

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Newport / Klinger MM2000* at our website: *Click HERE*

NAME

kl2000 – Newport MM2000/MM3000 motor controller

DESCRIPTION

The Newport (formerly Klinger) MM2000 and MM3000 motor controllers are supported by spec on both RS-232C and GPIB interfaces. The MM2000 is also supported on the ISA bus interface. On the serial interface, spec supports the daisy chaining available on the MM2000 and MM3000 controllers. All these controllers can be used both with DC motors (with encoders) and with the 1.5M-type stepper motors.

The controller is selected in the *config* file as

```
RS_MM2000 = device_name baud_rate number_of_motors
RS2_MM2000 = device_name baud_rate number_of_motors
GP_MM2000 = gpib_address number_of_motors
PC_MM2000 = base_address number_of_motors
```

depending on the interface being used. The second example above selects the daisy-chained RS-232C configuration.

When running the configuration editor, switch to the device configuration screen to select the appropriate interface. On the motor screen, select the `MM2000` or `MM2000_E` controller type for each motor to be controlled. The latter choice indicates encoders are being used.

By default, motors are automatically assigned to the controller channels in the same order they appear on the motor screen. If there is more than one controller, the channels are assigned to the controllers in the order in which the controllers appear on the motor device screen. Also available is the new nonconsecutive channel assignment feature that has been implemented for selected motor controllers. Controller unit and motor channel numbers can be selected and assigned on the second line of the motor configuration screen in any order. Not all channels need to be assigned.

When using the daisy chain feature, if automatic channel numbering is selected in the configuration file, the controller daisy-chain addresses must be assigned consecutively starting with address 0 using either the controller dip switch, the controller front panel menu or by sending a command over the serial interface. If the unit/channel configuration is used, the daisy-chain addresses will be derived from the channel number.

PARAMETERS

spec allows a number of additional parameters to be set in the configuration file and by the `motor_par()` function for these controllers. Use the configuration editor to establish the long-term values of the parameters in the configuration file. Use `motor_par()` to make changes to the parameters that

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

only are in effect during a particular spec session. To set the parameters from the configuration editor, use the **m** command from the main motor screen to access the optional-parameter screens.

The following parameters are associated with DC motor operation. The two-letter command sent to the motor controller associated with each parameter is given in parenthesis.

`motor_par(`*`motor`*`, "dc_proportional_gain" [, `*`value`*`])` – If *value* is given, sets the proportional gain, otherwise returns the current value. (`"KP"`)

`motor_par(`*`motor`*`, "dc_derivative_gain" [, `*`value`*`])` – If *value* is given, sets the derivative gain (or damping constant), otherwise returns the current value. (`"KD"`)

`motor_par(`*`motor`*`, "dc_integral_gain" [, `*`value`*`])` – If *value* is given, sets the integral gain, otherwise returns the current value. (`"KI"`)

`motor_par(`*`motor`*`, "dc_integration_limit" [, `*`value`*`])` – If *value* is given, sets the integrator limit, otherwise returns the current value. (`"IL"`)

`motor_par(`*`motor`*`, "dc_sampling_interval" [, `*`value`*`])` – If *value* is given, sets the derivative sampling interval, otherwise returns the current value. (`"DS"`)

`motor_par(`*`motor`*`, "dc_following_error" [, `*`value`*`])` – If *value* is given, sets the following-error threshold, otherwise returns the current value. (`"FE"`)

To set the gain for the `tth` motor, for example, use

```
motor_par(tth, "dc_proportional_gain", 1000)
```

The function

```
motor_par(tth, "dc_proportional_gain")
```

returns the current value of the `"dc_proportional_gain"` parameter.

Another `motor_par()` parameter is:

`motor_par(`*`motor`*`, "slop" [, `*`value`*`])` – If *value* is given, sets a threshold value in steps, below which spec suppresses position discrepancy messages when spec and the motor controller disagree about motor position, otherwise returns the current value.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

COMMAND PASS THROUGH

Command pass through is available using the following:

`motor_par(`*`motor`*`, "send", `*`cmd`*`)` - Sends the string `cmd` to the MM2000 channel associated with `motor`. For example, set `cmd` to `"SB1,3,8"` to set output bits 1, 3 and 8.

`motor_par(`*`motor`*`, "read", `*`cmd`*`)` - Sends the string `cmd` to the MM2000 channel associated with `motor`, as above, and returns a string containing the response.

JOYSTICK

To enable joystick control for a particular motor, use the command

```
motor_par(motor, "joy_on", channel)
```

where *`motor`* is the motor number and *`channel`* is the channel number, as explained in the Klinger manual for the `JY` command. To disable joystick control, use the command

```
motor_par(motor, "joy_off")
```

Not only does the `"joy_on"` command send the proper joystick enable codes to the MM2000/MM3000, but it also changes behavior within spec. In particular, position discrepancies between spec and the controller are automatically corrected in favor of the controller for all commands, except for the `sync` and `reconfig` commands. (A message is printed, though, when the position correction is made.) In addition, when joystick control is enabled, the motor controller is queried to read the current position for the motor on all `getangles` commands, not just when the motor is active, as is the joystick-disabled case.

Suggested macros for controlling the joystick are:

```
def joy_on '
    if ($# != 2) {
            print "Usage:  joy_on motor channel"
            exit
    }
    _check0 "$1"
    motor_par($1,"joy_on",$2)
'
def joy_off '
    if ($# != 1) {
            print "Usage:  joy_off motor"
            exit
    }
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
        _check0 "$1"
        motor_par($1,"joy_off")
    '
```

To make things even easier, you could define macros such as

```
    def on_joy '
        joy_on tth 1
        joy_on th  2
        joy_on chi 3
        joy_off phi 4
    '
```

and

```
    def off_joy '
        joy_off tth
        joy_off th
        joy_off chi
        joy_off phi
    '
```

LIMITATIONS

The MM2000/MM3000 motor controller only allows its internal position registers to be set to zero. Thus the `chg_dial()` function (used by the `set_dial` macro) will not accept nonzero position arguments. Similarly, if there is a motor position discrepancy between spec and the controller, if you choose to change the controller register position to agree with spec and if the spec dial position is not zero, the spec dial position will instead be changed to agree with the controller, and the the spec *offset* parameter will be changed to keep the same spec user-units position.

VERSION

The above description describes the MM3000 firmware version 2.3.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP