

**NAME**

Attocube – Attocube ANC350 motor controller

**DESCRIPTION**

spec supports the ANC350 piezo controller over an Ethernet connection.

The controller is selected on the hardware configuration editor **Devices** screen along the lines of:

```
MOTORS    DEVICE  ADDR <>MODE NUM          <>TYPE
        YES atto_host          8  Attocube ANC350 (Socket)
```

Here, `atto_host` is a resolvable host name or IP address. The default port number, 2101, is built into the code. If the controller is set up to use another port, enter the **DEVICE** as `atto_host:port`. On the **Motors** screen, choose **ANC350** as the controller type.

The parameter descriptions below are brief. Consult Attocube and the ANC350 documentation for complete details on operation of the controller. The current support for the ANC350 (as of spec release 5.08.03-4) does not support all the controller capabilities. Please contact CSS if support for additional functionality is needed.

**PARAMETERS**

The ANC350 controller has many configuration parameters for the individual axis channels. All the parameters names used in the “actor profile” (*.aps*) files supplied by Attocube for use with their software are recognized by spec. Note, spec can also read the *.aps* files directly with the “load” option described below. Additional parameters particular to the ANC350 are also recognized by spec.

Values for the parameters can also be included in spec’s hardware configuration file and maintained with the hardware configuration editor on the non-standard optional parameter screen. (Type **p** from the **Motor** screen.) Values for each of the parameters may also be set or read via the `motor_par()` function.

The device units (mm, deg, etc.) for many of the parameters are determined by the “sensorunit” parameter value as described below.

Note, setting either of the parameters “humpenable” and “stickyhump” to zero will interfere with spec’s ability to detect limits. Setting the parameter “relays” to zero will disable the channel’s output. spec will set each of these three parameters to one on hardware configuration.

The first set of `motor_par()` examples below lists parameters with the same names used in the Attocube *.aps* files:

`motor_par(mne, "actordir" [, arg])` – Actuator direction, forward is 0, backward is 1.

`motor_par(mne, "actorgain" [, arg])` – Loop gain in device units per Volt. The value is scaled by a factor of  $1e6$ .

`motor_par(mne, "actoroffset" [, arg])` – Loop offset in millivolts.

`motor_par(mne, "adaptsetpctrl" [, arg])` – Enables (1) or disables (0) approach speed function.

`motor_par(mne, "amplctrl" [, arg])` – Indicates the piezo drive control mode – 0 is speed, 1 is amplitude and 2 is step width.

In speed-control mode, the controller maintains a constant travel speed in a closed-loop mode. The speed is set by the “frequency” and “amplitude” parameters, and can be retrieved using the “speed” parameter.

In amplitude-control mode, the speed and step width are varied in an open-loop mode to maintain a constant amplitude level as set by the "amplitude" parameter.

In step-width mode, the amplitude is controlled via a closed-loop to maintain a constant step width. The speed may vary depending on the value set for the "frequency" parameter. The step width is set with the "amplitude" parameter and can be retrieved using the "step\_width" parameter.

`motor_par(mne, "amplctrlavg" [, arg])` – Average factor for speed feedback.

`motor_par(mne, "amplctrlsensitivity" [, arg])` – Sensitivity for speed feedback.  
The value is scaled by a factor of 1000.

`motor_par(mne, "humpenable" [, arg])` – Enables (1) or disables (0) hump detection.  
`spec` enables hump detection on hardware configuration in order to enable limit detection.

`motor_par(mne, "maxampl" [, arg])` – Maximum amplitude in millivolts.

`motor_par(mne, "maxfrequ" [, arg])` – Maximum frequency in Hz.

`motor_par(mne, "period" [, arg])` – Number of periods per device unit for optical sensors.

`motor_par(mne, "positionmin" [, arg])` – The minimum position of the sensor in device units. The value is scaled by a factor of 1000.

`motor_par(mne, "positionmax" [, arg])` – The maximum position of the sensor in device units. The value is scaled by a factor of 1000.

`motor_par(mne, "poslooprange" [, arg])` – The position loop range in device units.  
The value is scaled by a factor of 1000.

`motor_par(mne, "refoffset" [, arg])` – The reference offset in device units. The value is scaled by a factor of 1000.

`motor_par(mne, "rotary" [, arg])` – The value is 0 for linear and 1 for rotary stages.  
This parameter can be set as one of the standard optional parameters on the second optional motor parameter screen of the configuration editor.

`motor_par(mne, "sensoravg" [, arg])` – The sensor-average factor.

`motor_par(mne, "sensordir" [, arg])` – Sensor direction, forward is 0, backward is 1.

