



MEAT series

Intelligent Stepper Motor System Guideline

MINDMAN Industrial Company Limited

TEL : (02)25913001 FAX : (02)25912822

<http://www.mindman.com.tw>

E-mail: mindman@mindman.com.tw

Catalog

1. (PN) Parameter Table	3
2. Commands Table (Editable by Computer)	6
2-1) Motion control statements	7
2-2) Homing and Coordinate Configuration Statements	8
2-3) Servo Control Statem	8
2-4) Terminal Output Statements	10
2-5) Output Port Control Statements	10
2-6) Variable Statements	11
2-7) Program Flow Control Statements	14
2-8) Inquiry Statements	17
2-9) Other Statements	18
3. ERR error code reference	19
4. Modbus interface address table	20
5. Hyper Terminal Connection	22
6. Program Download and Update	24

1. (PN) Parameter Table

Parameter	Default	Range/Name	Function	Pattern of the application																		
PN1 MD	H0015		Select the working mode																			
			Bit 0 H000 ¹ <table border="1"> <tr> <th>Value</th> <th>Function</th> </tr> <tr> <td>0</td> <td>Pulse input control mode (accelerate/decelerate by the front-end controller)</td> </tr> <tr> <td>1</td> <td>Pulse input control mode (accelerate/decelerate by the buffer of driver)</td> </tr> <tr> <td>5</td> <td>Terminal mode</td> </tr> </table> Char 1 H000 ¹ CSC2 Terminal ID(TID) setting : 0~7 (Please don't set to 0, when connect to human machine interface).		Value	Function	0	Pulse input control mode (accelerate/decelerate by the front-end controller)	1	Pulse input control mode (accelerate/decelerate by the buffer of driver)	5	Terminal mode										
Value	Function																					
0	Pulse input control mode (accelerate/decelerate by the front-end controller)																					
1	Pulse input control mode (accelerate/decelerate by the buffer of driver)																					
5	Terminal mode																					
PN2	H0000		Set the servo control mode	A																		
			Bit 0 H000 ¹ <table border="1"> <tr> <th>Value</th> <th>Rotational direction of motor</th> </tr> <tr> <td>0</td> <td>Rotate counter clockwise when input positive command</td> </tr> <tr> <td>1</td> <td>Rotate clockwise when input positive command</td> </tr> </table>		Value	Rotational direction of motor	0	Rotate counter clockwise when input positive command	1	Rotate clockwise when input positive command												
			Value		Rotational direction of motor																	
			0		Rotate counter clockwise when input positive command																	
1	Rotate clockwise when input positive command																					
Bit 1 H000 ¹ <table border="1"> <tr> <th>Value</th> <th>Input pulse type</th> <th>Value</th> <th>Input pulse type</th> </tr> <tr> <td>0</td> <td>PLS/DIR pulse+direction</td> <td>1</td> <td>CW/CCW double pulse</td> </tr> <tr> <td>3</td> <td>AB wave input</td> <td></td> <td></td> </tr> </table>	Value	Input pulse type	Value	Input pulse type	0	PLS/DIR pulse+direction	1	CW/CCW double pulse	3	AB wave input			0・1									
Value	Input pulse type	Value	Input pulse type																			
0	PLS/DIR pulse+direction	1	CW/CCW double pulse																			
3	AB wave input																					
Bit 2 Reserved by Manufacturer Bit 3 H000 ¹ <table border="1"> <tr> <th>Value</th> <th>AUTO RUN</th> <th>value</th> <th>AUTO RUN</th> </tr> <tr> <td>0</td> <td>Doesn't execute the program automatically when start</td> <td>1</td> <td>Execute the program automatically when start</td> </tr> </table>	Value	AUTO RUN	value	AUTO RUN	0	Doesn't execute the program automatically when start	1	Execute the program automatically when start	Effective for controller version													
Value	AUTO RUN	value	AUTO RUN																			
0	Doesn't execute the program automatically when start	1	Execute the program automatically when start																			
PN3	H0020	When PN3=H0022, H0023, H0026, H0027, use IN1 as the signal of origin	Set the method of zeroing	5																		
			Bit 0 H000 ⁰ <table border="1"> <tr> <th>Value</th> <th>Zeroing direction (Effective when MD=1・5)</th> </tr> <tr> <td>0</td> <td>Return to zero with negative direction, enable CCWHC contact as the signal of origin (PN5=H030³ IN3_B contact)</td> </tr> <tr> <td>1</td> <td>Return to zero with positive direction, enable CWHC contact as the signal of origin (PN5=H030³ IN2_B contact)</td> </tr> <tr> <td>2</td> <td>Return to zero with negative direction, enable HORG contact as the signal of origin (PN4=H000⁰ IN1_A contact)</td> </tr> <tr> <td>3</td> <td>Return to zero with positive direction, enable HORG contact as the signal of origin (PN4=H000⁰ IN1_A contact)</td> </tr> <tr> <td>4</td> <td>Return to zero with negative direction, use VH speed to bounce the boundary of mechanism to search the origin</td> </tr> <tr> <td>5</td> <td>Return to zero with positive direction, use VH speed to bounce the boundary of mechanism to search the origin</td> </tr> <tr> <td>6</td> <td>Return to zero with negative direction, enable HORG contact as the signal of origin (PN4=H000⁰ IN1_B contact)</td> </tr> <tr> <td>7</td> <td>Return to zero with positive direction, enable HORG contact as the signal of origin (PN4=H000⁰ IN1_B contact)</td> </tr> </table>		Value	Zeroing direction (Effective when MD=1・5)	0	Return to zero with negative direction, enable CCWHC contact as the signal of origin (PN5=H030 ³ IN3_B contact)	1	Return to zero with positive direction, enable CWHC contact as the signal of origin (PN5=H030 ³ IN2_B contact)	2	Return to zero with negative direction, enable HORG contact as the signal of origin (PN4=H000 ⁰ IN1_A contact)	3	Return to zero with positive direction, enable HORG contact as the signal of origin (PN4=H000 ⁰ IN1_A contact)	4	Return to zero with negative direction, use VH speed to bounce the boundary of mechanism to search the origin	5	Return to zero with positive direction, use VH speed to bounce the boundary of mechanism to search the origin	6	Return to zero with negative direction, enable HORG contact as the signal of origin (PN4=H000 ⁰ IN1_B contact)	7	Return to zero with positive direction, enable HORG contact as the signal of origin (PN4=H000 ⁰ IN1_B contact)
			Value		Zeroing direction (Effective when MD=1・5)																	
			0		Return to zero with negative direction, enable CCWHC contact as the signal of origin (PN5=H030 ³ IN3_B contact)																	
			1		Return to zero with positive direction, enable CWHC contact as the signal of origin (PN5=H030 ³ IN2_B contact)																	
			2		Return to zero with negative direction, enable HORG contact as the signal of origin (PN4=H000 ⁰ IN1_A contact)																	
3	Return to zero with positive direction, enable HORG contact as the signal of origin (PN4=H000 ⁰ IN1_A contact)																					
4	Return to zero with negative direction, use VH speed to bounce the boundary of mechanism to search the origin																					
5	Return to zero with positive direction, use VH speed to bounce the boundary of mechanism to search the origin																					
6	Return to zero with negative direction, enable HORG contact as the signal of origin (PN4=H000 ⁰ IN1_B contact)																					
7	Return to zero with positive direction, enable HORG contact as the signal of origin (PN4=H000 ⁰ IN1_B contact)																					
Bit 1 (Baud Rates) H002 ⁰ <table border="1"> <tr> <th>Value</th> <th>Human machine interface</th> <th>Value</th> <th>Human machine interface</th> </tr> <tr> <td>0</td> <td>9600</td> <td>2</td> <td>38400</td> </tr> <tr> <td>8</td> <td>USB port on the side</td> <td></td> <td></td> </tr> </table>	Value	Human machine interface	Value	Human machine interface	0	9600	2	38400	8	USB port on the side												
Value	Human machine interface	Value	Human machine interface																			
0	9600	2	38400																			
8	USB port on the side																					
Bit 2 (Baud Rates) H00 ⁰ <table border="1"> <tr> <th>Value</th> <th>RS232 data transfer per second</th> <th>Value</th> <th>RS232 data transfer per second</th> </tr> <tr> <td>0</td> <td>9600</td> <td>2</td> <td>38400</td> </tr> <tr> <td>8</td> <td>9600</td> <td></td> <td></td> </tr> </table>	Value	RS232 data transfer per second	Value	RS232 data transfer per second	0	9600	2	38400	8	9600			A									
Value	RS232 data transfer per second	Value	RS232 data transfer per second																			
0	9600	2	38400																			
8	9600																					
Bit 3 (Echo) H000 ⁰ <table border="1"> <tr> <th>Value</th> <th>Echo method</th> <th>Value</th> <th>Echo method</th> </tr> <tr> <td>0</td> <td>Turn on echo function in terminal</td> <td>1</td> <td>Turn off echo function in terminal</td> </tr> </table>	Value	Echo method	Value	Echo method	0	Turn on echo function in terminal	1	Turn off echo function in terminal														
Value	Echo method	Value	Echo method																			
0	Turn on echo function in terminal	1	Turn off echo function in terminal																			

