Command Reference Manual

MODEL : PMC-xxxx

(Piezoelectric Motor Controller)

VER. 105

Piezoelectric Technology Co., Ltd.

#501 Sinnae Technotown, 485 Sangbong1 – dong,
Jungrang –gu, Seoul, Korea, 131 - 863
Tel. +82- 2 - 3421 – 0370~3 , Fax. +82 – 2 - 3421 – 0374
http://www.piezo-tech.com

Revision History

Version	Date	Changes	Status	Author/Approver
100	2013/03/21	First revision	draft	Jongmin Choi
101	2013/04/16	Added Velocity command	draft	Jongmin Choi
102	2013/08/26	Added PT Table	draft	Jongmin Choi
103	2013/10/18	Added Alarm list Added Received command from PMC1202 Modified command list	draft	Jongmin Choi
104	2013/10/28	Added MR-Encoder and MR-Sensor error of alarm list	draft	Jongmin Choi
105	2013/12/3	Added Encoder error of alarm list	draft	Jongmin Choi

– The information is subject to change without notice for technical improvement

CONTENTS

1. Communication Function of PMC-xxxx

1-1. Connecting the PC and PMC-xxxx Controller allows following operations from the PC :

1-1-1. Writing and reading out the setting value

1-1-2. Reading out the measurement value

1-1-3. Reading out the control output status

1-1-4. Operating the control input

2. Communication Specification

Baud rate	115200 bps
Transmission code	ASC II
Data length	8 bit
Stop bit length	1 bit
Parity check	Nil
Data classification	STX . ETX

3. Communication Data Format (Command)

Type	STX	COMMAND	ETX
Send to controller	’>’		‘₩r’
Receive from controller	’<’		‘₩r’

Type	STX	COMMAND	SPACE1	PARAMETER1	ETX
Send to controller	’>’		0x20	optional	‘₩r’
Receive from controller	’<’		0x20	optional	‘₩r’

Type	STX	COMMAND	SPACE1	PARAMETER1	SAPCE2	PARAMETER2	ETX
Send to controller	’>’		0x20	optional	0x20	optional	‘₩r’
Receive from controller	’<’		0x20	optional	0x20	optional	‘₩r’

STX	The code showing the head of transmit data/received data (‘>’ = 0x3e) (‘<’ = 0x3c).
COMMAND	Selects from the column of command on the command list to set.
SPACE1	Shows the separation between Command and Parameter1.
PARAMETER1	Numeric-string1 which is optional.
SPACE2	Shows the separation between Parameter1 and Parameter2.
PARAMETER2	Numeric-string1 which is optional.
ETX	The code showing the completion of transmit data/received data (‘₩r’ = 0x0d).

4. Command list

4-1. Motion Commands (Point to Point – PTP)

Command	Parameter1	Parameter2	Description
ma	-2,147,000,000 ~ +2,147,000,000	-	Move to the absolute target position, [counts]. The target position can be specified absolute position, using the absolute position parameter. 1) Parameter1 : Assigning an absolute target position, [counts] 2) Send to controller : printf(”>ma 1000₩r”); 3) Receive from controller : printf(“<ma 1000₩r”);
mr	-2,147,000,000 ~ +2,147,000,000	-	Move to the relative target position, [counts]. The target position can be specified relatively to the current desired position, using the relative position parameter. 1) Parameter1 : Assigning an relative target position, [counts] 2) Send to controller : printf(”>mr 10000₩r”); 3) Receive from controller : printf(“<mr 10000₩r”);
home	-	-	Move to the home position. 1) Send to controller : printf(”>home₩r”); 2) Receive from controller : printf(“<home₩r”);
stop	-	-	Aborts the motion immediately. 1) Send to controller : printf(”>stop₩r”); 2) Receive from controller : printf(“<stop₩r”);

4-2. Open-Loop Commands

Command	Parameter1	Parameter2	Description
openmode	0/1	-	Define mode of open-loop 1) 0 : Time(ms) [default], 1 : pulse 2) Send to controller : printf(”>openmode 0₩r”); 3) Receive from controller : printf(”<openmode 0₩r”);
duration	1 ~ 600,000	-	Define the moving on-time(pulse). 1) Send to controller : printf(”>duration 1000₩r”); 2) Receive from controller : printf(”<duration 1000₩r”);
interval	1 ~ 600,000	-	Define the time between the start of each step 1) Parameter1 : step interval (unit : ms) 2) Send to controller : printf(”>interval 1000₩r”); 3) Receive from controller : printf(”<interval 1000₩r”);
cycle	1 ~ +2,147,000,000	-	Define the optional number of run. 1) Send to controller : printf(”>cycle 10₩r”); 2) Receive from controller : printf(”<cycle 10₩r”);
re	-	-	Motor runs in reverse. 1) Send to controller : printf(”>re₩r”); 2) Receive from controller : printf(”<re₩r”);
fo	-	-	Motor runs forward. 1) Send to controller : printf(”>fo₩r”); 2) Receive from controller : printf(”<fo₩r”);
bi	-	-	Move to both direction. 1) Send to controller : printf(”>bi₩r”); 2) Receive from controller : printf(”<bi₩r”):

4-3. Status Commands (It is a read only commands, indicating the recording status.)

