

3. Serial command

Here the serial command is denoted.

3.1 Serial communication parameters

The serial communication parameters are below.

Parameter	Value
BaudRate	115200bps
DataBit	8bit
StopBit	1bit
Parity	None
Flow Control	None

3.1 Command list

Here the command list of serial communication is listed.

[Definition]

* “ACK1”

When a PWM-board gets a command, it replies “ACK1” within 30msec.

* “ACK2”

When a PWM-board finishes a motion and scenario operation, it replies “ACK2”.

However a PWM-board2 belong a latest command action when you send a new command during a operation a current command.

* SUM

SUM is calculated using only data except for a command and an ack.

[Set Position]

You can send a position data to a robot, and a robot will move to this position immediately.

SPD	CH0	...	CH23	SUM
-----	-----	-----	------	-----

FDh	Set position command	
00h~07h	Speed (0,1,2,3,4,5,6,7,10 and 11)	
00h~B4h	From 0 to 180 [degree] on each joint	
00h~7Fh	The value delete 1 bit of sum of all data except for command	
06h	Ack command	

[Read Position]

You can read a current position data from a robot.

[TX]

CMD

[RX]

ACK1	CH0	...	CH23	SUM
------	-----	-----	------	-----

CMD	FCh	Read Position command
CH0...CH23	00h~B4h	From 0 to 180 [degree] on each joint
SUM	00h~7Fh	The value delete 1 bit sum of al data except for command
ACK1	06h	Ack command

[Set Home Position]

You can register a current robot position as a robot home position.

[TX]

CMD

[RX]

ACK1

CMD	FBh	Set Position command
ACK1	06h	Ack command

[Read Home Position]

You can read a current robot home position from a robot even if current robot position isn't a robot home position.

[TX]

CMD

[RX]

ACK1	CH0	...	CH23	SUM
------	-----	-----	------	-----

CMD	FAh	Set position command
CH0...CH23	00h~B4h	From 0 to 180 [degree] on each joint
SUM	00h~7Fh	The value delete 1 bit sum of al data except for command
ACK1	06h	Ack command

[Write Position in a motion list]

You can write a position in a motion list. This operation writes a position data into a EEPROM directly, therefore, a robot doesn't move in this operation.

[TX]

CMD	MOT_NO	POS_INDEX	SPD	CH0	...	CH23	SUM
-----	--------	-----------	-----	-----	-----	------	-----

[RX]

ACK1

CMD	F9h	Write position command
MOT_NO	00h~27h	The selected motion number from 40
POS_INDEX	00h~27h	The sequence in 40 position
SPD	00h~07h	Speed (0,1,2,3,4,5,6,7,10 and 11)
CH0...CH23	00h~B4h	From 0 to 180 [degree] on each joint
SUM	00h~7Fh	The value delete 1 bit sum of al data except for command
ACK1	06h	Ack command

[Read Position in a motion list]

You can read a position data from a motion list in a PWM-board2.

[TX]

CMD	MOT_NO	POS_INDEX	SUM
-----	--------	-----------	-----


[RX]

ACK1	SPD	CH0	...	CH23	SUM
------	-----	-----	-----	------	-----

CMD	F8h	Read position command
MOT_NO	00h~27h	The selected motion number from 40
POS_INDEX	00h~27h	The sequence in 40 position
SPD	00h~07h	Speed (0,1,2,3,4,5,6,7,10 and 11)
CH0...CH23	00h~B4h	From 0 to 180 [degree] on each joint
SUM	00h~7Fh	The value delete 1 bit sum of al data except for command
ACK1	06h	Ack command

[Write the number of Position]

You can write the number of position in a motion list. PWM-board2 executes a position data upto this number in a motion data. Therefore you have to set this number as same as the length of a motion list.

 Danger	<p>If you set smaller number than the length of a motion list, PWM-board2 execute middle of a position in a motion list. Besides if you set bigger number than the length of a motion list, PWM-board2 execute unknown memory data, then a robot will excursion.</p>
---	--

[TX]

CMD	MOT_NO	COUNT	SUM
-----	--------	-------	-----

[RX]

ACK1

CMD	F7h	Write number of position command
MOT_NO	00h~27h	The number of motion
COUNT	00h~27h	The number of a position index in a play motion
SUM	00h~7Fh	The value delete 1 bit sum of al data except for command
ACK1	06h	Ack command

[Read the number of Position]

You can read the number of registered position number in a PWM-board2.

