

Tarocco Closed Loop Motor Controller



# Contents

Safety Information..... 3

Overview..... 4

Features..... 4

    SoC for Closed Loop Control..... 4

    Gate Driver..... 5

    MOSFETs in H Bridge Configuration..... 5

    Device Characteristics..... 6

Installation..... 7

    Motor and Power Supply Connectors..... 7

    Control Signal Connectors..... 7

    Serial port tuning..... 8

    Command Table..... 10

Operation..... 11

    Fault Conditions..... 11

Updating the firmware..... 13

Dimensional Drawing..... 16

## Safety Information

You should read these notes before installing this device.

The motor controller is capable of operating up to 36 V and 10 A continuously. Be careful when close to the device while in operation, it may get hot.

The user needs to mind power supply polarity, as the device does not possess reverse polarity protection for any of the inputs.

Disconnecting the motor from this device when powered is not advised, as it could damage the electronics. Avoid shorting the motor output terminals.

## Overview

This device is designed to control DC brushed motors up to 36 V and 10 A (almost ½ hp) for axis positioning in CNC machines such as routers, milling machines, laser cutters and 3D printers. It could also be used as the power stage of a mobile robot that requires odometry. Other uses are possible with firmware modification.

## Features

The board accepts the common STEP/DIR signals that are ubiquitous to low cost machine controllers, such as the RAMPs board and parallel port breakout boards for Mach3, LinuxCNC and Machinekit.

Position sensing is achieved by means of an incremental quadrature encoder. The controller is capable of stopping the machine when it detects one of several fault conditions (discussed later) to avoid ruining a job. A full duplex serial port is available for easy configuration and tuning of the servo driver.

Shrouded and keyed headers for the signal connections are installed on the board to keep the end user from accidentally shorting two pins together or plugging a connector backwards. Pluggable screw terminals for the power connections allow the board to be easily installed inside a case or in tight spaces without having the cables getting in the way of the user.

## SoC for Closed Loop Control

A Cypress Semiconductor PSoC4 chip (ARM Cortex M0 core plus internal programmable logic and analog hardware) is present to read the STEP/DIR signals from the machine controller, read the encoder input and perform the calculations required for closed loop control of the motor kinematics.

It controls two indicator LEDs for signaling the different fault modes the device is able to detect, such as motor overcurrent, driver undervoltage, missing encoder, exceeded encoder pulse rate and motor not able to reach position. It can also send an alarm to the rest of the motor drivers.

## Gate Driver

A full bridge MOSFET gate driver switches four transistors with an appropriate voltage with the appropriate commutation speed, while also protecting them from being damaged by a shoot-through condition or overheating due to insufficient gate voltage or an overcurrent event.

The IC also amplifies the voltage drop from an external 5 mOhm shunt resistor for low side current sensing.

Finally, the driver also performs current chopping (set at 12 A) to provide a second layer of overcurrent protection. The current level is set via two resistors in a voltage divider configuration.

## MOSFETs in H Bridge Configuration

The device utilizes four ~5 mOhm (at 25 °C) N channel MOSFETs with a 60 V breakdown voltage. The transistors use the board as a heatsink.