Command	Parameter1	Parameter2	Description
cp	-	-	This command reports the actual controller position (Encoder Value). 1) Send to controller : printf(”>cp₩r”); 2) Receive from controller : printf(”<cp %ld₩r”, current_position);
status	-	-	Get the alarm status. 1) Send to controller : printf(”>status₩r”); 2) Receive from controller : printf(”<status %d₩r”, alarm_status);
velr	-	-	Get the velocity value. 1) Send to controller : printf(”>velr₩r”); 2) Receive from controller : printf(”<vel %d₩r”, velocity);

inform	-	-	Reports the configuration setting of controller
			1) Send to controller : printf(”>inform₩r”);
			2) Receive from controller :
			printf(”<freq %ld₩r”, frequency);
			printf(”<volt %d₩r”, volta);
			printf(”<encoder %d₩r”, encoder_type);
			printf(”<resolution %ld₩r”, resolution);
			printf(”<encswap %d₩r”, encoder swapped);
			printf(”<vel %ld₩r”, velocity);
			printf(”<offset %ld₩r”, Home_offset);
			printf(”<lm %ld₩r”, reverse_limit);
			printf(”<lp %ld₩r”, forward_limit);
			printf(”<st %ld₩r”, stroke);

ver	-	-	Get the controller firmware version. 1) Send to controller : printf(”>ver₩r”); 2) Receive from controller : printf(”<ver %ld %ld₩r”, yymmdd, version)

4-4. Alarm list (It is a read only)

When power-on, the status of PMC-1202 is reset to 0x1000.

Alarm bit (Hex)	Function Name	Description
8000	MOTOR_RUNNING	1 = Motor is running
4000	-	
2000	-	
1000	HOME_MISSING	1 = Home position is not known
800	-	
400	-	
200	-	
100	ILLEGAL_CMD	1 = The command was improperly formatted (When next command received, bit is zero)
80	PARAMETER_ERR	1 = The parameter is out of range (When next command received, bit is zero)
40	MR_ENCODER_ERR	1 = MR-Encoder error
20	MR_SENSOR_ERR	1 = MR-Sensor error
10	ENCODER_ERR	1 = Encoder error (swapped line, connection missing, encoder error)
8	POSITION_ERR	1 = Position error - Resolution: 5um (Target position ±**3**pulse ≠ current position) - Resolution: 1um (Target position ±**3**pulse ≠ current position) - Resolution: 100nm (Target position ±**5**pulse ≠ current position) - Resolution: 10nm (Target position ±**10**pulse ≠ current position)
4	ENCODER_Z_ERR	1 = Z signal error (Contact to Piezoelectric Technology Co., Ltd)
2	-	
1	OVER_TEMP	1 = The specified temperature is exceeding (After 3minute, the PMC-1202 is normal operation)

4-5. Configuration Commands

Command	Parameter1	Parameter2	Description
freq	20 ~ 100	-	Define the driving frequency. 1) Parameter 1 : Driving frequency (20~100KHz) 2) Send to controller : printf(">freq 68₩r"); 3) Receive from controller : printf("<freq 68₩r");
duty	1~48	-	Define the duty(Open-loop only) 1) Parameter 1 : Duty (1 ~ 48%) 2) Send to controller : printf(">duty 25₩r"); 3) Receive from controller : printf("<duty 25₩r");
volt	16 ~ 35	-	Define the voltage. 1) Parameter 1 : Voltage (unit : V) 2) Send to controller : printf(">volt 30₩r"); 3) Receive from controller : printf("<volt 30₩r");
encoder	1 ~ 4	-	Define the encoder type. 1) Parameter 1 : Encoder type 1-1) A/B signal : 1 1-2) A/B/Z signal : 2 1-3) Opt-pass : 3 1-4) MR-Sensor : 4 1-5) MR-Encoder : 5 2) Send to controller : printf(">encoder 1₩r"); 3) Receive from controller : printf("<encoder 1₩r");
resolution	10 100 1000 5208	-	Define the resolution of encoder. 1) Parameter 1 : nanometer (unit : nm) 2) Send to controller : printf(">resolution 1000₩r"); 3) Receive from controller : printf("<resolution 1000₩r");
encswap	0 ~ 1	-	Swap Encoder A and Encoder B inputs 1) Parameter 1 : Input is inverted or not inverted 2) Send to controller : printf(">encswap 1₩r"); 3) Receive from controller : printf("<encswap 1₩r");
vel	3 ~ 40	-	Define the velocity 1) Parameter 1 : Velocity (unit : mm/s) 2) Send to controller : printf(">vel 10₩r"); 3) Receive from controller : printf("<vel 10₩r");
reset	-	-	Warm-start, it is same as the cold-start. 1) Send to controller : printf(">reset₩r"); 2) Receive from controller : printf("<reset₩r");
offset	-2,147,000,000 ~ +2,147,000,000	-	Define the offset of HOME position. 1) Send to controller : printf(">offset 100₩r"); 2) Receive from controller : printf("<offset 100₩r");
save	-	-	Save the current configuration to the internal flash memory permanently 1) Send to controller : printf(">save₩r"); 2) Receive from controller : printf("<save₩r");

