

Pulsar2 LX200 extension

04.06.2017.

Format for getting a parameter:

#:YGx# reply:aaaa,bbbb#

Format for parameter setting

#:YSxaaaa,bbbb# reply '1' if OK or 0 if not interpreted.

Allowed x values at left, possible answers or allowed values at right.

'A'	guiding speed	1-9,1-9# (1 = 0.1x sidereal speed)
'B'	center speed	1-999,1-999# (unit is sidereal speed)
'C'	find speed	1-999,1-999#
'D'	slew speed	1-999,1-999#
'E'	goto speed	1-999,1-999#
'I'	geographical position	+47.3996,-018.7005# (attention, EAST is negative)
'M'	mount type	3/2/1# (3=AltAz, 2=fork, 1=German)
'N'	physical side of pier	0/1# (East=0/West=1)
'P'	PEC corr	0/1,0/1# off/on
'Q'	polecross	0/1# (No/Yes)
'R'	refract corr	0/1,0/1# (off/on)
'S'	speed rate sel	0,0-6,0#
	STOP	0,0
	SID	1,0
	LUN	2,0
	SOL	3,0
	USER1	4,0
	USER2	5,0
	USER2	6,0

#:YGS# - reading the actual rate

#:YSS1,0# - setting the desired rate (sidereal, user, other)

For setting or reading the speed of the individual user rates, use these commands:

#:YGZ1# get userrate1(reply e.g:+0.0000000,+0.0000000#) unit rad/min

#:YSZ1# set userrate1

#:YGZ2# get userrate2

#:YSZ2# set userrate2

#:YGZ3# get userrate3

#:YSZ3# set userrate4

For setting a specific value for an immediate offset rate use this command (radian/min of time, 7 decimals) Send all zeros to reset sidereal rate.

#:YSZ+0.1234567,-0.1234567# (max allowed value 4.1887902 or 4 deg/sec) reply 1 if ok, 0 if failed.

#:YGZ# (for getting the actual value)

#:YGX# Home position eg:12.3456,012.3456# (alt/az)

#:Mgw0013# pulse guide moves 10 ms west, 3 is a dummy character (0-9), you can use the first 3 values only. The min motion is thus 10ms, the max is 999x10ms, in command #:Mgw9993# Returns nothing

#:Mge0013#

#:Mgn0013#

#:Mgs0013#

When this command is sent Pulsar2 will automatically execute it with the actual guide speed. The AG port works as before and normal move commands can still be used for guiding. Obviously only one channel per time can be active.

#:YGa#	autostop local enable	0/1#,second parameter always 0
'd'	display data	G-32 bytes of hand controller display +'# S-1 byte of hand controller keyboard key
'b'	backlash	0-9:0-59# (first param min,second sec)
'g'	goto current	0-2000, 0-2000# (mA)
'e'	mot.res	100/200/400,100/200/400# (step number)
'l'	lat/long	eg:-12.3456,+012.3456#
'm'	maingear	100-6000,100-6000#
'n'	rotation	0/1,0/1# (Right=0/Left=1)
'p'	ramp	1-10,1-10# (Ra and DEC)
'r'	reduction	100-6000,100-6000#
's'	stop current	0-2000, 0-2000# (mA)
't'	track current	0-2000, 0-2000# (mA)

GET only commands (can only be read)

#:YGh#	IsHomeSet	1# / 0#
#:YGi#	IsSlewing	1# / 0#
#:YGj#	IsMountParking	1# / 0#
#:YGk#	IsMountParked	1# / 0#
#:YGW#	Flip status	1# / 0# (1 flip needed, 0 normal state)
#:YV#	Version	'PULSAR Vx.xxx,2008.10.10#'
Parking and unparking		
'#:YH#'	Park	1# / 0# (1 executed, 0 failed)
'#:YL#'	Unpark	1# / 0#

PEC Commands:

'#:YPR#	PEC start recording
'#:YPD#	PEC table download //Download to PC
'#:YPU#	PEC table upload //Upload from PC
'#:YPC#	PEC synchronize
'#:YPL#	PEC load //resets the status before recording

'#:YPT# PEC stop recording
 '#:YPG# PEC read counter (e.g..123#) counter range 0-300
 '#:YPS# PEC saving recorded data to EEPROM

PEC recording example:

#:YPR#
 #:YPG# returns the value of the counter, stops when ready.
 Or stops with #:YPT#
 #:YPT#
 #:YPS# saves table in EEPROM

PEC Upload example:

Command reply
 #:YPU# „S”
 300*(RA_PEC_DATA+DEC_PEC_DATA)
 +1*(LEFT_ENC_POS,RIGHT_ENC_POS)
 +9*(DUMMY+DUMMY)
 (summa 2480 bytes)
 +sent data checksum: 1 byte „O”
 #:YPL# „1” -loads new PEC table

PEC Download example:

Command reply
 #:YPD# „S” + (int)PEC_SIZE -2480
 „o” get 40 bytes
 „o” get 40 bytes, 5 PEC records , 40 bytes
 .
 .
 „o” get 40 bytes „#” ready

PEC Data types:

PEC_RECORD RA_PEC_DATA,DEC_PEC_DATA
 RA_PEC_DATA,DEC_PEC_DATA= single float type (4 byte) ,PEC correction in radian
 RA_ENC_POS =float type (4 byte)
 DUMMY =float type (4 byte)

In PEC_RECORD[301] is the PEC encoder position (Hall sensor on the RA worm)