Mindman

CONTROLLER



CK10 series

User Manual

Communication Function MODBUS

Ver. 2301

Contents

1. Modbus Communication Functions 3
1.1 Communication Specifications 3
1.2 Modbus Protocol 4
1.3 Function and Data Field 5
1.4 Exception Responses 7

2. Address and Definitions of Registers 8
2.1 Status Registers 8
2.2 User IO Registers 9
2.3 Alarm Log 10
2.4 Motion Control Registers 10
2.5 Compare Out Function Register 12
2.6 Parameter Register 12
2.7 Position Table Registers 18
2.8 Drive Info Registers 20

3. Information for programming 21
3.1 Definitions of State Flags (Reg 0x3E8) 21
3.2 Definitions of Input Bits (Reg 0x3F2, 0x13DA~0x13F1) 22
3.3 Definitions of Output Bits (Reg 0x3F4, 0x13F2~0x1405) 23
3.4 Parameter Lists (Reg 0x1388 ~0x13D7) 24
3.5 Position Table Item (Reg 0x2710 ~ 0x370F) 29

1. Modbus Communication Functions

CK10 can be controlled up to 16 axis by multidrop link at RS-485 and support Plus-R and Modbus Protocols. And this manual describes communication functions when the Modbus protocol is used.

CK10 communicates using a master–slave configuration, in which only one master can initiate transactions. The slaves respond by sending the requested data to the master.

The master can address individual slaves or broadcasts message to all slaves. The slaves return a message to the individual queries except broadcast queries from the master.

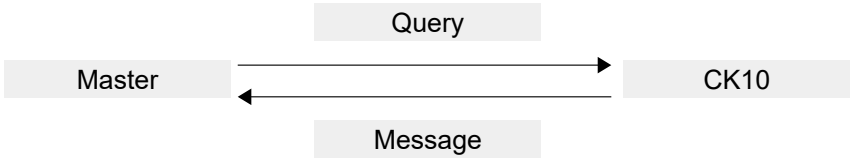
The Modbus protocol consists of the device (or broadcast) address, a function code defining the device action, data and an error–checking field. The response message of slave is also constructed using Modbus protocol. It includes a result of the requested action, data to be returned and an error–checking field. If an error occurred in the query, or if a slave is unable to execute the requested, the slave will ignore the query or respond an error message.

1.1 Communication Specifications

Specification	RS-485
Communication Type	Asynchronous
	Half-duplex
Baud Rate [bps]	9600, 19200, 38400, 57600, 115200
Data Type	8 bit
Parity	No
Stop Bit	1 bit
Error Check	Modbus ASC : LRC Modbus RTU : CRC
Max Cabling Length (Converter ⟷ Drive)	30 m
Min Cable length between drive	More than 60 cm
Number of Connected Axis	16 axis (No. 0~F)

1.2 Modbus Protocol

1. Overview of communication scheme



2. Coding system

① Modbus ASC

Hexadecimal numbers are represented by ASC II characters.

(Ex.: 0x3A → 0x33, 0x41 : Upper character is sent first)

(Ex.: 0x1234 → 0x31, 0x32, 0x33, 0x34 : Upper character(0x31) is sent first)

(Ex.: 0x12345678 → 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38 : Most Significant Character (0x12) is sent first)

② Modbus RTU

8bit binary coded

(Ex.: 0x3A → 0x3A)

(Ex.: 0x1234 → 0x12, 0x34 : Upper byte(0x12) is sent first)

(Ex.: 0x12345678 → 0x12, 0x34, 0x56, 0x78 : Most Significant Byte (0x12) is sent first)

3. Format of Message (Modbus ASC)

Start	Address	Function	Data	LRC	End
:	2 Chars	2 Chars	n Chars	2 Chars	2 Chars CRLF

① Start : " : " is the beginning if the format.

② Address : Address to select slave. The address(ID) of MECQ2-SERVO can be set address from 1 to 16. 0 is assigned for broadcast.

The slave address is the value of the rotary switch setting plus one.

③ Function : 2 character(1byte) function code to be executed.

④ Data : Data to be processed by the slave.

⑤ LRC : This field checks the integrity of the message, exclusive of the beginning "colon" and ending CRLF pair. The LRC is calculated by adding successive bytes expressed by the ASCII message, discarding any carries, and then two"s complementing the result.

⑥ The maximum interval between two consecutive characters is 1 second.

4. Format of Message (Modbus RTU)

Start	Address	Function	Data	CRC
Silent interval (3.5 char times)	1 byte	1 byte	n byte	2 Chars

① Start : In RTU mode, the messages start with a silent interval of at least 3.5 character times.

② Address : Address to select slave. This field is the first content of querys or messages.

The address(ID) of MECQ2-SERVO can be set address from 1 to 16. 0 is assigned for broadcast.

③ Function : 1byte function code to be executed.

④ Data : Data to be processed by the slave.

⑤ CRC : In RTU mode, messages include an error–checking field that is based on a Cyclical Redundancy Check (CRC) method. This field consists of 2 bytes and Lower byte is sent first. The CRC Check Method is CRC-16-IBM (X^16 + X^15 + X^2 + 1).

The CRC is started by first preloading a 16–bit register to all 1"s. Then the process begins of applying successive 8–bit bytes of the message to the current contents of the register. Only the eight bits of data in each character are used for generating the CRC. Start bit, stop bits and the parity bit are not apply to the CRC.

1.3 Function and Data Field

1. Read Holding Register (0x03)
This function reads contents of holding registers in the slave. Broadcast is not supported.

Query

Slave Add	Fn	Data				LRC/CRC	
0x□□	0x03	Start add (high)	Start add (low)	No of reg (high)	No of reg (low)	(low)	(high)
		0x□□	0x□□	0x□□	0x□□	0x□□	0x□□

The query message specifies a starting register and the quantity of registers to be read.

Response

Slave Add	Fn	Data						LRC/CRC	
0x□□	0x03	Byte count	Reg value (high)	Reg value (low)	Reg value (high)	...	No of reg (low)	(low)	(high)
		0x□□	0x□□	0x□□	0x□□	...	0x□□	0x□□	0x□□

The register value in the response message are packed as two bytes per register. For each register, the first byte contains the high order bits and the second contains the low order bits.

2. Write Single Register (0x06)

This function writes a value into a single holding register. When broadcast, this function writes the same register reference in all attached slaves.

Query

Slave Add	Fn	Data				LRC/CRC	
0x□□	0x06	Start add (high)	Start add (low)	Value (high)	Value (low)	(low)	(high)
		0x□□	0x□□	0x□□	0x□□	0x□□	0x□□

The query message specifies a register value to be written.

Response

Slave Add	Fn	Data				LRC/CRC	
0x□□	0x06	Start add (high)	Start add (low)	Value (high)	Value (low)	(low)	(high)
		0x□□	0x□□	0x□□	0x□□	0x□□	0x□□

The normal response is an echo of the query.

3. Write Multiple Register (0x10)

This function writes values into a sequence of holding registers. When broadcast, this function writes the same register references in all attached slaves.

Query

Slave Add	Fn	Data									LRC/CRC	
0x□□	0x10	Start add (high)	Start add (low)	No of reg (high)	No of reg (low)	Byte count	Val 1 (high)	Val 1 (low)	...	Val n (high)	Val n (low)	(low)	(high)
		0x□□	0x□□	0x□□	0x□□	0x□□	0x□□	0x□□	...	0x□□	0x□□	0x□□	0x□□

Response

Slave Add	Fn	Data				LRC/CRC	
0x□□	0x10	Start add (high)	Start add (low)	No of reg (high)	No of reg (low)	(low)	(high)
		0x□□	0x□□	0x□□	0x□□	0x□□	0x□□

The normal response returns its slave address, function code, starting address, and the quantity of registers being written.

4. Report Slave ID (0x11)

This function requests the type of slave, the current running status and other information specific to the slave device at the specified address.
Broadcast is not supported.

Query

Slave Add	Fn	LRC/CRC	
0x□□	0x11	(low)	(high)
		0x□□	0x□□

Response

Slave Add	Fn	Data			LRC/CRC	
0x□□	0x11	Byte Count	Slave ID	Run Status	(low)	(high)
		0x02	0x68	0xFF	0x□□	0x□□

CK10 returns 0x68(104) and 0xFF as a slave ID and Running status.

1.4 Exception Responses

The slaves respond to the master"s query except broadcast, and there are several cases of responses.

1. If the slave receives the master"s query without a communication error, and can execute it normally. it returns a normal response.
2. If the slave does not receive the master"s query, no response is returned. The master will eventually process a timeout condition for the query.
3. If the slave receives the master"s query, but detects a communication error (parity, LRC, or CRC), no response is returned. The master will eventually process a timeout condition for the query.
4. If the slave receives the master"s query without a communication error, but cannot execute it (for example, invalid address, invalid data or unsupported command), the slave will return an exception response to inform the cause of the error.

Two fields of the exception response are different from normal response.

1. Function Code Field

   In a normal response, the function code field of response is filled with that of the master"s query. And the most–significant bit(MSB) of the function code is "0".

