current Draw		30mA		mA
Input Impedance		100		$\Omega$
Input	0		5	VDC
Input Low	-0.3		0.8	VDC
Input High	2		5	VDC
I/O Output Voltage	0		3.3	VDC
Digital and Analog Input Voltage			5	VDC
Analog Useful Range	0		2	VDC
Analog Resolution		1		mV
Pulse Width	1		2	mS
Encoder Counters		32		Bits
Encoder Frequency			9,800,000	PPS
RS232 Baud Rate (Note 3)			460,800	Bits/s
RS232 Time Out (Note 3)	10			ms
Temperature Range	-40	40	100	$^{\circ}\text{C}$
Temperature Protection Range	85		100	$^{\circ}\text{C}$
Humidity Range			100 (4)	%

**Notes:**

1. Peak current is automatically reduced to the typical current limit as temperature approaches 85 $^{\circ}\text{C}$ .
2. Current is limited by maximum temperature. Starting at 85 $^{\circ}\text{C}$ , the current limit is reduced on a slope with a maximum temperature of 100 $^{\circ}\text{C}$ , which will reduce the current to 0 amps. Current ratings are based on ambient temperature of 25 $^{\circ}\text{C}$ .
3. RS232 format is 8Bit, No Parity and 1 Stop bit.
4. Condensing humidity will damage the motor controller.

## Warranty

Basicmicro warranties its products against defects in material and workmanship for a period of 1 year. If a defect is discovered, Basicmicro will, at our sole discretion, repair, replace, or refund the purchase price of the product in question. Contact us at [sales@basicmicro.com](mailto:sales@basicmicro.com). No returns will be accepted without the proper authorization.

## Copyrights and Trademarks

Copyright© 2015 by Basicmicro, Inc. All rights reserved. All referenced trademarks mentioned are registered trademarks of their respective holders.

## Liability Statement

By purchasing and using Basicmicro's products you acknowledge and agree to the following: Basicmicro has no liabilities (and, to Basicmicro's knowledge, there is no basis for any present or future action against the company giving rise to any liability) arising out of any injury to individuals or property as a result of ownership, possession or use of any product designed or sold by Basicmicro.

## Disclaimer

Basicmicro cannot be held responsible for any incidental or consequential damages resulting from use of products manufactured or sold by Basicmicro or its distributors. No products from Basicmicro should be used in any medical devices and/or medical situations. No product should be used in any life support situations.

## Contacts

Email: [sales@basicmicro.com](mailto:sales@basicmicro.com)

Tech support: <https://www.basicmicro.com/crm.asp?action=contactus>

Web: <http://www.basicmicro.com>

## Discussion List

A web based discussion board is maintained at <http://www.basicmicro.com>

## Technical Support

Technical support is available by opening a support ticket at <https://www.basicmicro.com/crm.asp?action=contactus>