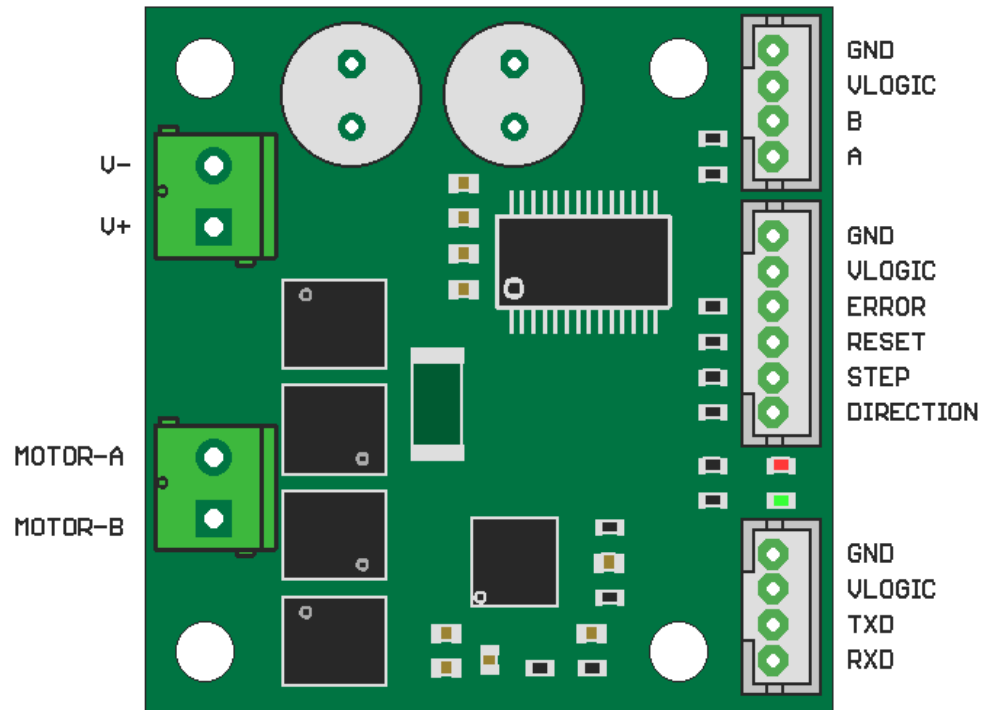


Device Characteristics

Parameter	Min	Typ	Max	Unit
Motor side voltage	6		36	V
Continuous current output <sup>1</sup>	0		10	A
Logic side voltage (VLOGIC)	1.8		5	V
Current consumption		50		mA
Step signal frequency	0		1000	kHz
Encoder frequency (4x)	0		1000	kHz
Error signal output current		4		mA
Serial port speed		19200		baud
Weight		17(0.6)		g(oz)
Operating ambient temperature <sup>2</sup>	0		40	°C

1 The use of 16 AWG cable is recommended for maximum continuous current output.  
2 40 °C is the maximum ambient air temperature for which the current output value is valid

## Installation



### Motor and Power Supply Connectors

The device comes with two pluggable screw terminals installed. The one labeled **MOTOR-A/MOTOR-B** should be connected to the pair of cables coming from the brushed motor. The other terminal, marked **V+/V-** should be connected to a power supply of adequate voltage for the motor, **ensuring the correct polarity of the connection.**

### Control Signal Connectors

One of the 4-way PH headers accepts the two quadrature inputs (**A** and **B**) from the encoder and supplies **VLOGIC** (typically 5V, but can be lower) to said encoder.

**The quadrature encoder signal must be present for the controller to work, in order to protect the machine and user when it cannot sense the position of the motor.**

The onboard PsoC can work with encoder pulse rates up to 1000 kHz (in 4x decoding mode). If the motor's angular speed in rpm and the number of counts per revolution (cpr) from the encoder are given, the pulse rate can be calculated as follows:

$$Pulse\ Rate[Hz] = \frac{rpm \cdot cpr}{60}$$

So for example, a motor with a maximum angular speed of 4500 rpm and a 1600 cpr encoder would give a maximum pulse rate of 120 kHz, well within the range of operation of the motor controller.

Some manufacturers give the counts per revolution value for a single encoder channel. In this case, the pulse rate is the value obtained with the above formula times four.

The 6-way PH header has the board power supply input, STEP/DIR inputs, RESET input and the open drain ERROR output. The GND and VLOGIC pins must be connected to a 1.8 to 5 V power supply, **minding the polarity of this connection**. When VLOGIC is supplied, the two onboard LEDs will pulse in lockstep for three seconds (it signals that the board can be programmed via its bootloader). After this period, only the VIN LED will remain lit, signaling that the controller is executing its program.