PN4	H0001	IN1 can be used for SVOFF function only If it not occupied by PN3 for origin setting. If IN1 has been occupied, please change to IN2	Set the SERVO OFF and EMC input signal	A								
			Bit 0 H0001 <table border="1"> <thead> <tr> <th>Value</th> <th>Enable SVOFF and input property</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>The SVOFF contact is invalid</td> </tr> <tr> <td>1</td> <td>Enable the input of SERVO OFF (IN1_A contact), if IN1 wasn't used as HORG (IN2_A contact), when IN1 was used as HORG.</td> </tr> <tr> <td>3</td> <td>Enable the input of SERVO ON (IN1_B contact), if IN1 wasn't used as HORG (IN2_B contact), when IN1 was used as HORG</td> </tr> </tbody> </table>		Value	Enable SVOFF and input property	0	The SVOFF contact is invalid	1	Enable the input of SERVO OFF (IN1_A contact), if IN1 wasn't used as HORG (IN2_A contact), when IN1 was used as HORG.	3	Enable the input of SERVO ON (IN1_B contact), if IN1 wasn't used as HORG (IN2_B contact), when IN1 was used as HORG
			Value		Enable SVOFF and input property							
			0		The SVOFF contact is invalid							
1	Enable the input of SERVO OFF (IN1_A contact), if IN1 wasn't used as HORG (IN2_A contact), when IN1 was used as HORG.											
3	Enable the input of SERVO ON (IN1_B contact), if IN1 wasn't used as HORG (IN2_B contact), when IN1 was used as HORG											
Bit 1 H0001 <table border="1"> <thead> <tr> <th>Value</th> <th>Braking method of SRVOFF</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>When SVOFF, the motor will decelerate. After the motor stop, the current will be cut off (102/03/20modified)</td> </tr> <tr> <td>1</td> <td>When SVOFF, the current will be shut down immediately and the motor runs freely (102/03/20modified)</td> </tr> </tbody> </table>	Value	Braking method of SRVOFF	0	When SVOFF, the motor will decelerate. After the motor stop, the current will be cut off (102/03/20modified)	1	When SVOFF, the current will be shut down immediately and the motor runs freely (102/03/20modified)						
Value	Braking method of SRVOFF											
0	When SVOFF, the motor will decelerate. After the motor stop, the current will be cut off (102/03/20modified)											
1	When SVOFF, the current will be shut down immediately and the motor runs freely (102/03/20modified)											
Bit 2 Reserved by Manufacturer Bit 3 Reserved by Manufacturer												
PN5	H0000	Assume IN2 are simultaneously occupied by SVOFF and CWHC, SVOFF will be applied prior to CWHC.	Set the CWHC/CCWHC input	5								
			Bit 0 H0101 <table border="1"> <thead> <tr> <th>Value</th> <th>Enable CWHC and input property</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>The CWHC contact is invalid</td> </tr> <tr> <td>1</td> <td>When the contact is valid, CWHC will be enabled (IN2_A contact)(Not used in SVOFF function)</td> </tr> <tr> <td>3</td> <td>When the contact is valid, CWHC will be enabled (IN2_B contact) (Not used in SVOFF function)</td> </tr> </tbody> </table>		Value	Enable CWHC and input property	0	The CWHC contact is invalid	1	When the contact is valid, CWHC will be enabled (IN2_A contact)(Not used in SVOFF function)	3	When the contact is valid, CWHC will be enabled (IN2_B contact) (Not used in SVOFF function)
			Value		Enable CWHC and input property							
			0		The CWHC contact is invalid							
			1		When the contact is valid, CWHC will be enabled (IN2_A contact)(Not used in SVOFF function)							
3	When the contact is valid, CWHC will be enabled (IN2_B contact) (Not used in SVOFF function)											
Bit 1 H0101 <table border="1"> <thead> <tr> <th>Value</th> <th>Brake method of CWHC</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>When CWHC, the motor will decelerate according to parameter VA.</td> </tr> <tr> <td>1</td> <td>When CWHC, the motor will decelerate. After the motor stop, the output current will be shut down.</td> </tr> </tbody> </table>	Value	Brake method of CWHC	0	When CWHC, the motor will decelerate according to parameter VA.	1	When CWHC, the motor will decelerate. After the motor stop, the output current will be shut down.						
Value	Brake method of CWHC											
0	When CWHC, the motor will decelerate according to parameter VA.											
1	When CWHC, the motor will decelerate. After the motor stop, the output current will be shut down.											
Bit 2 H0101 <table border="1"> <thead> <tr> <th>Value</th> <th>Enable CCWHC and input property</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>The CCWHC is invalid</td> </tr> <tr> <td>1</td> <td>When the contact is valid, enable CCWHC(IN3_A contact)</td> </tr> <tr> <td>3</td> <td>When the contact is valid, enable CCWHC(IN3_B contact)</td> </tr> </tbody> </table>	Value	Enable CCWHC and input property	0	The CCWHC is invalid	1	When the contact is valid, enable CCWHC(IN3_A contact)	3	When the contact is valid, enable CCWHC(IN3_B contact)				
Value	Enable CCWHC and input property											
0	The CCWHC is invalid											
1	When the contact is valid, enable CCWHC(IN3_A contact)											
3	When the contact is valid, enable CCWHC(IN3_B contact)											
Bit 3 H0101 <table border="1"> <thead> <tr> <th>Value</th> <th>Brake method of CCWHC</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>When CCWHC, the motor will decelerate according to parameter VA.</td> </tr> <tr> <td>1</td> <td>When CCWHC, the motor will decelerate. After the motor stop, the output current will be shut down.</td> </tr> </tbody> </table>	Value	Brake method of CCWHC	0	When CCWHC, the motor will decelerate according to parameter VA.	1	When CCWHC, the motor will decelerate. After the motor stop, the output current will be shut down.						
Value	Brake method of CCWHC											
0	When CCWHC, the motor will decelerate according to parameter VA.											
1	When CCWHC, the motor will decelerate. After the motor stop, the output current will be shut down.											
PN6	H0001	When PN6=H0001 and PN7=H0100 OUT1=INP OUT2=BRK PN7=H0010 OUT1=BRK OUT2=ALM PN7=H0001 OUT1=BRK OUT2=RDY	The BREAK setting of brake output(BRK)	A								
			Bit 0 H0001 (Attention : Only BREAK or READY can be enabled) <table border="1"> <thead> <tr> <th>Value</th> <th>Enable BREAK (BRK) and signal property</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>The BREAK signal output is invalid</td> </tr> <tr> <td>1</td> <td>After the servo is ready, the brake output will become OFF(Form A contact)</td> </tr> <tr> <td>3</td> <td>After the servo is ready, the brake output will become ON(Form B contact)</td> </tr> </tbody> </table>		Value	Enable BREAK (BRK) and signal property	0	The BREAK signal output is invalid	1	After the servo is ready, the brake output will become OFF(Form A contact)	3	After the servo is ready, the brake output will become ON(Form B contact)
			Value	Enable BREAK (BRK) and signal property								
			0	The BREAK signal output is invalid								
1	After the servo is ready, the brake output will become OFF(Form A contact)											
3	After the servo is ready, the brake output will become ON(Form B contact)											
Bit 1 H0001 <table border="1"> <thead> <tr> <th>Value</th> <th>Time delay of BREAK On</th> </tr> </thead> <tbody> <tr> <td>0~F</td> <td>The time delay before the brake is on (Unit : 100ms)</td> </tr> </tbody> </table>	Value	Time delay of BREAK On	0~F	The time delay before the brake is on (Unit : 100ms)	A							
Value	Time delay of BREAK On											
0~F	The time delay before the brake is on (Unit : 100ms)											
Bit 2 H0001 <table border="1"> <thead> <tr> <th>Value</th> <th>Time delay of BREAK Off</th> </tr> </thead> <tbody> <tr> <td>0~F</td> <td>The time delay before the brake is off (Unit : 100ms)</td> </tr> </tbody> </table>	Value	Time delay of BREAK Off	0~F	The time delay before the brake is off (Unit : 100ms)								
Value	Time delay of BREAK Off											
0~F	The time delay before the brake is off (Unit : 100ms)											
			Bit 3 Reserved by Manufacturer									

