

NAME

PMC DCX-100 – PMC Corporation DCX-100 Motor Controllers

DESCRIPTION

The PMC Corporation DCX-100 motor controller is an ISA PC card that has eight module positions. All eight modules can be used to control individual motors or one of the modules can instead be an RS-232C interface. `spec` can control the DCX-100 cards either locally using the ISA memory-mapped interface or else remotely over the RS-232C interface. In the latter case, the send, receive and ground signals of the RS-232C interface modules from each of multiple boards can be wired in parallel, and all can be controlled by `spec` through one serial port of the controlling computer. (Other module configurations are possible, but are not currently supported by `spec`.) (Note, the DCX-100 boards are used as OEM components in motor controllers sold by the Dynaoptic Motion company.)

The controller is selected on the `edconf` devices screen along the lines of:

MOTORS	DEVICE	ADDR	<>MODE	NUM	<>TYPE
YES		0xd000		8	PMC Corp DCX-100 PC Board
YES	/dev/ttyS0	<>	9600	16	PMC Corp DCX-100 (Serial)

The `config` file will have:

```
PC_DCX    = mem_map_address number_of_motors
RS_DCX    = device_name baud_rate number_of_motors
```

for the above two configurations.

For the PC board configuration, the memory-mapping address is selected by the on-board rotary switch SW1. Position 0 corresponds to address 0xd000, position 1 corresponds to address 0xd100, etc. For the serial (or RS-232C) configuration, boards are also selected by address. That address is used for the individual motor channel configuration as follows.

DCX motors on serial ports require unit/channel configuration. Unit numbers are assigned to each DCX controller in the order they appear on the Device screen, starting with unit 0. Channel numbers are of the form X1 to X8 where X is the board number, matching the SW1 setting, and the number is module number on the board.

MOTOR 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 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. ("SG")

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

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

`motor_par(motor, "dc_integration_limit" [, value])` – If *value* is given, sets the integration 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. ("FR")

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

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.

SPECIAL COMMANDS

`motor_par(motor, "wdelay" [, value])` – If *value* is given, sets the delay before writing a command to *value* microseconds, otherwise returns the current value. The default value is 20000 (20 msec).

`motor_par(motor, "rdelay" [, value])` – If *value* is given, sets the delay before reading a response to *value* microseconds, otherwise returns the current value. The default value is 50000 (50 msec).

`motor_par(motor, "status_bits")` – Returns the value returned by the controller's TS command.

`motor_par(motor, "send", cmd)` – Sends the string *cmd* to the DCX unit associated with *motor*.

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

`motor_par(motor, "usend", cmd)` – Sends the string *cmd* to the DCX unit associated with *motor*, but doesn't address the command to a specific motor or add a line terminator. Thus *cmd* must include the full DCX command syntax.

`motor_par(motor, "uread", cmd)` – Sends the string *cmd* to the DCX unit associated with *motor*, as above, and returns a string containing the response.