`motor_par(mne, "sensorres" [, arg])` – Value for internal signal averaging.

`motor_par(mne, "sensorunit" [, arg])` – The code for the sensor device units, as given in the table below:

- 0 millimeter
- 1 micrometer
- 2 nanometer
- 3 picometer
- 20 degree
- 21 millidegree
- 22 microdegree

`motor_par(mne, "singlecircle" [, arg])` – Enables (1) or disables (0) the shortest way algorithm.

`motor_par(mne, "slowspeed" [, arg])` – Target approach speed in device units per second. The value is scaled by a factor of 1e6.

`motor_par(mne, "speedgain" [, arg])` – Approach speed gain in Hz. The value is scaled by a factor of 1000.

`motor_par(mne, "targetctrlavg" [, arg])` – Averaging factor for position feedback.

`motor_par(mne, "targetctrlsensitivity" [, arg])` – Sensitivity for position feedback. The value is scaled by a factor of 1000.

`motor_par(mne, "targetrange" [, arg])` – Positioning accuracy in device units. The value is scaled by a factor of 1000.

`motor_par(mne, "targettime" [, arg])` – Minimum time (in milliseconds) at position to consider approach to target a success.

`motor_par(mne, "transfergain" [, arg])` – Gain for resistive transfer function in device units per Volt. The value is scaled by a factor of 1000.

`motor_par(mne, "transferoffs" [, arg])` – This obsolete parameter appears in the *.aps* files, but is no longer supported by the controller firmware. *spec* recognizes but ignores the parameter.

`motor_par(mne, "transfertype" [, arg])` – Type of sensor, 0 is optical, 1 is resistive.

The following parameters do not appear in the *.aps* files, but are also available:

`motor_par(mne, "amplitude" [, arg])` – The amplitude of the drive voltage for the piezo drive in minivolts. Changing this value changes the step size. Allowed values are from 0 to 70 Volts.

`motor_par(mne, "dclevel" [, arg])` – The DC level represents the applied DC voltage to the piezo.

`motor_par(mne, "frequency" [, arg])` – The frequency is used to adjust the travel speed in speed and step-width drive-control modes. This parameter is used as and tracks the standard steady-state rate parameter from the *config* file.

`motor_par(mne, "refvolt" [, arg])` – Global controller reference voltage in millivolts. The value is applied to the controller associated with motor *mne*.

`motor_par(mne, "relays" [, arg])` – Enables (1) or disables (0) the output relays. The relays must be enabled for the actuators to function. *spec* enables the relays during hardware configuration.

`motor_par(mne, "stickyhump" [, arg])` – Enables (1) or disables (0) the “sticky” hump bit. *spec* enables this bit during hardware configuration in order for limit detection to work properly.

The following parameters are read only:

`motor_par(mne, "capacitance")` – Returns the result of a capacitance measurement of the connected positioner.

`motor_par(mne, "maxpos")` – Maximum position for actuator in device units.

`motor_par(mne, "minpos")` – Minimum position for actuator in device units.

`motor_par(mne, "position")` – Current position of actuator in device units.

`motor_par(mne, "refpos")` – Reference position of actuator in device units.

`motor_par(mne, "speed")` – The speed setting in device units per second. The value is scaled by a factor of 1000.

`motor_par(mne, "step_width")` – The width per step in device units. The value is scaled by a factor of 1000.

Finally, `spec` supports the following additional commands for the ANC350 controller:

`motor_par(mne, "load", filename)` – Reads the named file, which should be in the format of the Attocube `.aps` “actor profile” files, and sets the parameters to the values given in the file. The format of the files is the parameter name followed by a space and then a value, with one entry per line.

`motor_par(mne, "dump")` – Prints a formatted display of all the motor parameters for the given motor. The display includes the parameter description, the scaled value with units, the unscaled value in parenthesis and the parameter name in quotation marks.

`motor_par(mne, "step", value)` – Single steps the given axis the number of steps given by `value`. Positive values step forward using the `ID_ANC_SGL_FWD` command. Negative values step backward using the `ID_ANC_SGL_BKWD` command. The change in position detected by the sensor may be greater than the number of steps given as an argument, depending on values of other parameters, such as the amplitude. If the change in position steps is greater than the motor read-back “slop” parameter and the hardware “read\_mode” parameter requires queries on position discrepancies greater than the read-back slop parameter, `spec` will issue the standard position discrepancy message when motor positions are next requested.

`motor_par(mne, "measure_capacitance")` – `spec` initiates a capacitance measurement, pauses 0.5 seconds for the measurement to complete, then returns the measured value. The “capacitance” option, above, returns the last measured value, but doesn’t initiate a measurement.