   In an exception response, the slave sets the MSB of the function code to "1". With this setting MSB to 1, the master can recognize the exception response and can examine the data field as the exception code.

2. Data Field

   In a normal response, the slave may return requested data in the data field.

   In an exception response, the slave returns an exception code in the data field. This field defines the cause of the exception.

Query

Slave Add	Fn	Data				LRC/CRC	
0x□□	0x06	Start add (high)	Start add (low)	Value (high)	Value (low)	(low)	(high)
		0x□□	0x□□	0x□□	0x□□	0x□□	0x□□

Exception Response

Slave Add	Fn	Data	LRC/CRC	
0x□□	0x86	Exception code	(low)	(high)
		0x02	0x□□	0x□□

This example response shows the exception of "ILLEGAL DATA ADDRESS".

Exception Codes of CK10

Code	Name	Description
01	ILLEGAL FUNCTION	The function code of the query is not supported.
02	ILLEGAL DATA ADDRESS	The data address of the query is not allowed.
03	ILLEGAL DATA VALUE	The data value in the query is not allowed.
04	SLAVE DEVICE FAILURE	An error occurred while the slave was trying to execute the requested function.

2. Address and Definitions of Registers

CK10 can be controlled by accessing registers in Modbus communication mode. The registers are 16bit(1word), 32bit(2words) 64bit(4 words) wide. The 32bit(2word) and 64bit(4word) register must be accessed from low address(most significant word) to high address(least significant word) consecutively. If the order of master"s accessing is different from the defined one, the drive might returns unexpected values or result.

There are three kinds of attributes : RO (Read Only, WO (Write Only), RW (Read & Write)) attribute of register. Attribute violations can cause communication errors. Please note them.

Some of registers are used to execute action only. For these kinds of registers, the read value may not be the same as the written one. Please refer to the descriptions for more details.

2.1 Status Registers

Add (Hex)	Add (Dec)	Size (word)	Name	RW	Description
3E8	1000	2	MotionStatus	RO	Motion status register. Please refer to "3.1 Definitions of status Flags".
3EA	1002	2	ActPos	RO	Actual Position
3EC	1004	2	CmdPos	RO	Command Position
3EE	1006	2	Velocity	RO	Velocity
3F0	1008	2	PosError	RO	Position Error
3F2	1010	2	InputAll	RO	Input port value. The register reflects user input and control input. Please refer to "3.2 Definition of Input Bits".
3F4	1012	2	OutputAll	RO	Output port value. This register reflects user output and control output. Please refer to "3.3 Definition of Output Bits".
3F6	1014	1	CurPtNo	RO	Current PT number
3F7	1015	1	PT Status	RO	PT Status (0 = Stop, 1 = Running)
3F8	1016	1	PushStatus	RO	Push Motion Status 0=Normal motion 1=Push motioning but the work is not detected 2=Work detected and torque is maintained 3=Work is not detected within push motion area
3F9	1017	1	CurAlarm	RO	Current alarm number
3FA	1018	1	MotorID	RO	Motor ID Number

2.2 User IO Registers

Add (Hex)	Add (Dec)	Size (word)	Name	RW	Description
406	1030	1	USR_IN_ALL	RO	Current input status of User IN0~IN8 - Bit0=USR_IN0 - Bit1=USR_IN1 - … - Bit7=USR_IN7 - Bit8=USR_IN8
407	1031	1	USR_IN0	RO	Current input status of User IN0 (0 or 1)
408	1032	1	USR_IN1	RO	Current input status of User IN1 (0 or 1)
409	1033	1	USR_IN2	RO	Current input status of User IN2 (0 or 1)
40A	1034	1	USR_IN3	RO	Current input status of User IN3 (0 or 1)
40B	1035	1	USR_IN4	RO	Current input status of User IN4 (0 or 1)
40C	1036	1	USR_IN5	RO	Current input status of User IN5 (0 or 1)
40D	1037	1	USR_IN6	RO	Current input status of User IN6 (0 or 1)
40E	1038	1	USR_IN7	RO	Current input status of User IN7 (0 or 1)
40F	1039	1	USR_IN8	RO	Current input status of User IN8 (0 or 1)
410	1040	1	USR_OUT_ALL	RW	User output register - Bit0=USR_OUT0 - … - Bit8= USR_OUT8 Bit set = Activation, Bit reset= Deactivation
411	1041	1	USR_OUT0	RW	User OUT0 bit register A writing the value except "0" to this register activates "User OUT 0", and "0" deactivates it.
412	1042	1	USR_OUT1	RW	User OUT1 bit register
413	1043	1	USR_OUT2	RW	User OUT2 bit register
414	1044	1	USR_OUT3	RW	User OUT3 bit register
415	1045	1	USR_OUT4	RW	User OUT4 bit register
416	1046	1	USR_OUT5	RW	User OUT5 bit register
417	1047	1	USR_OUT6	RW	User OUT6 bit register
418	1048	1	USR_OUT7	RW	User OUT7 bit register
419	1049	1	USR_OUT8	RW	User OUT8 bit register

2.3 Alarm Log

Add (Hex)	Add (Dec)	Size (word)	Name	RW	Description
423	1059	1	ALM_CLR	WO	A writing 0xFFFF to this register clears alarm log.
424	1060	1	ALM_LOG0	RO	Alarm Log#0 (Oldest alarm)
425	1061	1	ALM_LOG1	RO	Alarm Log#1
...	...	...	...	...	…
440	1088	1	ALM_LOG28	RO	Alarm Log#28
441	1089	1	ALM_LOG29	RO	Alarm Log#29

Alarm type : No alarm(0), OverCurrent(1), OverSpeed(2), Position Tracking(3), OverLoad(4), OverTemperature(5), BackEMF(6), MotorConnect(7), EncoderConnect(8), InPosition(10), ROMdevice(12), Position Overflow(15).

2.4 Motion Control Registers

Add (Hex)	Add (Dec)	Size (word)	Name	RW	Description
7D0	2000	2	AxisStartSpd	RW	Defines the start speed of motion. This is an alias for the register with address 0x138C.
7D2	2002	1	AxisAccTime	RW	Define the acceleration time of motion. This is an alias for the register with address 0x138E.
7D3	2003	1	AxisDecTime	RW	Define the deceleration time of motion. This is an alias for the register with address 0x1390
7D4	2004	2	AxisMoveSpd	RW	Defines the move speed of motion.
7D6	2006	2	SetDistPos	RW	Defines absolute position or distance to move.
7D8	2008	1	MoveTypeTrg	RW	A writing the below motion type to this register starts the specified motion. 1=Moves to the specified relative position. 2=Moves to the specified absolute position. The last updated motion type is returned when it read.
7D9	2009	1	JogAccDecTime	RW	Defines the acc/dec time of motion. This is an alias for the register with address 0x1398.
7DA	2010	2	JogStartSpd	RW	Defines the start speed of jog motion. This is an alias for the register with address 0x1396.
7DC	2012	2	JogSpd	RW	Defines the move speed of jog motion. This is an alias for the register with address 0x1394.
7DE	2014	1	JogDir	RW	Defines the direction of jog motion. 0 = CCW, 1=CW
7DF	2015	1	JogTrg	RW	A writing "1" to this register starts jog motion. It returns '0' when reading.

Continue

Add (Hex)	Add (Dec)	Size (word)	Name	RW	Description
7E0	2016	2	PushPosStSpd	RW	Defines the start speed of the position motion.
7E2	2018	2	PushPosSpd	RW	Defines the move speed of the position motion.
7E4	2020	1	PushPosAcc	RW	Defines the acceleration time of the position motion.
7E5	2021	1	PushPosDec	RW	Defines the deceleration time of the position motion.
7E6	2022	1	PushMod	RW	Defines the behavior when a work is detected in the push motion. 0=Stop Mode 1=Non-Stop Mode Please refer to the "6-6 Push Motion" in the text section of user manual .
7E7	2023	1	PushRatio	RW	Defines the torque value in the push motion.
7E8	2024	2	PushSpd	RW	Defines the speed of the push motion.
7EA	2026	2	PushStartPos	RW	Defines the absolute position to start the push motion after the position motion.
7EC	2028	2	PushEndPos	RW	Defines the absolute position as which the push motion ends.
7EE	2030	1	DoPushMotion	RW	A writing "1" to this register starts the position motion followed by the push motion. It returns "0" when reading.
7EF	2031	1	MoveOrigin	RW	A writing "1" to this register starts the homing operation specified by parameters. It returns "0" when reading.
7F0	2032	1	MoveStop	RW	Stops current motion. 1=Deceleration Stop 0=Immediate Stop It returns "0" when reading.
7F1	2033	1	ServoOn	RW	Enable or Disable motor drive. 0=Disable the drive The other value = Enable the drive
7F2	2034	1	ClrPos	RW	A writing "1" to this register clears command position and actual position to "0". It returns "0" when reading.
7F3	2035	1	RstAlarm	RW	A writing "1" to this register clears alarm state. If the drive is in EMG state, it tries to exit EMG state. It returns "0" when reading.
7F5	2037	1	SavePT	RW	Save position table (RAM) to flash memory.
7F6	2038	1	PtNoToStart	RW	Defines PT number to execute.
7F7	2039	1	DoPT	RW	1 =Start PT specified in "PtNoToStart" 2 =Do single PT specified in "PtNoToStart" 3 =Do the next PT of current one

2.5 Compare Out Function Register

This function generates pulses periodically in specified conditions. Pulses are output to the port set to "Compare Out".