PN7	H0110	When PN6=H0000 and PN7=H0110 ,then OUT1=INP OUT2=ALM PN7=H0101 OUT1=INP OUT2=RDY PN7=H0011 OUT1=RDY OUT2=ALM	Signal output setting		A
			Bit 0 H0000		
			Value	Enable READY(RDY) and signal property	
			0	The READY signal output is invalid	
1	When the servo is ready, the transistor will become ON				
3	When the servo is ready, the transistor will become OFF				
Bit 1 H0000		Value	Enable ALARM(ALM) and signal property	0	The ALARM signal output is invalid
		1	When the drive malfunctions, the transistor will become ON	3	When the drive malfunctions, the transistor will become OFF
Bit 2 H0000		value	Enable POSOK(INP) and signal property	0	The in position signal is invalid
		1	When POSOK is enabled, the transistor will become ON	3	When POSOK is enabled, the transistor will become OFF
Bit 3 Reserved by Manufacturer					A
PN10 VM	900	1~3000	Set the speed of motor (unit : rpm)		
			1. When MD=5, it's the speed of MA command		5
			2. When MD=1, it's the maximum speed of motion command		1
PN11 VA	100	1~2000	Set the acceleration (Unit : rps ²)		1,5
PN12 PSC1	1	1~9999	Set the ratio of pulse multiplier (electronic gear ratio)		0,1
PN13 PSC2	1	1~9999	Set the ratio of pulse divider (electronic gear ratio)		0,1
PN14	900	0~5000	JOG speed setting		
PN18 EP	0	1~999	Set the tolerance for In Position of motor : The range setting of POSOK.		0,1
PN19 VF	3000	1~4000	Set the baseline to calculate maximum speed of motor (unit:rpm)		A
PN22 VB	15		The speed of departing the origin immediately after it return.		
PN23 VH	300	1~1200	The speed while return to origin.		5
PN24	0	0			5
PN25 EL	400	20~8000	Set the limit for position tolerance		0,1,5
			When the position tolerance of motor exceeds the value of EL, Err-04 will appear ※This parameter will differ from the model of motor and encoder.		
PN29		DLY	The delay time when terminal response		
PN30	KP				
PN31	KD				
PN32	KI				
PN33	DM				
PN37	TL				
PN44	1	MSC1	The electric gear ratio in control mode		
PN45	1	MSC2			
PN47	H0000		Bit 0 - 1 H0000		
			Value	STN(0~255) MODBUS communication setting	
			Bit 2 H0100		
			Value	MODBUS synchronization operation setting	
		0	MODBUS function is invalid		
		2	Connect MODBUS : RTU984(SLAVE-RS485 port)		
		3	ASCII(SLAVE-RS232 port)		
PN66	20	IMN			
PN76	90	IMX			

※ This instruction would be modified at indefinite time for error correction, new function etc. If you have any question regarding this product, please come to our website(<http://www.mindman.com.tw/>) to download the newest version or call us.

2. Commands Table (Editable by Computer)

All symbol of parameters used in the tables are listed and explained as below

f32 : 32bit real number or floating number

d16 : 16bit integer , $-32768 \leq d16 \leq 32767$

n16 : 16bit integer , $0 \leq n16 \leq 255$ ~ ~ ~ ~ ~

n8 : 8bit positive integer , $0 \leq n8 \leq 255$

Rn : Real variable R0 ~ R63

Nn : Integer variable , N0 ~ N63

Pn : Output port , P0 ~ P2 (The number of I/O ports depends on the motor model)

In : Input port , I0 ~ I5 (The number of I/O ports depends on the motor model)

Ln : Line number of program , \$lb(Label) can be used instead.

EVn : Event code , EV0 ~ EV3

The commands are classified in three applicable types, program, immediateness and external.

Program: The commands are wrote and executed in the program.

Immediateness: The commands are executed immediately while the system went on standby. The commands can be input as string using RS232(like terminal , VB).

External: The commands can be executed not only when system went on standby but also when other program is executing.