The RESET signal is active low to allow the ERROR signal to self reset the board. A general system stop can be achieved by connecting the ERROR and RESET pins from every controller together. When a fault condition is detected by one of the drives, its ERROR pin will pull low the RESET pin from every device. An emergency stop button can be easily included in this configuration.

The other 4-way PH header provides a full duplex serial port. It is used to configure and tune the device to drive the particular motor the user has with optimal performance.

**Make sure that the TXD pin on the Tarocco is connected to the RX pin on your USB serial converter and that the RXD pin on the Tarocco is connected to the TX pin on the converter.**

## Serial port tuning

The motor controller can be configured on the fly by sending the appropriate



commands to its serial port. The new values will be stored on its non volatile memory.

The serial port is configured as:

- Baud rate: 19200
- Data bits:8
- No parity bit
- One stop bit

A valid command is a packet with the following structure:

Command type – Value – End character

Command type is an ASCII character that tells the device the variable that must be updated with Value.

Value consist on ASCII characters that represent the magnitude of the variable chosen to be updated.

End character is simply an ASCII semicolon.

A valid command could be, for example, **p107;** that loads the variable corresponding to the proportional coefficient with a value of 107 units.

The serial port is also configured to report the position error and motor's current. It sends the following ASCII packet approximately 100 times a second.

Start character – Position error – Current – End character

Start character is simply a \$.

Current consist on one to five ASCII character representing the motor's current in mA.

Position error consist on one to five ASCII character representing how many encoder counts the motor its off from the desired position. It has an offset of 500 units (a value of 500 means no error).

End character is, again, a semicolon.



## Command Table

Command	Description	Default
p	Proportional term coefficient (0 to 65535)	2500
i	Integral term coefficient (0 to 65535)	0
d	Derivative term coefficient (0 to 65535)	0
c	Current limit value in mA/40 (0 to 255)	255 (10.2 A)
r	Invert direction of rotation flag (boolean) (0 to 1)	0
f	Maximum allowable following error in encoder counts (0 to 32000) <sup>1</sup>	255
u	Transmit diagnostic data flag (boolean) (0 to 1)	1
e	Clear all faults (boolean) (0 to 1)	0
+	Move the motor the given number of counts (0 to 255)	n/a
-	Move the motor the given number of counts in the opposite direction (0 to 255)	n/a

<sup>1</sup> It is possible to use a higher value, but it is not recommended, as it will allow the error counter to overflow before signaling a fault.

## Operation

The first step should be adjusting the motor controller parameters for a satisfactory operation. If you need information on how to tune a PID controller check the Resources section.

After the device has been tuned, it will attempt to make the motor spin to the position given by the state of the DIRECTION pin and number of STEP pulses it receives.

## Fault Conditions

The controller can detect errors in three different subsystems. These subsystems are the encoder, MOSFET gate driver and current measurement.

The encoder fault condition is triggered when the position error exceeds the following error value. This could mean:

Failure mode	Fix
Motor torque is exceeded, and cannot spin to the commanded position	Check the machine mechanism for defects Verify motor is not at end of travel
Motor spins in the wrong direction	Set the invert direction flag
Encoder signal is lost	Verify that the encoder's signal is reaching the board

This condition stops the motor and activates the ERROR output. The condition is signaled by turning on the ERR and VIN LEDs. The board will need a reset (remove the VLOGIC supply) to clear the fault.

A gate driver fault could mean:

Failure mode	Fix
Not enough voltage to turn on the MOSFETs	Check power supply connections Make sure the power supply is on or is delivering enough voltage
Board is too hot	Provide better cooling to the controller
Motor power supply takes too much time to turn on.	The power supply has one second to turn on after VLOGIC is supplied Turn the PSU on earlier.

This condition stops the motor and activates the ERROR output. The condition is signaled by turning on the ERR LED. The board will need a reset (remove the VLOGIC supply) to clear the fault.