Add (Hex)	Add (Dec)	Size (word)	Name	RW	Description
7F8	2040	2	CmpStPos	RW	Defines the start position to output pulse. The position is in absolute coordinate system.
7FA	2042	2	CmpPeriod	RW	Defines the position period to output pulses.
7FC	2044	1	CmpPlsWidth	RW	Defines the pulse width Range : 1 ~ 1000[ms]
7FD	2045	1	CmpEnable	RW	0 = Disable the function 1 = Enable the function
7FE	2046	1	CmpStat	RO	0 = Disabled 1 = Enabled

2.6 Parameter Register

1. Parameter Control

Add (Hex)	Add (Dec)	Size (word)	Name	RW	Description
7F4	2036	1	SaveAllParam	RW	It saves all parameter from RAM (0x1388~0x1431) to flash memory(0x1450~0x14F9) when writing "1". (Parameter and IO setting) It returns "0" when reading.

2. Parameters in RAM

This area is used to set the drive's motion, I/O, and other functions. Since it is stored in RAM, the setting value is lost when the power is turned off. Please use the 0x7F4 register ("SaveAllParam") to store in flash memory.

Add (Hex)	Add (Dec)	Size (word)	Name	RW	Description
Motion Parameters					
1388	5000	2	Pulse per Revolution	RW	Please refer to the "3-4 Parameter List" or the"8-2 Parameter Description" in the text section of user manual .
138A	5002	2	Axis Max Speed	RW	
138C	5004	2	Axis Start Speed	RW	
138E	5006	2	Axis Acc Time	RW	
1390	5008	2	Axis Dec Time	RW	
1392	5010	2	Reserved	RW	
1394	5012	2	Jog Speed	RW	
1396	5014	2	Jog Start Speed	RW	
1398	5016	2	Jog Acc Dec Time	RW	
139A	5018	2	S/W Limit Plus value	RW	
139C	5020	2	S/W Limit Minus value	RW	

Continue

Add (Hex)	Add (Dec)	Size (word)	Name	RW	Description
Motion Parameters					
139E	5022	2	S/W Limit Stop Method	RW	Please refer to the "3-4 Parameter List" or the"8-2 Parameter Description" in the text section of user manual .
13A0	5024	2	H/W Limit Stop Method	RW	
13A2	5026	2	Reserved	RW	
13A4	5028	2	Org Speed	RW	
13A6	5030	2	Org Search Speed	RW	
13A8	5032	2	Org Acc Dec Time	RW	
13AA	5034	2	Org Method	RW	
13AC	5036	2	Org Dir	RW	
13AE	5038	2	Org Offset	RW	
13B0	5040	2	Org Position Set	RW	
13B2	5042	2	Reserved	RW	
13B4	5044	2	Position Loop Gain	RW	
13B6	5046	2	Inpos Value	RW	
13B8	5048	2	Pos Tracking Limit	RW	
13BA	5050	2	Motion Dir	RW	
13BC	5052	2	Reserved	RW	
13BE	5054	2	Org Torque Ratio	RW	
13C0	5056	2	Pos. Error Overflow Limit	RW	
13C2	5058	2	Reserved	RW	
13C4	5060	2	Run Current	RW	
13C6	5062	2	Boost Current	RW	
13C8	5064	2	Stop Current	RW	
13CA	5066	2	EMG Mode	RW	
13CC	5068	2	EMG Brake mode	RW	
13CE	5070	2	Operating Mode	RW	
13D0	5072	2	Tact Switch Func	RW	
13D2	5074	2	Baudrate	RW	
13D4	5076	2	Restore Pos/PtNo	RW	
13D6	5078	2	Motor No Display	RW	
IO Settings					
13D8	5080	2	IOAssign flag	RW	Reserved
13DA	5082	2	IOAssign N1	RW	Defines the function of IN1~12 pin. Please refer to"3-2 Definitions of Input Bits". Ex.: "0x00000002" assigns "Limit-"
13DC	5084	2	IOAssign IN2	RW	
13DE	8086	2	IOAssign IN3	RW	

Continue

Add (Hex)	Add (Dec)	Size (word)	Name	RW	Description
IO Settings					
13E0	5088	2	IOAssign IN4	RW	Defines the function of IN1~12 pin. Please refer to"3-2 Definitions of Input Bits". Ex.: "0x00000002" assigns "Limit-"
13E2	5090	2	IOAssign IN5	RW	
13E4	5092	2	IOAssign IN6	RW	
13E6	5094	2	IOAssign IN7	RW	
13E8	5096	2	IOAssign IN8	RW	
13EA	5098	2	IOAssign IN9	RW	
13EC	5100	2	IOAssign IN10	RW	
13EE	5102	2	IOAssign IN11	RW	
13F0	5104	2	IOAssign IN12	RW	
13F2	5106	2	IOAssign OUT1	RW	Defines the function of OUT1~10 pin. Please refer to "3-3 Definitions of Output Bits" Ex.: "0x04000002" assigns "Inposition " "0x04000000" assigns "User IN 0 "
13F4	5108	2	IOAssign OUT2	RW	
13F6	5110	2	IOAssign OUT3	RW	
13F8	5112	2	IOAssign OUT4	RW	
13FA	5114	2	IOAssign OUT5	RW	
13FC	5116	2	IOAssign OUT6	RW	
13FE	5118	2	IOAssign OUT7	RW	
1400	5120	2	IOAssign OUT8	RW	
1402	5122	2	IOAssign OUT9	RW	
1404	5124	2	IOAssign OUT10	RW	
1406	5126	2	IO Polarity IN1	RW	Defines the Polarity of IN 1~12 pin. When Polarity = 0 : OFF → 0, ON → 1 When Polarity = 1 : OFF → 1, ON → 0
1408	5128	2	IO Polarity IN2	RW	
140A	5130	2	IO Polarity IN3	RW	
140C	5132	2	IO Polarity IN4	RW	
140E	5134	2	IO Polarity IN5	RW	
1410	5136	2	IO Polarity IN6	RW	
1412	5138	2	IO Polarity IN7	RW	
1414	5140	2	IO Polarity IN8	RW	
1416	5142	2	IO Polarity IN9	RW	
1418	5144	2	IO Polarity IN10	RW	
141A	5146	2	IO Polarity IN11	RW	
141C	5148	2	IO Polarity IN12	RW	

Continue

Add (Hex)	Add (Dec)	Size (word)	Name	RW	Description
IO Settings					
141E	5150	2	IO Polarity OUT1	RW	Defines the polarity of OUT1~12 pin. When polarity = 0 : 0 → OFF, 1 → ON When polarity = 1 : 0 → ON, 1 → OFF
1410	5152	2	IO Polarity OUT2	RW	
1412	5154	2	IO Polarity OUT3	RW	
1414	5156	2	IO Polarity OUT4	RW	
1416	5158	2	IO Polarity OUT5	RW	
1418	5160	2	IO Polarity OUT6	RW	
142A	5162	2	IO Polarity OUT7	RW	
142C	5164	2	IO Polarity OUT8	RW	
142E	5166	2	IO Polarity OUT9	RW	
1430	5168	2	IO Polarity OUT10	RW	

3. Parameters in ROM(Flash)

This area is used to read the parameters in the ROM(Flash). It cannot be written directly.

Add (Hex)	Add (Dec)	Size (word)	Name	RW	Description
Motion Parameters					
1450	5200	2	Pulse per Revolution	RO	
1452	5202	2	Axis Max Speed	RO	
1454	5204	2	Axis Start Speed	RO	
1456	5206	2	Axis Acc Time	RO	
1458	5208	2	Axis Dec Time	RO	
145A	5210	2	Reserved	RO	
145C	5212	2	Jog Speed	RO	
145E	5214	2	Jog Start Speed	RO	
1460	5216	2	Jog Acc Dec Time	RO	
1462	5218	2	S/W Limit Plus Value	RO	
1464	5220	2	S/W Limit Minus Value	RO	
1466	5222	2	S/W Limit Stop Method	RO	
1468	5224	2	H/W Limit Stop Method	RO	
146A	5226	2	Reserved	RO	
146C	5228	2	Org Speed	RO	
146E	5230	2	Org Search Speed	RO	
1470	5232	2	Org Acc Dec Time	RO	
1472	5234	2	Org Method	RO	
1474	5236	2	Org Dir	RO	
1476	5238	2	Org Offset	RO	
1478	5240	2	Org Position Set	RO	