2-1) Motion control statements

Command	Function description	Program	Instantly	External
MA f32	Move Absolutely Set the destination using absolute coordinate and rotate the motor. When execute this command, the velocity of motor is determined by PN10(VM) and the acceleration is determined by PN11 (VA).	<input type="radio"/>	<input type="radio"/>	
MA Rn	Same as above, but set the coordinate parameter using Rn	<input type="radio"/>		
MR f32	Move Relatively Set the distance using relative coordinate and rotate the motor. When execute this command, the velocity of motor is determined by PN10(VM) and the acceleration is determined by PN11 (VA).	<input type="radio"/>	<input type="radio"/>	
MR Rn	Same as above, but set the distance parameter using Rn instead	<input type="radio"/>		
JGF	Jog Forward -- Start the motor and rotate CW continuously When execute this command, the motor will rotate immediately and continuously to positive direction on axis until receiving the JG0 command. The velocity of motor is determined by PN14(VJ), the acceleration is determined by PN11 (VA) and the positive direction is determined by PN02.	<input type="radio"/>	<input type="radio"/>	
JGR	Jog Reverse -- Start the motor and rotate CCW continuously When execute this command, the motor will rotate immediately and continuously to negative direction on axis until receiving the JG0 command. The velocity of motor is determined by PN14(VJ), the acceleration is determined by PN11 (VA) and the negative direction is determined by PN02.	<input type="radio"/>	<input type="radio"/>	
JG0	Jog Stop Stop the JGF or JGR command.	<input type="radio"/>	<input type="radio"/>	

2-2) Homing and Coordinate Configuration Statements

Command	Function description	Program	Instantly	External
H	Homing -- Return to original position When execute this command, the motor will start to search home using speed setting PN23(VH) and direction setting PN2 bit 3~0(HM). When the signal of origin is triggered, the motor will decelerate immediately. While it need some decelerating time, the position would possibly exceed the home. For accurate positioning, the motor will slowly return to the position where the signal is triggered using speed setting PN22(VB) . The final position is machine reference point	<input type="radio"/>	<input type="radio"/>	
CS f32	Coordinate Set – Set the current coordinate Set the coordinate of current position as f32 The f32 is floating number. It can be a positive or negative number with or without decimal (About the settings of axis coordinate, please see the descriptions of parameter PN44(MSC1) and PN45(MSC2))	<input type="radio"/>	<input type="radio"/>	
CS Rn	Same as above, but set the coordinate using Rn instead	<input type="radio"/>		

2-3) Servo Control Statements

Command	Function description	Program	Instantly	External
HOFF	Hold Off -- Turn off the servo control of motor Generally, the motor is controlled by the system when power on. Even when the system went on standby, the motor can't be moved by external force. If the users wanted to push the motor directly, they must execute this command to turn off the servo control of motor. This command is used in manual teaching mode generally, the users can move the motor by hand and set the coordinate of destination. While in the system with encoder, the current coordinate of motor from can be read by encoder even when the servo control is off. Using this command in teaching mode, the users can set the movement path of motor by hand.	<input type="radio"/>	<input type="radio"/>	
HON	Hold On – Restore the servo control of motor. This command is used to restore the servo control turned off by HOFF command.	<input type="radio"/>	<input type="radio"/>	

STOP	<p>Stop</p> <p>The motor will decelerate and stop immediately. The deceleration is determined by parameter PN11(VA).</p>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
EPI	<p>Enable Pulse input</p> <p>Enable the function of pulse input and let the motor to be controlled by input pulse like pulse input control mode (MD=1).</p> <p>While the pulse input function is enabled, the movement control command of motor(MA/MR/JGF/JGR etc) will be disabled until DPI command is executed. You can choose the pulse type of encoder using PN02.</p> <p>Warning : It shouldn't be any program existed after EPI was executed.</p>			
DPI	<p>Disable Pulse input</p> <p>Turn off the pulse input function as above and restore the movement control function.</p>	<input type="radio"/>	<input type="radio"/>	
FT d16	<p>Fit Torque</p> <p>Turn the motor into torque mode. The torque and direction are determined by d16(positive/negative value).</p>	<input type="radio"/>	<input type="radio"/>	
FT Nn	<p>Same as above, but use parameter Nn instead of d16.</p>	<input type="radio"/>		
TM=n16	<p>Set the timer</p> <p>The system timer TM is a 16bit timer. The timer starts counting while powered on. The counter plus 1 every 0.1 second and circulates from 0 to 65535. The command can give a new initial value. When a new value is given, the timer will plus 1 every 0.1s to new value. You can use Nn=TM to read the system timer whenever necessary. Also you can use SET Evn, TM, Nn as interrupt to trigger the signal source.</p>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
TM=Nn	<p>Same as above, but use parameter Nn instead of n16.</p>	<input type="radio"/>		
TM2=n16	<p>Same as TM=n16 command.</p>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
TM2=Nn	<p>Same as TM=Nn command</p>	<input type="radio"/>		
RESET	<p>System reset</p> <p>When the system error occurs, the motor will reset the servo control and clear the error. Restart the servo control of motor.</p> <p>Before execute this command, please make sure all the reason of errors has been cleared.</p>	<input type="radio"/>	<input type="radio"/>	
RESET H	<p>Soft reset</p> <p>This command can emulate the switch of external power supply to restart the system.</p>	<input type="radio"/>	<input type="radio"/>	

RX	Clear the maximum value of records including maximum current(IMX), maximum load(LDMX), max voltage (ACMX), minimum voltage (ACMN), maximum temperature (TPMX) etc.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
NOP	No operation performed, just occupy one line. Normally, it is used to adjust the line number of program If you use \$Lb(Label) instead of line number while programming, then you don't need this command.	<input type="radio"/>		
DN	Done Waiting the current movement to finish and then execute another commands. Warning: when the servo is off (HOFF), this command shouldn't be executed after JGF/JGR command. Or the system will be in waiting status forever(It can only be cancelled using interrupt)	<input type="radio"/>		

2-4) Terminal Output Statements

Command	Function description	Program	Instantly	External
OUT Nn	Output the value of parameter Nn through RS232.	<input type="radio"/>		
OUT Rn	Output the value of parameter Rn through RS232.	<input type="radio"/>		
OUT "string"	Output the string through RS232 You can combine multiple OUT command into one line using comma Eg. : Executes OUT "N2=",N2 commands If N2=123 , then string N2=123 will be output through RS232.	<input type="radio"/>		

2-5) Output Port Control Statements

Command	Function description	Program	Instantly	External
SET Pn	Enable output port Pn (Enable) It means to turn on the transistor on output port. Some output ports have multiple or default functions. Before you use this command, please turn off those functions. Or it will cause errors (Please see the description of system parameters PN06 ~ PN07)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
CLR Pn	Disable output port Pn (Disable) It means to turn off the transistor on output port.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

OUTP n16	<p>Set all commands of output port simultaneously</p> <p>Using the Bit 0 of parameter n16 to set output port P0, bit 1 to set output port P1, bit 2 to set output port P2 ...if the value of bit is 1, then turn on the relative output port. If 0, then turn off the relative output port</p> <p>For example: OUT0,1,2. It must use binary to calculate 1+2+4=7 , then the comman should be OUTP 7.</p>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
OUTP Nn	Same as above but use parameter Nn instead of d16	<input type="radio"/>		

2-6) Variable Statements

Command	Function description	Program	Instantly	External
Nn=d16 Nn=n16	<p>Set Nn=d16</p> <p>The parameter can be a integer from -32768 to 32767. It can also be positive integer from 0 to 65536. Or hexadecimal from H0000 to HFFFF. If you use ?Nn to inquiry, all return will be integer. The value exceeds 32768 will be displayed in negative integer.</p>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Nn=Nm	Set Nn=Nm	<input type="radio"/>		
Nn=Rm	<p>Set Nn=Rm</p> <p>The variable N is a 16bit integer, R is a 32bit real number. Therefor, it has some limit when you convert. The variable R must be between -32768 and 32767 and all decimals will be discard.</p>	<input type="radio"/>		
Nn=ERR	<p>ERR=Error</p> <p>This command can read the system ERR into variable Nn , for determining in program flow control</p> <p>Please see appendix D_ERR for more information.</p>	<input type="radio"/>		
Nn=AC	<p>Read the voltage of power supply. The unit is Volt.</p> <p>Please notice AC represents the voltage of current, it doesn't mean you can use alternating current as power supply for the motor.</p>	<input type="radio"/>		
Nn=TM	<p>TM=Timer</p> <p>This command can read the value of system timer into variable Nn.</p>	<input type="radio"/>		
Nn=INP	<p>INP=Input Port The status of input port</p> <p>INP is a 16bit integer, each bit represents a port status. Bit 0 represent port IN0, bit 1 represent port IN1, Bit 2 represent port IN2 ...</p>	<input type="radio"/>		
Nn=TEMP	TEMP=Temperature The current temperature of system	<input type="radio"/>		

