

NAME

burleigh – Burleigh 6200ULN with 671 interface

DESCRIPTION

The Burleigh 6200ULN Inchworm Controller operates the Burleigh Inchworm motors in closed-loop mode. `spec` supports this controller using the Burleigh Model 671 ISA PC card interface. Currently, only one 671 card at a time is supported, but support for multiple cards will be added when required by a `spec` user. The 671 card supports three 6200ULN controllers with each controller supporting up to three motors.

CONFIGURATION

The motor controller type is `BURL671` on the motor screen of the configuration editor. The steps-per-degree parameter is only used to determine the resolution at which you can specify moves. A value of 100 is appropriate for a 0.02 um encoder resolution.

To avoid frequent messages regarding position discrepancies, choose a reasonable number of steps for the motor “slop” parameter on the optional motor-parameter screen of the configuration editor.

CALIBRATE

To achieve well-defined speeds, it is necessary to calibrate each motor’s movement. The `motor_par()` option “calibrate”, described below, does that. A motor is moved at a series of speeds, and the distance traveled and time taken is measured. The measured values are fit to a simple power law described by two constants, with separate parameters calculated in forward in reverse directions for high and low speeds. Thus, eight parameters are found for each motor.

Note, the calibration operation takes a couple of minutes. `spec` turns off stall detection, if on, during calibration, and restores it afterwards.

The calibration parameters for nine motors can be saved in ASCII format to a file according to Burleigh conventions. `spec` will automatically load calibrations from a file if it exists and can be found using the following rules. If the environment variable `BURLEIGH_CALIB` set, its value is taken as the pathname of the calibration file. If it is not set, `spec` looks for a file named `671calib.txt` in the current directory. The parameters are loaded from the calibration file, if it exists, when the controller is initialized at program start up and on hardware reconfiguration.

While running `spec`, the `motor_par()` options “calib_file”, “load_calib” and “save_calib”, also described below, can be used to change the name of the file and to load or save calibration parameters.

STALL

The controller has the ability to detect a stalled motor. The `motor_par()` option “stall” is used to turn stall-detection on or off. When stall detection is turned on, the controller makes a configurable number of checks that the encoder has sent a configurable number of pulses over a configurable interval. If stall is detected, the controller stops the motor and returns a status indication. The above three parameters can be entered directly using the `motor_par()` options “stall_tests”, “stall_interval” and “stall_tolerance” described below. Since the stall detection depends on the motor speed, the above parameters need to be changed to match the speed.

`spec` provides an alternative method for setting the stall parameters that automatically adjusts to the motor speed. The `motor_par()` options “stall_time” and “stall_percent”, described below, are used. The “stall_time” parameter is the number of seconds in total to spend doing the stall tests, while “stall_percent” is the percentage value of the current speed below which the motor is considered stalled.

Setting any of the first three parameters switches to the mode where the parameters are set directly. Setting either of the last two parameters switches to the alternative mode.

SPECIAL COMMANDS

The following special commands are available through the `motor_par()` function.

`motor_par(motor, "calib_file" [, file])` – Sets or returns the pathname of the calibration file. The same calibration file is used for all motors. The file name is not reset during hardware reconfiguration. The *motor* parameter is any motor associated with the 671 controller.

`motor_par(motor, "load_calib" [, file])` – Loads calibration parameters for all motors from the current calibration file or from the *file* specified as the optional argument. The *motor* parameter is any motor associated with the 671 controller.

`motor_par(motor, "save_calib" [, file])` – Writes calibration parameters for all motors to the current calibration file or to the *file* specified as the optional argument. The *motor* parameter is any motor associated with the 671 controller.

`motor_par(motor, "calibrate")` – Runs the calibration test for the specified motor. Use the "save_calib" option to save the calibration parameters to a file.

`motor_par(motor, "maint" [, how])` – Turns position maintenance on or off for *motor* if the optional argument *how* is nonzero or zero. Without the arguments, returns nonzero or zero reflecting the current state of maintenance mode.

`motor_par(motor, "dwell" [, how])` – If the optional argument is present enables the dwell test for *motor*. Otherwise returns a status indicating whether the dwell-test mode is active. Consult the Burleigh documentation for details.

`motor_par(motor, "stall" [, how])` – If the optional argument is present, turns stall detection on or off for *motor* depending on whether *how* is nonzero or zero. Otherwise, returns the value matching the current state of stall detection.

`motor_par(motor, "stall_tests" [, value])` – Sets or returns the numbers of consecutive tries for the stall detection to consider *motor* stalled. The default value is 5.

`motor_par(motor, "stall_interval" [, value])` – Sets or returns the time interval for each stall test in seconds for *motor*. The default value is 0.05 seconds.

`motor_par(motor, "stall_tolerance" [, value])` – Sets or returns the number of encoder counts expected during each stall test for *motor*. The default value is 30.

`motor_par(motor, "stall_time" [, value])` – Sets or returns the total stall time for the alternative stall parameter calculation for *motor*. The default value is 0.2 seconds.

`motor_par(motor, "stall_percent" [, value])` – Sets or returns the percent of current speed below which *motor* is considered stalled. The default value is 50 for fifty percent.

`motor_par(motor, "mvsubstep", substeps)` – Moves *motor* the selected number of *substeps* and returns when the movement is complete. The motor position is updated at the end of the move.

`motor_par(motor, "display" [, how])` – If *how* is present, selects the units for the front panel display for *motor* as follows:

- 1 inches
- 2 millimeters
- 4 microns

Without the optional argument, the value corresponding to the current display setting is returned.