Continue

Add (Hex)	Add (Dec)	Size (word)	Name	RW	Description
Motion Parameters					
147A	5242	2	Reserved	RO	
147C	5244	2	Position Loop Gain	RO	
147E	5246	2	Inpos Value	RO	
1480	5248	2	Pos Tracking Limit	RO	
1482	5250	2	Motion Dir	RO	
1484	5252	2	Reserved	RO	
1486	5254	2	Org Torque Ratio	RO	
1488	5256	2	Pos. Error Overflow Limit	RO	
148A	5258	2	Reserved	RO	
148C	5260	2	Run Current	RO	
148E	5262	2	Start Current	RO	
1490	5264	2	Stop Current	RO	
1492	5266	2	EMG Mode	RO	
1494	5268	2	EMG Brake Mode	RO	
1496	5270	2	Operating Mode	RO	
1498	5272	2	Tact Switch Func	RO	
149A	5274	2	Baudrate	RO	
149C	5276	2	Restore Pos/PtNo	RO	
149E	5278	2	Motor No Display	RO	
IO Settings					
14A0	5280	2	IOAssign flag	RO	
14A2	5282	2	IOAssign IN1	RO	
14A4	5284	2	IOAssign IN2	RO	
14A6	5286	2	IOAssign IN3	RO	
14A8	5288	2	IOAssign IN4	RO	
14AA	5290	2	IOAssign IN5	RO	
14AC	5292	2	IOAssign IN6	RO	
14AE	5294	2	IOAssign IN7	RO	
14B0	5296	2	IOAssign IN8	RO	
14B2	5298	2	IOAssign IN9	RO	
14B4	5300	2	IOAssign IN10	RO	
14B6	5302	2	IOAssign IN11	RO	
14B8	5304	2	IOAssign IN12	RO	
14BA	5306	2	IOAssign OUT1	RO	
14BC	5308	2	IOAssign OUT2	RO	
14BE	5310	2	IOAssign OUT3	RO	

Continue

Add (Hex)	Add (Dec)	Size (word)	Name	RW	Description
IO Settings					
14C0	5312	2	IOAssign OUT4	RO	
14C2	5314	2	IOAssign OUT5	RO	
14C4	5316	2	IOAssign OUT6	RO	
14C6	5318	2	IOAssign OUT7	RO	
14C8	5320	2	IOAssign OUT8	RO	
14CA	5322	2	IOAssign OUT9	RO	
14CC	5324	2	IOAssign OUT10	RO	
14CE	5326	2	IO Polarity IN1	RO	
14D0	5328	2	IO Polarity IN2	RO	
14D2	5330	2	IO Polarity IN3	RO	
14D4	5332	2	IO Polarity IN4	RO	
14D6	5334	2	IO Polarity IN5	RO	
14D8	5336	2	IO Polarity IN6	RO	
14DA	5338	2	IO Polarity IN7	RO	
14DC	5340	2	IO Polarity IN8	RO	
14DE	5342	2	IO Polarity IN9	RO	
14E0	5344	2	IO Polarity IN10	RO	
14E2	5346	2	IO Polarity IN11	RO	
14E4	5348	2	IO Polarity IN12	RO	
14E6	5350	2	IO Polarity OUT1	RO	
14E8	5352	2	IO Polarity OUT2	RO	
14EA	5354	2	IO Polarity OUT3	RO	
14EC	5356	2	IO Polarity OUT4	RO	
14EE	5358	2	IO Polarity OUT5	RO	
14F0	5360	2	IO Polarity OUT6	RO	
14F2	5362	2	IO Polarity OUT7	RO	
14F4	5364	2	IO Polarity OUT8	RO	
14F6	5366	2	IO Polarity OUT9	RO	
14F8	5368	2	IO Polarity OUT10	RO	

2.7 Position Table Registers

1. Position Table Control

Add (Hex)	Add (Dec)	Size (word)	Name	RW	Description
7F5	2037	1	PosTableWriteROM	RW	It saves all Position Table from RAM (0x2710~0x370F) to flash memory (0x4E20~0x5E1F) when writing "1". It returns "0" when reading.

2. Position Table in RAM

Add (Hex)	Add (Dec)	Size (word)	Name	RW	Description
2710	10000	2	PT0 Position	RW	Specifies distance or absolute position to move. [pulse]
2712	10002	2	PT0 Low Speed	RW	Specifies start speed [pulse/s]
2714	10004	2	PT0 High Speed	RW	Specifies motion speed [pulse/s]
2716	10006	1	PT0 Accel Time	RW	Specifies acceleration rate [ms]
2717	10007	1	PT0 Decel Time	RW	Specifies deceleration rate [ms]
2718	10008	1	PT0 Command	RW	Specifies the type of motion. It is ignored when written to 0xFFFF.
2719	10009	1	PT0 WaitTime	RW	Specifies the waiting time before starting the next PT. [ms] If the next PT is not present, this register is ignored.
271A	10010	1	PT0 Reserved1	RW	Reserved for continuous motion. Please set to "0".
271B	10011	1	PT0 Jump Table No	RW	When this register specifies, the system jumps to "JP Table No" and execute it after the current position table(PT). If "JP Table No" specifies as 10XXX, system jumps to Position No XXX as soon as "JPT Start" single is asserted. This is ignored when written to 0xFFFF.
271C	10012	1	PT0 JPT0	RW	If any of these registers are specified, and there are corresponding input signals of JPT input0,1 ,or 2, system jumps to JPT0, 1, or 2 accordingly regardless of specified "Jump Table No". This is ignored when written to 0xFFFF.
271D	10013	1	PT0 JPT1	RW	
271E	10014	1	PT0 JPT2	RW	
271F	10015	1	PT0 LoopCount	RW	If these registers are specified, system repeats PT action(s) specified times (Loop Count) and after then jumps to corresponding PT to "Loop Jump Table No" regardless of specified "Jump Table No".
2720	10016	1	PT0 Loop Jump Table No.	RW	
2721	10017	1	PT0 PT set	RW	Specified output signals such as PT Output0, 1, 2 for the start, pass, or end signals. Bit0 : Select(1)/ Deselect(0) PT Output0 Bit1 : Select(1)/ Deselect(0) PT Output1 Bit2 : Select(1)/ Deselect(0) PT Output2 Bit[4..5]: "00"= Assert signals when start PT "01"= Assert signals when completing PT "10"= Assert signals when reaching the "Trigger Position"

Continue

Add (Hex)	Add (Dec)	Size (word)	Name	RW	Description
2722	10018	1	PT0 LoopCountClear	RW	This register specifies the PT number of the "Loop Count" to clear. This is ignored when written to 0xFFFF.
2723	10019	1	PT0 CheckInposition	RW	This register specifies whether to wait Inposition signals to complete the current PT. 0= Ignore Inposition 1= Wait Inposition
2724	10020	2	PT0 ComparePos	RW	Specifies the position where the PT Output signals is output.
2726	10022	1	PT0 CompareOnTime	RW	Specifies the pulse width of PT Output0~2 signals.
2727	10023	1	PT0 PushRatio	RW	Specifies motor torque ratio for push motion.
2728	10024	2	PT0 PushSpeed	RW	Specifies motion speed of push motion.
272A	10026	2	PT0 PushPosition	RW	Specifies absolute target position of push motion.
272C	10028	1	PT0 PushMode	RW	Specifies the push mode 0 = Stop mode 1~10,000 = Non-stop mode
272D	10029	3	PT0 Reserved2	RW	Please do not write these registers.
2730	10032	32	PT1		
2750	10064	32	PT2		
...	...		...		
36F0	14064	32	PT127		

Decimal address of PTn = n * 32 + 10000 (In case of hexadecimal address "n * 0x20 + 0x2710")
Erroneous data would cause errors while executing. Please be careful when writing registers.

3. Position Table in ROM

This area is used to read the position table in the ROM(Flash) area. It cannot be written directly.

Add (Hex)	Add (Dec)	Size (word)	Name	RW	Description
4E20	20000	2	PT0	RO	Read only area. For more details, please refer to the previous chapter.
4E40	20032	32	PT1	RO	
4E60	20064	32	PT2	RO	
...	...		...	RO	
5E00	24064	32	PT127	RO	

2.8 Drive Info Registers

Add (Hex)	Add (Dec)	Size (word)	Name	RW	Description
834	2100	2	MODEL_ID	RO	Model ID of drive CK10 returns 104.
836	2102	1	HW_TYPE	RO	H/W ID of drive CK10 returns 31
837	2103	1	FW_VER	RO	F/W version
838	2104	2	NO_PWR_CYCLE	RO	Total number of power cycle
83A	2106	4	TURN_TOTAL	RO	Total revolution count of driving [Rev]
83E	2110	4	TURN_CURPWR	RO	Revolution count after power on

3. Information for programming

3.1 Definitions of State Flags (Reg 0x3E8)

These definitions are used to distinguish motion status flags.