Nn=RPM	Read the rpm of motor	<input type="radio"/>		
Nn=IMX	Read the maximum current	<input type="radio"/>		
Nn=LDMX	Read the maximum load	<input type="radio"/>		
Nn=ACMX	Read the maximum input voltage	<input type="radio"/>		
Nn=ACMN	Rea the minimum input voltage	<input type="radio"/>		
n=TPMX	Read the maximum temperature	<input type="radio"/>		
Nn=PNm	Read variable PNm into variable Nn	<input type="radio"/>		
Nn=NS(d16)	Read array variable NS(d16)(saved in EEPROM) into variable Nn	<input type="radio"/>		
Nn=NS(Nm)	Same as above but use parameter Nn instead of NS(d16)	<input type="radio"/>		
Nn=TpST	Read parameter ST of Tp motor	<input type="radio"/>		
Nn=TpERR	Read parameter ERR of Tp motor	<input type="radio"/>		
Nn+=d16	$Nn=Nn+d16$	<input type="radio"/>		
Nn+=Nm	$Nn=Nn+Nm$	<input type="radio"/>		
Nn-=d16	$Nn=Nn-d16$	<input type="radio"/>		
Nn-=Nm	$Nn=Nn-Nm$	<input type="radio"/>		
Nn*=d16	Nn= multiply Nn by d16	<input type="radio"/>		
Nn*=Nm	Nn= multiply Nn by Nm	<input type="radio"/>		
Nn/=d16	Nn= divide Nn by d16	<input type="radio"/>		
Nn/=Nm	Nn=divide Nn by Nm	<input type="radio"/>		
Nn%=d16	Nn= The remainder in the division of Nn by d16	<input type="radio"/>		
Nn%=Nm	Nn= The remainder in the division of Nn by Nm	<input type="radio"/>		
Nn&=d16	Nn= Logical conjunction(And) of Nn and d16	<input type="radio"/>		
Nn&=Nm	Nn= Logical conjunction(And) of Nn and Nm	<input type="radio"/>		

$Nn \mid = d16$	$Nn =$ Logical disjunction(Or) of Nn and $d16$	<input type="radio"/>		
$Nn \mid = Nm$	$Nn =$ Logical disjunction(Or) of Nn and Nm	<input type="radio"/>		
$Nn \wedge = d16$	$Nn =$ Exclusive disjunction(Exclusive OR) of Nn and $d16$	<input type="radio"/>		
$Nn \wedge = Nm$	$Nn =$ Exclusive disjunction(Exclusive OR) of Nn and Nm	<input type="radio"/>		
$Rn = \text{SIN}(Rm)$	Sine function in mathematics	<input type="radio"/>		
$Rn = \text{COS}(Rm)$	Cosine function in mathematics	<input type="radio"/>		
$Rn = \text{TAN}(Rm)$	Tangent function in mathematics	<input type="radio"/>		
$Rn = \text{ATN}(Rm)$	Arctangent function in mathematics	<input type="radio"/>		
$Rn = \text{SQRT}(Rm)$	Square function in mathematics	<input type="radio"/>		
$Rn += f32$	$Rn = Rn + f32$	<input type="radio"/>		
$Rn += Rm$	$Rn = Rn + Rm$	<input type="radio"/>		
$Rn -= f32$	$Rn = Rn - f32$	<input type="radio"/>		
$Rn -= Rm$	$Rn = Rn - Rm$	<input type="radio"/>		
$Rn *= f32$	$Rn = Rn * f32$	<input type="radio"/>		
$Rn *= Rm$	$Rn = Rn * Rm$	<input type="radio"/>		
$Rn /= f32$	$Rn = Rn / f32$	<input type="radio"/>		
$Rn /= Rm$	$Rn = Rn / Rm$	<input type="radio"/>		

2-7) Program Flow Control Statements

Command	Function	Program	Instantly	External
G	Start the program		<input type="radio"/>	
G \$lb	Start the subroutine in \$lb until the RET command finished		<input type="radio"/>	
SG	Execute the program step by step. Only execute one line every time the enter key was hit. It is usually used when debugging in the early stage of program development		<input type="radio"/>	
QUIT	Quit the current executed program immediately and turn off the servo control. The motor will decelerate and then stop.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
EXIT	Quit the current executed program and go back to the terminal control mode (The executed command will continue until the end of program).	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
JP \$lb	Jump to \$lb and continue the program	<input type="radio"/>		
CALL \$lb	Call the subroutine Jump to \$lb and execute the subroutine. Return back to original line until the RET command is executed and continue the next line.	<input type="radio"/>		
RET	Quit the current executed subroutine. Return to the line next to where you call the subroutine and continue. If no subroutine is called, quit program mode and go back to terminal mode.	<input type="radio"/>		
JI li,\$lb	Check input port li. If it's enabled, jump to \$lb and continue.If not, go to next line	<input type="radio"/>		
JNI li,\$lb	Check input port li. If it is disabled, jump to \$lb and continue.If not, go to next line	<input type="radio"/>		
JZ Nn,\$lb	Check variable Nn. If it equal 0, jump to \$lb and continue.If not, go to next line.	<input type="radio"/>		
JZ Rn,\$lb	Check variable Rn. If it equal 0, jump to \$lb and continue.If not, go to next line.	<input type="radio"/>		
JNZ Nn,\$lb	Check variable Nn. If it doesn't equal 0, jump to \$lb and continue.If it is, go to next line.	<input type="radio"/>		
JNZ Rn,\$lb	Check variable Rn. If it doesn't equal 0, jump to \$lb and continue.If it is, go to next line.	<input type="radio"/>		

DJNZ Nn,\$Ib	<p>Subtract 1 from Nn. If the result doesn't equal 0, jump to \$Ib and continue.If it is, go to next line.</p> <p>This command can be used in a loop.Generally, \$Ib should be used before this command. Nn means how many iterations in a loop. All codes between \$Ib and this command will be executed Nm times repeatedly.</p>	<input type="radio"/>		
JTI li,t16,\$Ib	<p>This command is similar to JI li,\$Ib , but add a parameter for waiting time.</p> <p>It will check the input port li repeatedly. If li is enabled, jump to \$Ib and continue until the time limit(t16) was exceeded.</p>	<input type="radio"/>		
JTNI li,t16,\$Ib	<p>This command is similar to above command, but check opposite property.</p> <p>It will check the input port li repeatedly. If li is disabled, jump to \$Ib and continue until the time limit(t16) was exceeded.</p>	<input type="radio"/>		
JE Nn,Nm,\$Ib	Check variable Nn and Nm. If it's equal, jump to \$Ib and continue.If not, go to next line.	<input type="radio"/>		
JE Nn,d16,\$Ib	Check variable Nn. If it equal d16, jump to \$Ib and continue.If not, go to next line.	<input type="radio"/>		
JE Rn,Rm,\$Ib	Check variable Rn and Rm. If it's equal, jump to \$Ib and continue.If not, go to next line.	<input type="radio"/>		
JE Rn,f32,\$Ib	Check variable Rn. If it equal f32, jump to \$Ib and continue.If not, go to next line.	<input type="radio"/>		
JNE Nn,Nm,\$Ib	Check variable Nn and Nm. If it doesn't equal, jump to \$Ib and continue.If not, go to next line.	<input type="radio"/>		
JNE Nn,d16,\$Ib	Check variable Nn. If it doesn't equal d16, jump to \$Ib and continue.If not, go to next line.	<input type="radio"/>		
JNE Rn,Rm,\$Ib	Check variable Rn and Rm. If it isn't equal, jump to \$Ib and continue.If not, go to next line.	<input type="radio"/>		
JNE Rn,f32,\$Ib	Check variable Rn. If it doesn't equal f32, jump to \$Ib and continue.If not, go to next line.	<input type="radio"/>		
JG Nn,Nm,\$Ib	Check variable Nn and Nm. If Nn is greater than Nm, jump to \$Ib and continue.If not, go to next line.	<input type="radio"/>		
JG Nn,d16,\$Ib	Check variable Nn. If Nn is greater than d16, jump to \$Ib and continue.If not, go to next line.	<input type="radio"/>		