## Updating the firmware

Every Tarocco board comes with a bootloader already installed. You can reprogram it using just a USB to serial converter. You'll need to download and install PsoC creator for Windows. You also need to download cncboot and the latest firmware version from the Tarocco github repository.

<http://www.cypress.com/products/psoc-creator-integrated-design-environment-ide>

<https://github.com/ottoragam/Tarocco/tree/master/Bootloader/cncboot.cydsn>

<https://github.com/ottoragam/Tarocco/tree/master/Firmware/Tarocco.cydsn>

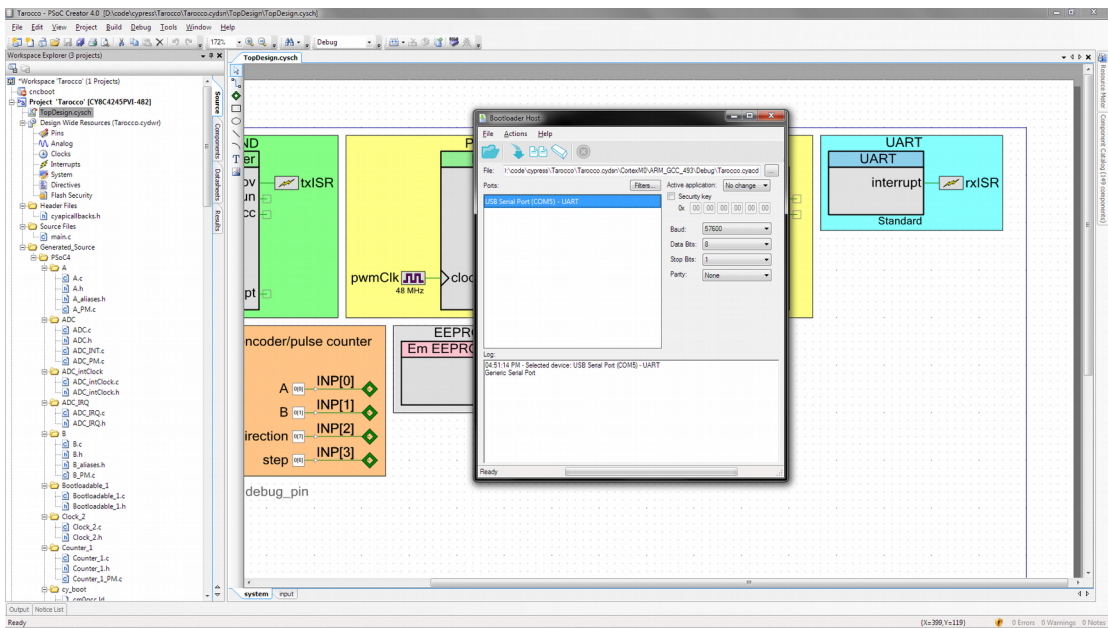
The next step is to load the Tarocco project to in Psoc Creator.





Open the Bootloader Host application in the tools menu, and adjust the parameters as seen in the image (select the COM port for your USB to serial converter and verify that the path for the firmware file you want to load is correct). The bootloader operates at 57600-8-N-1. Reboot the controller and start the programming process when the green and red LEDs start to blink in lockstep. Keep in mind that you have three seconds to press the program button, or else the bootloader will exit and the board will start to run its firmware.

You can put the board in bootloader mode too if you keep the RESET pin low before powering the board. After programming the board, make sure the RESET pin is no longer kept low.



You should see the 'Programming Successful' log event when the process is done.

If you're getting the message "the flash row is not valid for the selected array" try using the alternative cncboot project present in the repo:

[https://github.com/ottoragam/Tarocco/tree/master/Bootloader/cncboot\\_no\\_optimizations/cncboot.cydsn](https://github.com/ottoragam/Tarocco/tree/master/Bootloader/cncboot_no_optimizations/cncboot.cydsn)

Remember to build the project each time you make changes to it.

# Dimensional Drawing