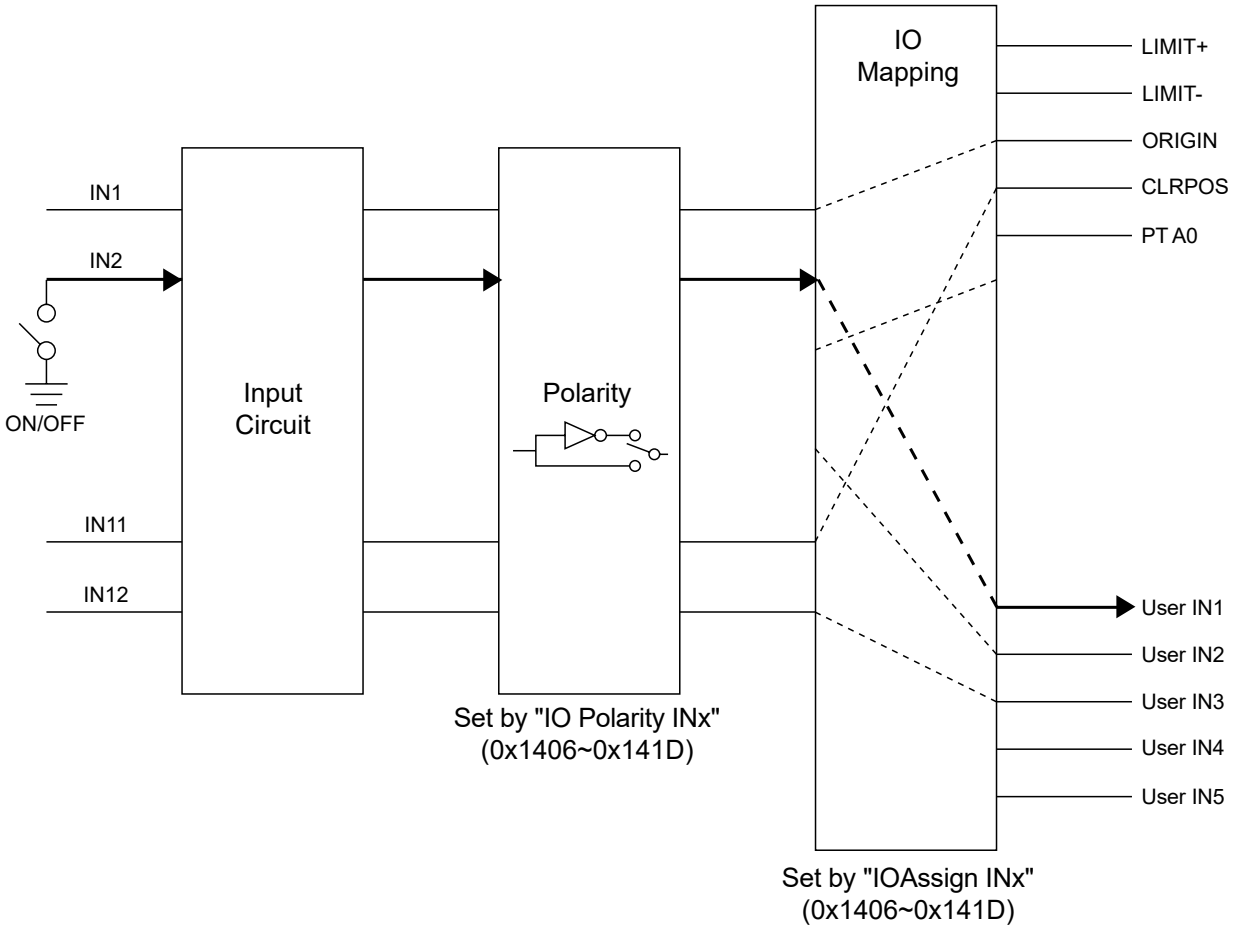
Name of flag define	Contents	Bit position
FFLAG_ERRORALL	One or more error occurred.	0x00000001
FFLAG_HWPOSILMT	"+" direction limit sensor is detected.	0x00000002
FFLAG_HWNEGALMT	"-" direction limit sensor is detected.	0x00000004
FFLAG_SWPOSILMT	"+" direction program limit is exceeded.	0x00000008
FFLAG_SWNEGALMT	"-" direction program limit is exceeded.	0x00000010
Reserved		0x00000020
Reserved		0x00000040
FFLAG_ERRPOSOVERFLOW	Position error is higher than "Pos Error Overflow Limit" parameter after position command.	0x00000080
FFLAG_ERROVERCURRENT	Drive has detected over-current.	0x00000100
FFLAG_ERROVERSPEED	The motor speed exceeded 3300 [rpm].	0x00000200
FFLAG_ERRPOSTRACKING	Position error is higher than "Pos Tracking Limit" parameter during position command run.	0x00000400
FFLAG_ERROVERLOAD	Load exceeded the max torque of the motor is loaded more than 5 seconds.	0x00000800
FFLAG_ERROVERHEAT	The internal temperature of the drive exceeded 85°C.	0x00001000
FFLAG_ERRBACKEMF	A counter electromotive force of the motor exceeded 48V.	0x00002000
Reserved	Always "0"	0x00004000
FFLAG_ERRINPOSITION	After operation is finished, a position error occurs for more than 3 seconds.	0x00008000
FFLAG_EMGSTOP	Drive is under emergency stop state.	0x00010000
FFLAG_SLOWSTOP	Motor is under general stop.	0x00020000
FFLAG_ORIGINRETURNING	Motor is returning to the origin.	0x00040000
FFLAG_INPOSITION	Inposition has been finished.	0x00080000
FFLAG_SERVOON	Drive is servo on state.	0x00100000
FFLAG_ALARMRESET	AlarmReset has run.	0x00200000
FFLAG_PTSTOPED	Position Table operation has been finished.	0x00400000
FFLAG_ORIGINSENSOR	Origin sensor input is activated.	0x00800000
FFLAG_ZPULSE	Motor is in the z-pulse position of encoder.	0x01000000
FFLAG_ORIGINRETOK	Origin return operation has been finished.	0x02000000
FFLAG_MOTIONDIR	To display the motor operating direction(+:Off, -:On)	0x04000000
FFLAG_MOTIONING	The motor is running.	0x08000000
Reserved	Always "0"	0x10000000
FFLAG_MOTIONACCEL	Motor is in acceleration section.	0x20000000
FFLAG_MOTIONDECEL	Motor is in acceleration section.	0x40000000
FFLAG_MOTIONCONST	Motor is in the constant speed section, not in acceleration or deceleration section.	0x80000000

3.2 Definitions of Input Bits (Reg 0x3F2, 0x13DA~0x13F1)

These definitions are used to distinguish which signals are inputs and to set the input ports to a specific function. In case of setting input port, it can be set one function per input port.

Signal name	Bit mask	Signal name	Bit mask	Signal name	Bit mask	Signal name	Bit mask
Limit+	0x00000001	PT A4	0x00000100	Alarm Reset	0x00010000	JPT input 2	0x01000000
Limit-	0x00000002	PT A5 / User IN 6	0x00000200	Servo ON	0x00020000	JPT Start	0x02000000
Origin	0x00000004	PT A6 / User IN 7	0x00000400	Reserved	0x00040000	User IN 0	0x04000000
Clear Position	0x00000008	PT A7 / User IN 8	0x00000800	Org Search	0x00080000	User IN 1	0x08000000
PT A0	0x00000010	PT Start	0x00001000	Reserved	0x00100000	User IN 2	0x10000000
PT A1	0x00000020	Stop	0x00002000	E-stop	0x00200000	User IN 3	0x20000000
PT A2	0x00000040	Jog+	0x00004000	JPT Input 0	0x00400000	User IN 4	0x40000000
PT A3	0x00000080	Jog-	0x00008000	JPT Input 1	0x00800000	User IN 5	0x80000000

The function and polarity of input port are programmable. Please refer to the below block-diagram.
Ex.: "User IN 1" is 1 when "IO polarity IN2"=0, "IO Assign IN2"=0x08000000(User IN1) and IN2=ON.