JG Rn,Rm,\$lb	Check variable Rn and Rm. If Rn is greater than Rm, jump to \$lb and continue.If not, go to next line.	<input type="radio"/>		
JG Rn,f32,\$lb	Check variable Rn. If Rn is greater than f32, jump to \$lb and continue.If not, go to next line.	<input type="radio"/>		
JNG Nn,Nm,\$lb	Check variable Nn and Nm. If Nn isn't greater than Nm, jump to \$lb and continue.If not, go to next line.	<input type="radio"/>		
JNG Nn,d16,\$lb	Check variable Nn. If Nn isn't greater than d16, jump to \$lb and continue.If not, go to next line.	<input type="radio"/>		
JNG Rn,Rm,\$lb	Check variable Rn and Rm. If Rn isn't greater than Nm, jump to \$lb and continue.If not, go to next line.	<input type="radio"/>		
JNG Rn,f32,\$lb	Check variable Rn. If Rn isn't greater than f32, jump to \$lb and continue.If not, go to next line.	<input type="radio"/>		
WT t16	Wait, Let the program paused for a period of time(parameter t) and then continue. The unit of t is 0.01 second.	<input type="radio"/>		
WT Nn	Same as above, but use variable N instead of t.	<input type="radio"/>		
WT li	Wait Input -- Let the program paused until input port li is enabled.	<input type="radio"/>		
WT Nli	Wait No Input -- Let the program paused until input port li is disabled.	<input type="radio"/>		

2-8) Inquiry Statements

Command	Function description	Program	Instantly	External
?AC	Inquiry the voltage of power supply. The unit is Volt.		<input type="radio"/>	<input type="radio"/>
?ERR	Inquiry parameter ERR. Please see appendix D_ERR for more information.		<input type="radio"/>	<input type="radio"/>
?IC	Inquiry current and maximum current of motor. The unit is 0.01 Amp.		<input type="radio"/>	<input type="radio"/>
?Nn	Inquiry current value of variable N.		<input type="radio"/>	<input type="radio"/>
?NS(n8)	Inquiry current value of variable NS.		<input type="radio"/>	<input type="radio"/>
?PC	Inquiry current command position of motor. It will return the coordinate in floating number.		<input type="radio"/>	<input type="radio"/>
?PE	Inquiry real position of motor(Encoder Position). It will return the coordinate in floating number.		<input type="radio"/>	<input type="radio"/>
?RPM	Inquiry current rpm of motor.		<input type="radio"/>	<input type="radio"/>
?RS(n7)	Inquiry current value of variable RS.		<input type="radio"/>	<input type="radio"/>
?TEMP	Inquiry current temperature of motor.		<input type="radio"/>	<input type="radio"/>
?TM	Inquiry current value of timer.		<input type="radio"/>	<input type="radio"/>
?VER	Inquiry the version of drive.		<input type="radio"/>	<input type="radio"/>
?MT	Inquiry the model of motor.		<input type="radio"/>	<input type="radio"/>

2-9) Other Statements

Command	Function description	Program	Instantly	External
PG	Program Generate – Open a new program. If there was a program existed, it will prompt a message to ask you deleting the old one.		<input type="radio"/>	
PA	Program Append – Append the new command to the end of existed program.		<input type="radio"/>	
PI n	Insert the new command before nth line.		<input type="radio"/>	
PD n	Delete the command of nth line		<input type="radio"/>	
PE n	Edit the command of nth line		<input type="radio"/>	
ULP	Upload Program – Upload existed program from motor system. The program can be backup to text file in computer with capture text function of terminal.		<input type="radio"/>	
DLP	Download Program – Download new program to motor system. If there was a program existed, it will prompt a message to ask you deleting the old one. If you want to keep it in EEPROM, use SAVE P command.		<input type="radio"/>	
ULC	Upload Config -- Upload current parameter from motor system. The parameters can be backup to text file in computer with capture text function of terminal.		<input type="radio"/>	
PSID	Encrypt the program (Program Secure ID)		<input type="radio"/>	
DF	Default – Reset all parameter to system default. Because some default value of parameters are relative to PN01(MD). If you modify parameter PN01, it's recommended to execute this command and SAVE C.		<input type="radio"/>	
SAVE C	Save current parameter to system		<input type="radio"/>	
SAVE P	Save current program to system		<input type="radio"/>	
SAVE Dn	Save current variable NS,RS to area Dn in EEPROM(0 <= n <= 3)		<input type="radio"/>	
LOAD Dn	Restore values from area Dn in EEPROM to variable NS,RS (0 <= n <= 3)		<input type="radio"/>	

3. ERR error code reference

The table of exception alarm

ERR Bit	ERR2 Bit	Exception description	Possible Reason	Recommended process
01		Over voltage	The voltage is unstable. Generally, happen in the night.	Install a voltage stabilizer
02		Over load	The load has exceeded Wrong parameter	Reduce the load or speed
03	01	Brief overcurrent(IL1*1.2)	Accelerated too fast Wrong parameter	Decrease the acceleration VA(PN11) and speed VM(PN10) VJ(PN14)
	02	Instantaneous overcurrent (IL1)		
	03	integral overcurrent (IL2)	Overloaded	
04		Following exception	Overloaded Accelerated too fast EL setting is too small Wrong parameter	Decrease the acceleration VA(PN11) and speed VM(PN10) VJ(PN14)
05	04	Encoder UVW exception	Wrong wiring or interference	
	05	Encoder Z phase exception	The connector wasn't fixed or well contacted	
06		Low voltage	The voltage is unstable.	1.When happend during power on, please check the input voltage and wiring. If it's right, please send it back to manufacturer. 2.If happened frequently while running, please use power supply with higher voltage
07		ltrip	Wrong wiring MT error setting Wrong parameter	Check the order of red, white, black, green wire Check encoder type
08		regenerative voltage exception	Overloaded The braking circuit of system malfunctioned	Check if the power led will be immediately off when shutdown
09		Over pulse rate	The pulse frequency is too high	Adjust SC1 and SC2
10		Over speed	VFsetting is too low	
11		Hard clip or EMC	Wrong wiring or the polar setting of sensor is wrong	
12		EERom write error		
13		Higt temperature	Overloaded for a long time The temperature sensor malfunctioned	Touch by hand to see if the temperature is too high
14	06	Reset error while power on	Immediately power on after shutdown Abnormal power supply	
	07	EERom error while power on		
	08	Current sensor error while power on		
15		Communication error between modules.	Wrong wiring or interference The baudrate is too high Didn't add resistances to the front and end of module	

※ Before you clear the alarm and the system retrun to normal, please confirm :

1. If no commands are input to drive from controller.
2. If all exceptions are excluded (**It's possible to have multiple alarms**), in order to avoid damage to the drive again.