4-6. Control filters Commands

Command	Parameter1	Parameter2	Description
kp	0 ~ +32767	-	KP is the position loop gain. 1) Parameter 1 : Proportional position gain. 2) Send to controller : printf(”>kp 100₩r”); 3) Receive from controller : printf(”<kp 100₩r”);
ki	0 ~ +32767	-	KI is the velocity loop integral term gain. 1) Parameter 1 : Integral gain, in velocity loop. 2) Send to controller : printf(”>ki 90₩r”); 3) Receive from controller : printf(”<ki 90₩r”);
kd	0 ~ +32767		KD is the velocity loop overall gain multiplier 1) Parameter 1 : Derivative gain, in velocity loop. 2) Send to controller : printf(”>kd 10₩r”); 3) Receive from controller : printf(”<kd 10₩r”);
kf	0 ~ +32767	-	IIR filter 1) Parameter 1 : Specifies the parameter of position controller IIR filter. 2) Send to controller : printf(”>kf 1000₩r”); 3) Receive from controller : printf(”<kf 1000₩r”);

♣ **Under most circumstances, this commands is used only by the tuning environment.**

♣ **This commands is not supported by the standard firmware version of the controller.**

♣ **Please consult Piezoelectric Technology experts for more information.**

4-7. Position-Time Commands

Command	Parameter1	Parameter2	Description
pt1~pt50	-2,147,000,000 ~ +2,147,000,000	1 ~ 600,000	Position-Time Table_x 1) Parameter 1 : Assigning an absolute target position, [counts] 2) Parameter 2 : step interval (unit : ms) 3) Send to controller : printf(”>pt1 100 10₩r”); 4) Receive from controller : printf(”<pt1 100 10₩r”); 5) Send to controller : printf(”>pt2 -100 100₩r”); 6) Receive from controller : printf(”<pt2 -100 100₩r”); 7) Send to controller : printf(”>pt50 1000 100₩r”); 8) Receive from controller : printf(”<pt50 1000 100₩r”);
delete	0 ~ 50	-	Delete the PT table. (0 : all clear) 1) Send to controller : printf(”>delete 10₩r”); 2) Receive from controller : printf(”<delete 10₩r”);
save	-	-	Save the current condition to the internal flash memory permanently. 1) Send to controller : printf(”>save₩r”); 2) Receive from controller : printf(”<save₩r”);
ptread	-	-	Reports the actual controller PT table. 1) Send to controller : printf(”>ptread₩r”); 2) Receive from controller : printf(”<ptread %d %d %d₩r”, number, position, interval);
step	0 ~ +2,147,000,000	-	Define the optional number of run. (0 : unlimited) 1) Send to controller : printf(”>step 100₩r”); 2) Receive from controller : printf(”<step 100₩r”);
ptstart	-	-	Motor runs. 1) Send to controller : printf(”>ptstart₩r”); 2) Receive from controller : printf(”<ptstart₩r”);
stop	-	-	Aborts the motion immediately. 1) Send to controller : printf(”>stop₩r”); 2) Receive from controller : printf(”<stop₩r”);

5. Contact Information

Contact your local distributor or Piezoelectric Technology Co., Ltd.

5-1. Head Office

Room 501, 503 Sinnae Technotown, 485 Sangbong1dong, Jungrang-gu, Seoul, Korea. 131-863
Tel: +82-2-3421-0370~3
Fax: +82-2-3421-0374
Web site: www.piezo-tech.com
Email: piezo-tech@piezo-tech.com