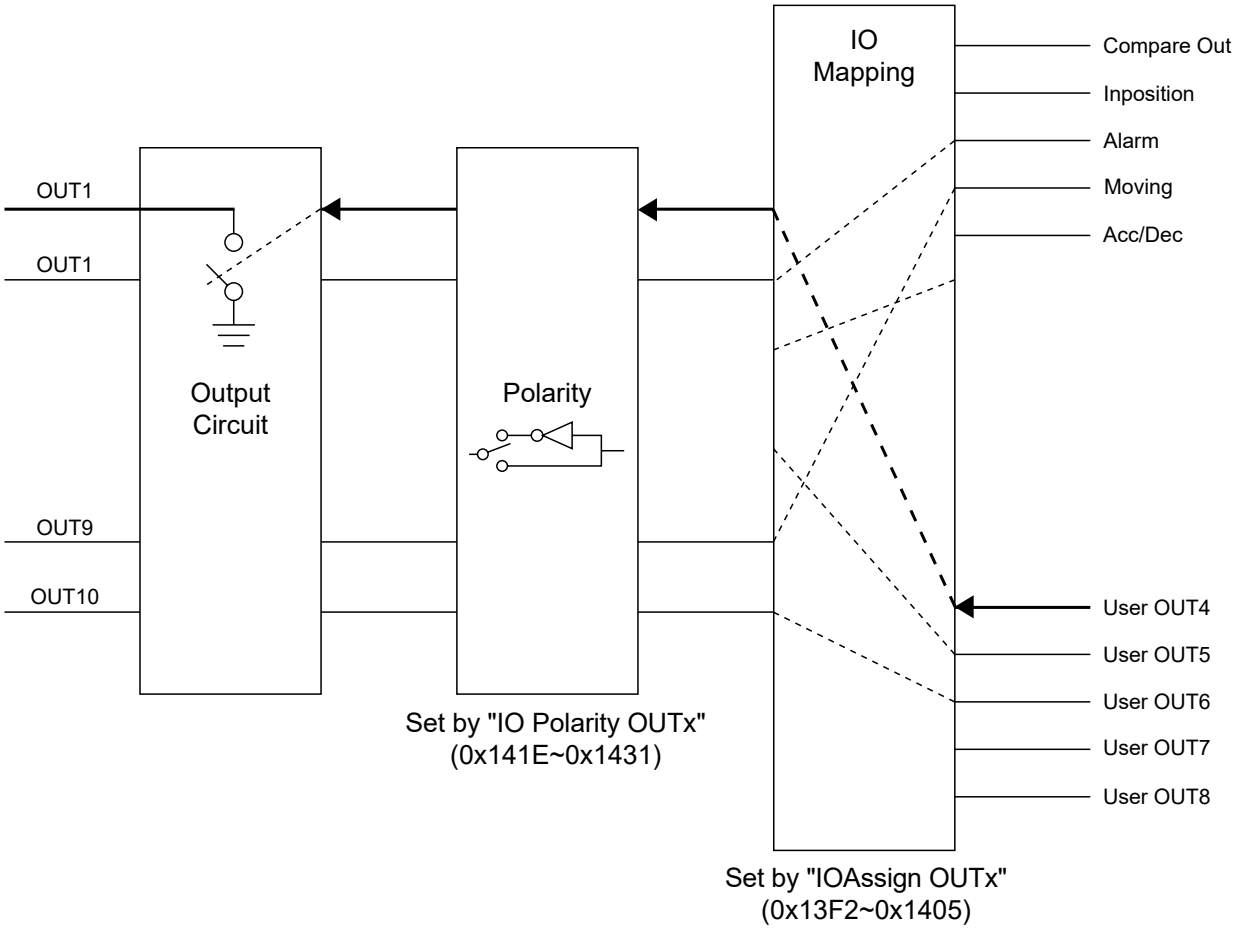
3.3 Definitions of Output Bits (Reg 0x3F4, 0x13F2~0x1405)

These definitions are used to read back and check output data, and set the output ports to a specific function. In case of setting output port, it can be set one function per input port.

Signal name	Relevant bit position	Signal name	Relevant bit position	Signal name	Relevant bit position
Compare Out	0x00000001	Origin Search OK	0x00000100	User Out 1	0x00010000
Inposition	0x00000002	Servo Ready	0x00000200	User Out 2	0x00020000
Alarm	0x00000004	reserved	0x00000400	User Out 3	0x00040000
Moving	0x00000008	reserved	0x00000800	User Out 4	0x00080000
Acc/Dec	0x00000010	PT Output0	0x00001000	User Out 5	0x00100000
ACK	0x00000020	PT Output1	0x00002000	User Out 6	0x00200000
END	0x00000040	PT Output2	0x00004000	User Out 7	0x00400000
AlarmBlink	0x00000080	User Out 0	0x00008000	User Out 8	0x00800000

The function and polarity of output port are programmable. Please refer to the below block-diagram.

Ex.:"OUT1" is ON when "IO polarity OUT1"=0, "IO Assign OUT1"=0x00080000(User OUT4) and User OUT4=1.



3.4 Parameter Lists (Reg 0x1388 ~0x13D7)

<table>
<tr><th>No.</th><th>Parameter name / Description</th><th>Unit</th><th>Lower limit</th><th>Upper limit</th><th>Default</th></tr>
<tr><td>0</td><td>Pulses per revolution
Number of pulses per revolution (P/R).
Changing this value will switch the motor to Servo OFF status.
<table>
<tr><th>Value</th><th>Pulses per revolution</th><th>Value</th><th>Pulses per revolution</th></tr>
<tr><td>0</td><td>MDB setting</td><td>8</td><td>4,800</td></tr>
<tr><td>1</td><td>500</td><td>9</td><td>5,000</td></tr>
<tr><td>2</td><td>1,000</td><td>10</td><td>6,400</td></tr>
<tr><td>3</td><td>1,600</td><td>11</td><td>8,000</td></tr>
<tr><td>4</td><td>2,000</td><td>12</td><td>9,600</td></tr>
<tr><td>5</td><td>3,200</td><td>13</td><td>10,000</td></tr>
<tr><td>6</td><td>3,600</td><td>14</td><td>16,000</td></tr>
<tr><td>7</td><td>4,000</td><td>15</td><td>20,000</td></tr>
</table></td><td>pulse/rev</td><td>0</td><td>50,000</td><td>0</td></tr>
<tr><td>1</td><td>Maximum axis speed – This parameter sets the maximum allowable speed when executing a position move command.
* The upper limit depends on the pulses per revolution setting:
If 10,000 P/R: 500,000 pps
If 16,000 P/R: 800,000 pps</td><td>pps</td><td>1</td><td>800,000</td><td>500,000</td></tr>
<tr><td>2</td><td>Axis start speed
This parameter sets the starting speed for position move commands.</td><td>pps</td><td>1</td><td>35,000</td><td>1</td></tr>
<tr><td>3</td><td>Axis acceleration time
This parameter sets the acceleration time for position move commands.</td><td>msec</td><td>1</td><td>9,999</td><td>100</td></tr>
<tr><td>4</td><td>Axis deceleration time
This parameter sets the deceleration time for position move commands.</td><td>msec</td><td>1</td><td>9,999</td><td>100</td></tr>
<tr><td>5</td><td>Reserved</td><td></td><td>100</td><td>100</td><td>100</td></tr>
<tr><td>6</td><td>Jog speed
This parameter sets the motor speed for jog operations.</td><td>pps</td><td>1</td><td>800,000</td><td>5,000</td></tr>
<tr><td>7</td><td>Jog start speed
This parameter sets the starting speed for jog operations.</td><td>pps</td><td>1</td><td>35,000</td><td>1</td></tr>
<tr><td>8</td><td>Jog acc/dec time
This parameter sets the acceleration/ deceleration time for jog operations.</td><td>msec</td><td>1</td><td>9,999</td><td>100</td></tr>
<tr><td>9</td><td>Software positive limit
This parameter sets the positive position limit for position moves (32 bits).</td><td>pulse</td><td>-2,147,483,648</td><td>+2,147,483,647</td><td>+2,147,483,647</td></tr>
<tr><td>10</td><td>Software negative limit
This parameter sets the negative position limit for position moves (32 bits).</td><td>pulse</td><td>-2,147,483,648</td><td>+2,147,483,647</td><td>+2,147,483,647</td></tr>
</table>