4. Moter Modbus interface address

Modbus interface address table

Address	Command	Class	Read /Write	Function	Comment
00001	01H/05H	B	R/W	Rotate CW continuously	JGF/JG0
00002	01H/05H	B	R/W	Rotate CCW continuously	JGR/JG0
00003	05H	B	W1	Positioning of absolute axis coordinate. Use parameter r to set the coordinate	MA r
00004	05H	B	W1	Positioning of relative axis coordinate. Use parameter r to set the relative distance	MR r
00005	01H/05H	B	R/W	Turn off sero control	HOFF/HON
00006	05H	B	W1	Reset system	RESET
00007	05H	B	W1	Reset coordinate. Use parameter r to set the coordinate	CS r
00008	05H	B	W1	Go back to machine refenence point	H
00009	05H	B	W1	Motor emergency shutdown	STOP
00010	01H/05H	B	R/W	Motor pause	PZ/REDO
00011	05H	B	W1	Reset monitoring parameter	RX
00012	05H	B	W1	Save system parameter	SAVE
00013	05H	B	W1	Enable the trigger of origin	EN HT
00014	06H	B	W1	Disable the trigger of origin	DS HT
00033	05H	B	W1	Execute the program	G
00034	05H	B	W1	Stop the program	EXIT
00035	05H	B	W1	Enable executing the program step by step	SG
00036	05H	B	W1	Execute the program by one step	
00037	05H	B	W1	Add a line to the program	
00038	05H	B	W1	Insert a line to the program	
00039	05H	B	W1	Delete a line in the program	
00040~00055				Execute SAVE Dn command n =address-40	SAVE Dn
00060~00075				Execute LOAD Dn command n =address-60	LOAD Dn
00100~00107	05H	B	W1	Execute the command saved in 40900 ~ 40956 (8 set totally)	
00200~00205	01H/05H	B	R/W	Set the output port 0~5	The max value depends on models
30001	04H	W	R	Read RPM	
30002	04H	DW	R	Read PE	
30004	04H	DW	R	Read PC	
30006	04H	W	R	Read INP	
30009	04H	W	R	Read ERR	
30010	04H	W	R	Read OUTP	
30011	04H	W	R	Read Ready	ST&0x37FF = 0x0009
30012	04H	W	R	Read ERR2	
30013	04H	W	R	Read the trigger of origin	?HT
30014	04H	DW	R	Read the coordinate of trigger of origin	?HTP
30020	04H	W	R	Read ProgCnt	

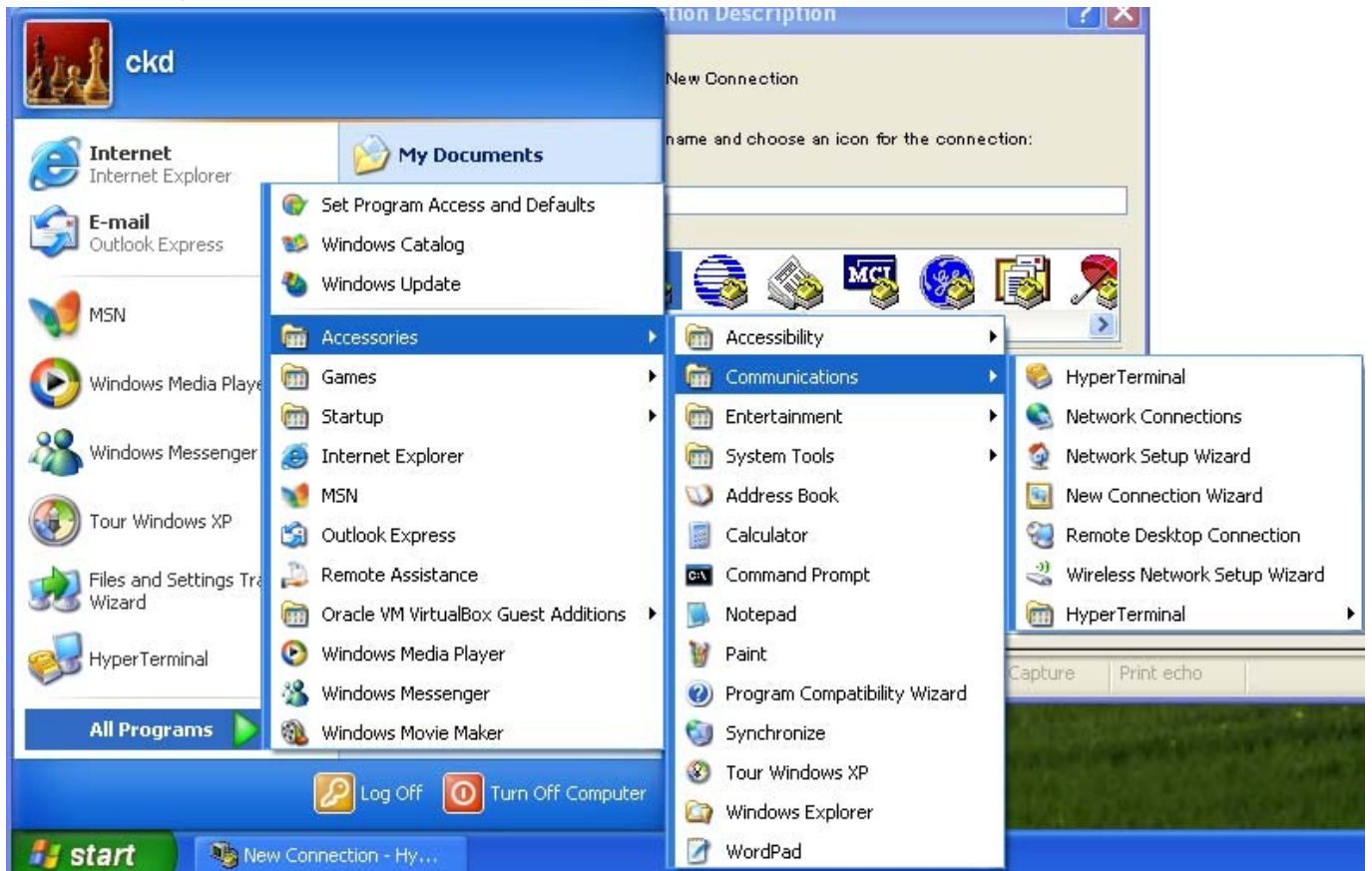
30021	04H	W	R	Read ProgLineNo	
30030	04H	W	R	Read Volt	
30031	04H	W	R	Read VoltMax	
30032	04H	W	R	Read VoltMin	
30033	04H	W	R	Read Inow	
30034	04H	W	R	Read Imax	
30035	04H	W	R	Read LDnow	
30036	04H	W	R	Read LDmax	
30037	04H	W	R	Read PosErr	
30038	04H	W	R	Read PosEmax	
30039	04H	W	R	Read Temp	
30040	04H	W	R	Read TempMax	
40001	03H/10H	F	R/W	Parameter r	
40100_40107	03H/06H/10H	W	R/W	Corresponding to MBC0~7	Command register
40200_40207	03H/06H/10H	W	R/W	Corresponding to MBS0~7	Command register
40300~40305	03H/06H/10H	W	R/W	Corresponding to the year,month,day,hour,min and second of RTC	
40400~40407	03H/06H/10H	W	R/W	Corresponding to N0~N7	
40500~40514	03H/10H	F	R/W	Corresponding to R0~R7 n = (address-40200)/2	
40700~40763	03H/06H/10H	W	R/W	Corresponding to PN0~PN63	
40800,40808, 40816,40824, 40832,40840, 40848,40856	03H/10H	String	R/W	Execute the command immediately (8 set totally)	
40900,40908, 40916,40924, 40932,40940, 40948,40956	03H/10H	String	R/W	Edit the executable string command saved in EEROM(8 set totally). Execute the commands until address 100~107 was triggered	
42000~42999	03H/06H/10H	W	R/W	Corresponding to NS(n), n =address-42000	The max value of n depends on models
43000~43998	03H/10H	F	R/W	Corresponding to RS(n), n =(address-43000)/2	Same as above
44000~43999	03H/06H/10H	W	R/W	Corresponding to ND(n), n =address-44000	Same as above
45000~45998	03H/10H	F	R/W	Corresponding to RD(n), n =(address-45000)/2	Same as above
46000	03H/06H/10H	W	R/W	The line Number of program	
46100	03H	String	R	Read Label	
46120	03H	String	R	Read code	
46200	10H	String	W	Write Label and code	
47000	03H/10H	String	R/W	Write the immediate command	
47100	03H	String	R	Read the immediate response	