[TX]							
CMD	MOT_NO	SUM					
[RX]							
ACK1	COUNT	SUM					
CMD	F6h	Read number of position command					
MOT_NO	00h~27h	The number of motion					
COUNT	00h~27h	The number of a position index in a play motion					
SUM	00h~7Fh	The value delete 1 bit sum of al data except for command					
ACK1	06h	Ack command					

[Write motion number in a scenario list]

You can write a motion number in a scenario list.

[TX]					
CMD	SCEN_NO	MOT_INDEX	MOT_NO	SUM	
[RX]					
ACK1					
CMD	F5h	Write motion number in a scenario list			
SCEN_NO	00h~02h	The scenario number you register			
MOT_INDE	00h~C7h	The number of scenario			
MOT_NO	00h~27h	The number of a position index in a play motion			
SUM	00h~7Fh	The value delete 1 bit sum of al data except for command			
ACK1	06h	Ack command			


[Read motion number in a scenario list]

You can read a motion list from a scenario in a PWM-board2.

[TX]							
CMD	SCEN_NO	MOT_INDEX	SUM				
[RX]							
ACK1	MOT_NO	SUM					
CMD	F4h	Read motion number in a scenario list					
SCEN_NO	00h~02h	The number of scenario					
MOT_INDEX	00h~C7h	The index of a motion you select					
MOT_NO	00h~27h	The motion number					
SUM	00h~7Fh	The value delete 1 bit sum of al data except for command					
ACK1	06h	Ack command					

[Write the number of motion data]

You can write the number of motion in a motion list. PWM-board2 executes a motion data upto this number in a scenario data. Therefore you have to set this number as same as the length of a scenario list.

 Danger	<p>If you set smaller number than the length of a scenario list, PWM-board2 execute middle of a motion in a scenario list. Besides if you set bigger number than the length of a scenario list, PWM-board2 execute unknown memory data, then a robot will excursion.</p>
---	--

[TX]

CMD	SCEN_NO	COUNT	SUM
-----	---------	-------	-----

[RX]

ACK1

CMD	F3h	Write the number of Motion data					
SCEN_NO	00h~02h	Scenario number					
COUNT	00h~C7h	The motion number					
SUM	00h~7Fh	The value delete 1 bit sum of al data except for command					
ACK1	06h	Ack command					

[Read the number of motion]

You can read the number of registered motion number in a PWM-board2.

[TX]						
CMD	SCEN_NO	SUM				
[RX]						
ACK1	COUNT	SUM				
CMD	F2h	Read the number of motion				
SCEN_NO	00h~02h	The scenario number				
COUNT	00h~C7h	The motion number				
SUM	00h~7Fh	The value delete 1 bit sum of al data except for command				
ACK1	06h	Ack command				

[Play a motion data]

You can play a registered motion data.

[TX]			
CMD	NO	SUM	
[RX]			
ACK1			
ACK2			
CMD	EFh	Play motion data command	
NO	00h~27h	The number of motion	
SUM	00h~7Fh	The value delete 1 bit sum of al data except for command	
ACK1	06h	Ack command	
ACK2	07h	Ack command	

[Play a scenario data]

You can play a registered scenario data.

[TX]

CMD	SCEN_NO	SUM
-----	---------	-----

[RX]

ACK1
ACK2

CMD	EEh	Play scenario data command
SCEN_NO	00h~02h	The number of scenario
SUM	00h~7Fh	The value delete 1 bit sum of al data except for command
ACK1	06h	Ack command
ACK2	07h	Ack command

[Write Trim data]

You can write a trim data into a PWM-board2.

[TX]

CMD	CH0	...	CH23	SUM
-----	-----	-----	------	-----

[RX]

ACK1

CMD	E9h	Write Trim data command
CH0...CH23	00h~27h	From -20 to 20 [degree] on each joint
SUM	00h~7Fh	The value delete 1 bit sum of al data except for command
ACK1	06h	Ack command

[Read Trim data]

You can read a trim data from a PWM-board2.

[TX]

CMD

[RX]

ACK1	CH1	...	CH23	SUM
------	-----	-----	------	-----

CMD	E8h	Read Trim data command
CH0~CH23	00h~27h	From -20 to 20[degree] on each joint
SUM	00h~7Fh	The value delete 1 bit sum of al data except for command
ACK1	06h	Ack command