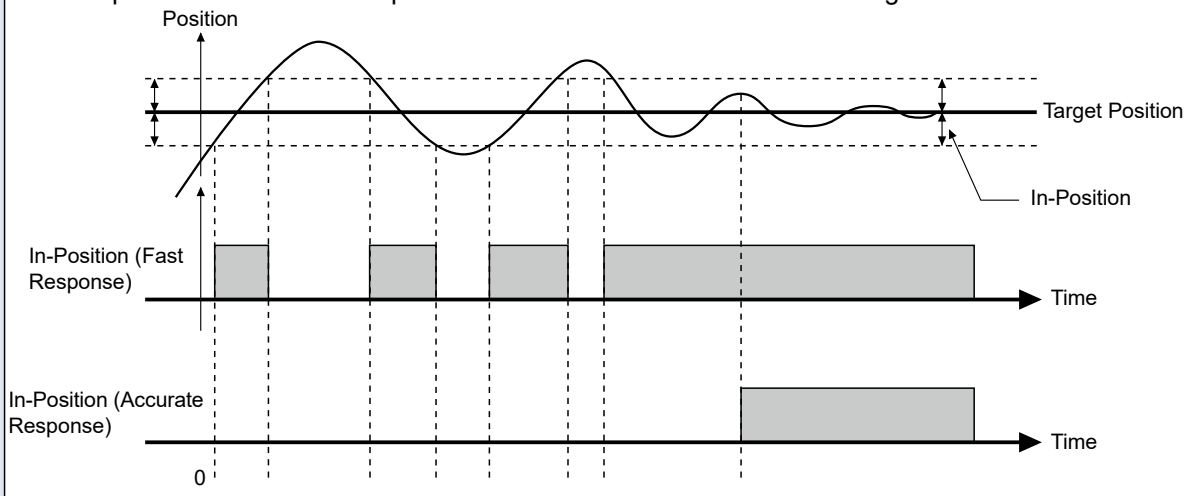
Continue

<table>
<tr><th>No.</th><th>Parameter name / Description</th><th>Unit</th><th>Lower limit</th><th>Upper limit</th><th>Default value</th></tr>
<tr><td>11</td><td>Software limit stop method – Sets the stop method when the motor reaches the software limit:
0 : Stop the motor using Emergency Stop
1 : Stop the motor using Deceleration Stop
2 : Do not use the software limit</td><td></td><td>0</td><td>2</td><td>2</td></tr>
<tr><td>12</td><td>Hardware limit stop method – Sets the stop method when the motor reaches the hardware limit:
0 : Stop the motor using Emergency Stop
1 : Stop the motor using Deceleration Stop</td><td></td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>13</td><td>Reserved</td><td></td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>14</td><td>Origin speed – This parameter sets the speed before detecting the origin during an origin return operation.</td><td>pps</td><td>1</td><td>800,000</td><td>5,000</td></tr>
<tr><td>15</td><td>Origin search speed – This parameter sets the speed returning to the origin after detecting it during an origin return operation.</td><td>pps</td><td>1</td><td>80,000</td><td>1,000</td></tr>
<tr><td>16</td><td>Origin acc/dec time – This parameter sets the acceleration and deceleration time for origin return operation.</td><td>msec</td><td>1</td><td>9,999</td><td>50</td></tr>
<tr><td>17</td><td>Homing method
User-selectable origin return method
<table><tr><td>0</td><td>Move the motor at origin speed to the origin sensor, then accurately return to the origin at origin search speed.</td></tr><tr><td>1</td><td>Move the motor at origin speed to the origin sensor, then accurately search the encoder Z-phase at origin search speed.</td></tr><tr><td>2</td><td>Move the motor at origin speed to the limit sensor and stop immediately.</td></tr><tr><td>3</td><td>Move the motor at origin speed to the limit sensor, then accurately search the encoder Z-phase at origin search speed.</td></tr><tr><td>4</td><td>Set the current position as the origin.</td></tr><tr><td>5</td><td>Search for the Z-phase at origin search speed as the origin.</td></tr><tr><td>6</td><td>Move in torque mode, stop after hitting the hardware limit, and set the current position as the origin.</td></tr><tr><td>7</td><td>Move in torque mode, stop after hitting the hardware limit, and search the Z-phase at origin search speed.</td></tr></table>* For more details, refer to 5.3 Origin return</td><td></td><td>0</td><td>7</td><td>0</td></tr>
<tr><td>18</td><td>Homing direction – This parameter sets the motor rotation direction during origin return.
0 : CW, rotate the motor clockwise
1 : CCW, rotate the motor counterclockwise</td><td></td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>19</td><td>Origin offset – After returning to the origin, the motor will apply the offset according to this setting and initialize the position.</td><td>pulse</td><td>-2,147,483,648</td><td>+2,147,483,647</td><td>0</td></tr>
<tr><td>20</td><td>Origin position – After returning to the origin, this value will be written to the command/actual position.</td><td>pulse</td><td>-2,147,483,648</td><td>+2,147,483,647</td><td>0</td></tr>
</table>

Continue

<table>
<tr><th>No.</th><th>Parameter name / Description</th><th>Unit</th><th>Lower limit</th><th>Upper limit</th><th>Default value</th></tr>
<tr><td>21</td><td>Reserved</td><td></td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>22</td><td>Position Loop Gain
After the motor stops, this parameter controls the motor"s response based on load. This is a relative value, not the actual internal drive value. For example, changing this value from 3 to 6 does not double the response time. A smaller parameter value makes the motor stop more sensitively, shortening the stop time; a larger value makes the stop less sensitive, lengthening the stop time.
Setup procedure is as follows
1. Set the value to 0
2. Increase the parameter until motor response stabilizes
3. Adjust the setting by increasing or decreasing one or two steps from the current value
<table>
<tr><th>Value</th><th>Integral time constant</th><th>Proportional gain</th><th>Value</th><th>Integral time constant</th><th>Proportional gain</th></tr>
<tr><td>0</td><td>1</td><td>1</td><td>32</td><td>5</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>2</td><td>33</td><td>5</td><td>2</td></tr>
<tr><td>2</td><td>1</td><td>3</td><td>34</td><td>5</td><td>3</td></tr>
<tr><td>3</td><td>1</td><td>4</td><td>35</td><td>5</td><td>4</td></tr>
<tr><td>4</td><td>1</td><td>5</td><td>36</td><td>5</td><td>5</td></tr>
<tr><td>5</td><td>1</td><td>6</td><td>37</td><td>5</td><td>6</td></tr>
<tr><td>6</td><td>1</td><td>7</td><td>38</td><td>5</td><td>7</td></tr>
<tr><td>7</td><td>1</td><td>2</td><td>39</td><td>5</td><td>2</td></tr>
<tr><td>8</td><td>2</td><td>1</td><td>40</td><td>6</td><td>1</td></tr>
<tr><td>9</td><td>2</td><td>2</td><td>41</td><td>6</td><td>2</td></tr>
<tr><td>10</td><td>2</td><td>3</td><td>42</td><td>6</td><td>3</td></tr>
<tr><td>11</td><td>2</td><td>4</td><td>43</td><td>6</td><td>4</td></tr>
<tr><td>12</td><td>2</td><td>5</td><td>44</td><td>6</td><td>5</td></tr>
<tr><td>13</td><td>2</td><td>6</td><td>45</td><td>6</td><td>6</td></tr>
<tr><td>14</td><td>2</td><td>7</td><td>46</td><td>6</td><td>7</td></tr>
<tr><td>15</td><td>2</td><td>8</td><td>47</td><td>6</td><td>8</td></tr>
<tr><td>16</td><td>3</td><td>1</td><td>48</td><td>7</td><td>1</td></tr>
<tr><td>17</td><td>3</td><td>2</td><td>49</td><td>7</td><td>2</td></tr>
<tr><td>18</td><td>3</td><td>3</td><td>50</td><td>7</td><td>3</td></tr>
<tr><td>19</td><td>3</td><td>4</td><td>51</td><td>7</td><td>4</td></tr>
<tr><td>20</td><td>3</td><td>5</td><td>52</td><td>7</td><td>5</td></tr>
<tr><td>21</td><td>3</td><td>6</td><td>53</td><td>7</td><td>6</td></tr>
<tr><td>22</td><td>3</td><td>7</td><td>54</td><td>7</td><td>7</td></tr>
<tr><td>23</td><td>3</td><td>2</td><td>55</td><td>7</td><td>2</td></tr>
<tr><td>24</td><td>4</td><td>1</td><td>56</td><td>8</td><td>1</td></tr>
<tr><td>25</td><td>4</td><td>2</td><td>57</td><td>8</td><td>2</td></tr>
<tr><td>26</td><td>4</td><td>3</td><td>58</td><td>8</td><td>3</td></tr>
<tr><td>27</td><td>4</td><td>4</td><td>59</td><td>8</td><td>4</td></tr>
<tr><td>28</td><td>4</td><td>5</td><td>60</td><td>8</td><td>5</td></tr>
<tr><td>29</td><td>4</td><td>6</td><td>61</td><td>8</td><td>6</td></tr>
<tr><td>30</td><td>4</td><td>7</td><td>62</td><td>8</td><td>7</td></tr>
<tr><td>31</td><td>4</td><td>8</td><td>63</td><td>8</td><td>8</td></tr>
</table></td><td></td><td>0</td><td>63</td><td>4</td></tr>
</table>

Continue

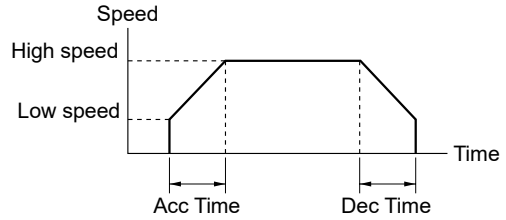
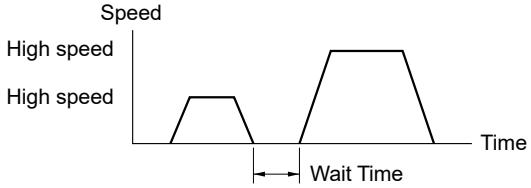
No.	Parameter name / Description	Unit	Lower limit	Upper limit	Default value
23	**In-position range** – Sets the conditions for the Inposition signal output. After a position command ends, if the position error is within the Inposition value, the completion signal is displayed and output. The position error for the Inposition signal output is 0–63, set values according to control mode as follows: 1. Fast Response Mode: 0–63 2. Accurate Response Mode: 64–127 Position error range is 0–63 depending on mode		0	127	0
	Fast response and accurate response control methods are shown in the figure below 				
24	**Position following error limit** – If the motor is running and the position error exceeds this value, an alarm is issued and the motor is disabled	pulse	1	+134,217,727	2,500
25	**Rotation direction** – Sets the motor rotation direction when executing a position command. 0 : Motor rotates clockwise 1 : Motor rotates counter-clockwise		0	1	0
26	Reserved		0	1	0
27	**Homing torque ratio** Sets the minimum torque to stop the motor when Origin Return Method parameter is set to 5 or 6	%	20	90	50
28	**Position overflow error limit** – Protects the controller and motor. When the motor is stopped and Servo is ON, if the position error exceeds the set value, an alarm is issued and the motor is disabled.	pulse	1	+134,217,727	2,500
29	Reserved	msec	10	5,000	200
30	**Running current** – Running current is the motor current during operation, automatically adjusted based on motor rated current and load. This value is related to motor torque; higher values result in higher torque. If torque is insufficient, increasing the running current can improve torque. ⚠ Caution : Increasing the current increases motor temperature rise ⚠ Caution : Maximum controller current must not exceed 4.8A	*10%	5	15	10