5. Hyper Terminal Connection

<<Step 1>>

Open HyperTerminal located in

Start→All Programs→Accessories→Communications→HyperTerminal



<<Step 2>>

Set the connection description. You can customize the name and icon.



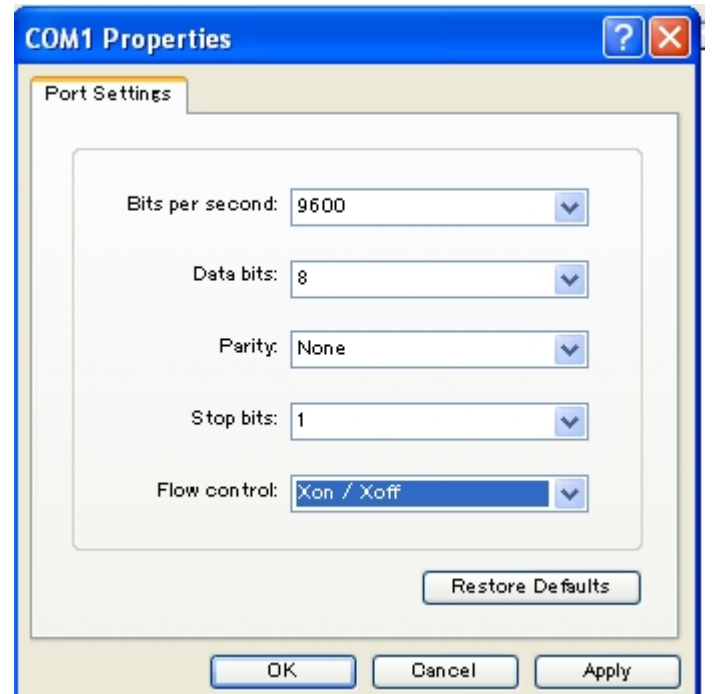
<<Step 3>>

Select the com port of RS232 in drop-down menu of Connect using. In this example, it's COM1. Please select the correct depend on your system.



<<Step 4>>

Set Bits per seconds to 9600, Parity to None, Flow control to Xon/Xoff, and others as default. After everything is done, press apply and then ok.



<<Step 5>>

Power on our motor. If connected successfully, you can see "Servo On" and "System standby" on the screen.

Servo On
System standby

6. Program Download and Update

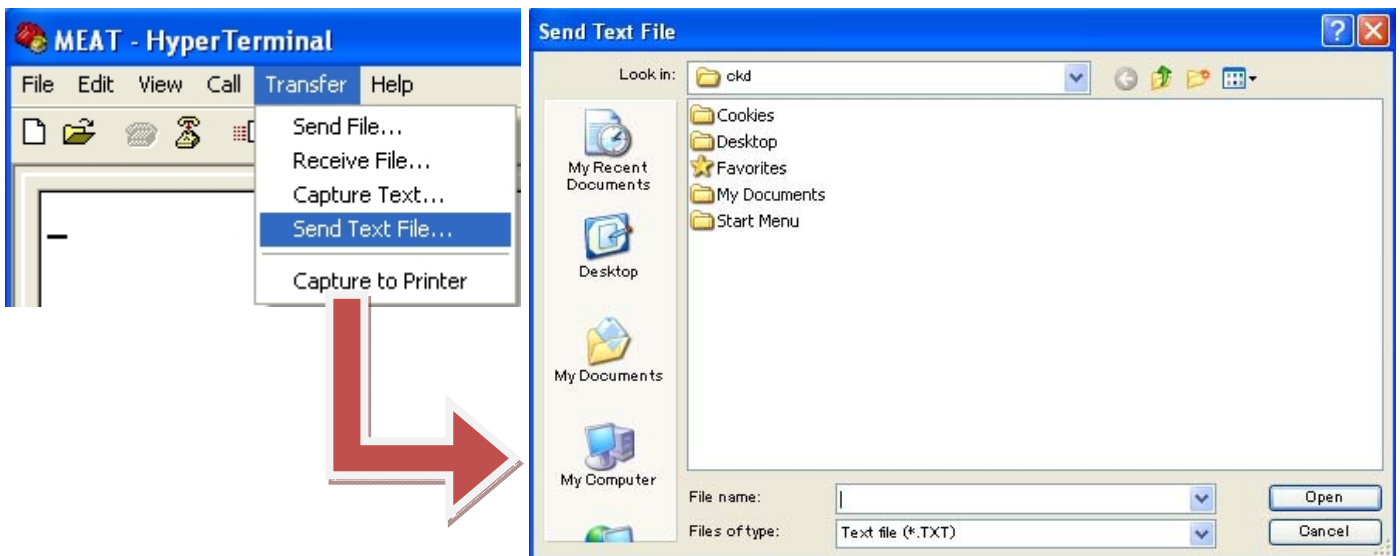
Input “EXIT” in Hyper Terminal and hit enter. Then you will see “ok” (Exit the program).

```
Data Segment 0 Loaded
Servo On
System standby
EXIT
ok
```

Input “DLP” and hit “Y” to continue.

```
DLP
This will delete all current program.
Continue? (Y/N)
```

Please download the program now.



When you see “Please download the program now” , select “Transfer → Send Text File” on the menu bar and open your program(text file).

Input “DLP” and hit enter in terminal.

Check the program(60 lines in this example).
If everything is correct, then reboot the motor.

```
L49      @166          N4&=4
L50      @170          JI I5,$STOP
L51      @174          JNZ N4,$DN1
L52      @178          SET P3
L53      @180          RET
L54      @182          $STOP: STOP
L55      @184          DN
L56      @186          SET P3
L57      @188          $STOP1: JI I5,$STOP1
L58      @192          RET
L59      @194          $ERR: CLR P3
L60      @196          SET P1
```

連線 00:04:12 ANSIW 9600 8-N-1 SCROLL CAPS NUM 插 列印