Continue

No.	Parameter name / Description	Unit	Lower limit	Upper limit	Default value
31	**Starting current** – When the motor cannot fully achieve the set acceleration time, supply current is increased to improve acceleration performance. Example 1. 28L (Current : 0.67 [A]) 2. Running current : 10 (100[%]) 3. Start current : 1 (50[%]) 4. Acceleration current : 0.67[A]+0.67[A]*50[%] = 1.00[A] ⚠ Caution : Increasing current will raise motor temperature. ⚠ Caution : Maximum controller current must not exceed 4.8A.	*50%	0	7	0
32	**Stop current** – Stop current is automatically set 0.1s after motor stops to reduce temperature during long idle periods. ⚠ Caution : Increasing current will raise motor temperature.	*10%	2	10	5
33	**Emergency mode** – When MG input port is active, execute 0 : Quick stop 1 : Quick stop, servo OFF 2 : Deceleration stop 3 : Deceleration stop, servo OFF 4 : Servo OFF		0	4	0
34	**Emergency brake mode** Select brake control in EMG state 0 : Brake locked 1 : Brake released (Brake excitation)		0	1	0
35	**Operation mode** – Set the controller operating mode. 0 : Single pulse mode(Pulse/Dir) 1 : Double pulse mode (CCW/DIR) 2 : Quadrature mode (A/B) 3 : Plus-R mode 4 : MODBUS RTU mode 5 : MODBUS ASC mode ⚠ Caution : Changing this parameter requires controller power cycle.		0	5	3
36	DIP switch function (SW3) 0 : Disabled 1 : Alarm reset		0	1	0
37	**Baud rate** – Sets communication baud rate. 0 : 9,600 1 : 19,200 2 : 38,400 3 : 57,600 4 : 115,200 ⚠ Caution : Changing this parameter requires controller power cycle.		0	4	4
38	**Stored position / Position table No.** This parameter sets whether CK10 saves the current position and position table to non-volatile memory when power is off. 0 : Disabled 1 : Enabled ⚠ Caution : Saved positions may not reflect actual positions.		0	1	1
39	Motor ID display This parameter sets whether the motor number is displayed via LED blinking. This function delays initialization by a few seconds. 0 : Disabled 1 : Enabled		0	1	0

3.5 Position Table Item (Reg 0x2710 ~ 0x370F)

For more details of position table functions, Please refer to the "Position table section of user manuals".

<table>
<tr><th>Designated item</th><th>Parameter name / Description</th><th>Unit</th><th>Lower limit</th><th>Upper limit</th></tr>
<tr><td>Command</td><td>Specified type of motion
<table>
<tr><th>Value</th><th>Command name</th></tr>
<tr><td>0</td><td>Abs Move low speed</td></tr>
<tr><td>1</td><td>Abs Move high speed</td></tr>
<tr><td>2</td><td>Abs Move high speed with deceleration</td></tr>
<tr><td>3</td><td>Abs Move with acceleration and deceleration</td></tr>
<tr><td>4</td><td>Inc Move low speed</td></tr>
<tr><td>5</td><td>Inc Move high speed</td></tr>
<tr><td>6</td><td>Inc Move high speed with deceleration</td></tr>
<tr><td>7</td><td>Inc Move with acceleration and deceleration</td></tr>
<tr><td>8</td><td>Move to Origin</td></tr>
<tr><td>9</td><td>Clear Position</td></tr>
<tr><td>10</td><td>Push Abs Move</td></tr>
<tr><td>11</td><td>Stop</td></tr>
</table></td><td>–</td><td>0</td><td>9</td></tr>
<tr><td>Position</td><td>Specified position/movement scale by number of pulse.</td><td>pulse</td><td>-2,147,483,648</td><td>+2,147,483,647</td></tr>
<tr><td>Low Speed</td><td>Specified low speed by number of pulse in accordance with type of motion.</td><td>pps</td><td>1</td><td>35,000</td></tr>
<tr><td>High Speed</td><td>Specified high speed by number of pulse in accordance with type of motion.</td><td>pps</td><td>1</td><td>500,000</td></tr>
<tr><td>ACC Time</td><td>Specified acceleration time by msec when starting motion.</td><td>ms</td><td>1</td><td>9,999</td></tr>
<tr><td>DEC Time</td><td>Specified acceleration time by msec when stopping motion.</td><td>ms</td><td>1</td><td>9,999</td></tr>
<tr><td colspan="5">
</td></tr>
<tr><td>Wait Time</td><td>Specified waiting time by msec for starting motion of next PT when specifying PT No. for jump/skip.
If JP Table No is specified as blank, this is ignored.</td><td>ms</td><td>0</td><td>60,000</td></tr>
<tr><td colspan="5">

* Even if Wait Time is specified as 0[ms], the system waits for the completion signal of position setting (INP signal) or motor stop signal before starting next Position Table.</td></tr>
</table>

Continue

<table>
<tr><th>Designated item</th><th>Parameter name / Description</th><th>Unit</th><th>Lower limit</th><th>Upper limit</th></tr>
<tr><td>JP Table No.</td><td>When this item specified, the system jump to JP Table No and execute it after completing action of current position. If Position No is specified as 10XXX, system jumps to Position No XXX as soon as "JPT Start" begins, one of this input digital signals from controller to outside, becomes ON.
For program exit, specified as blank.</td><td>–</td><td>0</td><td>127</td></tr>
<tr><td></td><td></td><td></td><td>10,000</td><td>10,127</td></tr>
<tr><td>JPT 0</td><td rowspan="6">If any of these items is checked and there are corresponding input signals of JPT input0, JPT input1 or JPT input2, system jumps to JPT 0, JPT 1 or JPT 2 accordingly regardless of specified "Jump Table No".</td><td></td><td>0</td><td>127</td></tr>
<tr><td></td><td></td><td>10,000</td><td>10,127</td></tr>
<tr><td>JPT 1</td><td></td><td>0</td><td>127</td></tr>
<tr><td></td><td></td><td>10,000</td><td>10,127</td></tr>
<tr><td>JPT 2</td><td></td><td>0</td><td>127</td></tr>
<tr><td></td><td></td><td>10,000</td><td>10,127</td></tr>
<tr><td colspan="5"><table>
<tr><th>Input signal</th><th>Corresponding input jump position</th></tr>
<tr><td>JPT input0</td><td>Input Jump Position No 0</td></tr>
<tr><td>JPT input1</td><td>Input Jump Position No 1</td></tr>
<tr><td>JPT input2</td><td>Input Jump Position No 2</td></tr>
</table></td></tr>
<tr><td>Loop Count</td><td rowspan="3">If these items are specified, system repeats action of the position under specified times (Loop Count) and after then jumps to corresponding position to Loop Jump Table No regardless of specified "Jump Table No".</td><td>–</td><td>0</td><td>100</td></tr>
<tr><td>Loop Jump Table No.</td><td></td><td>0</td><td>127</td></tr>
<tr><td></td><td></td><td>10,000</td><td>10,127</td></tr>
<tr><td>PT set</td><td>Specifies output signals such as PT Output0, PT Output1, PT Output2, in order to confirm the start, pass or end of motor operation for each position.
0, 8, 16: Not use output signal
1~7: Specifies output function when starting operation
9~15: Specifies output function when completing operation
17~23: Specifies output function when the position reach to "Trigger Position"</td><td></td><td>0</td><td>23</td></tr>
<tr><td>Loop Counter Clear</td><td>If this item is checked, Loop Count of specified No of PT is to be cleared.</td><td>–</td><td>0</td><td>127</td></tr>
<tr><td>Check Inpos</td><td>If this item is checked, stop condition is recognized as Inposition finishes.</td><td>–</td><td>0</td><td>1</td></tr>
<tr><td>Trigger Pos</td><td>Specifies position where the PT Output0, PT Output1, PT Output2 signal is ON in case of "PT set" is 17~23.</td><td>pulse</td><td>-2,147,483,648</td><td>+2,147,483,647</td></tr>
<tr><td>Trigger Time</td><td>Specifies pulse width where the PT Output0, PT Output1, PT Output2 signal is ON in case of "PT set" is 17~23.</td><td>ms</td><td>0</td><td>65535</td></tr>
<tr><td>Push Ratio</td><td>Specifies motor torque ratio for push motioning.</td><td>%</td><td>20</td><td>90</td></tr>
<tr><td>Push Speed</td><td>Specifies motion speed of push motioning. (max 200 [rpm])</td><td>pps</td><td>1</td><td>33333</td></tr>
<tr><td>Push Position</td><td>Specifies absolute target position of push motioning.</td><td>pulse</td><td>-2,147,483,648</td><td>+2,147,483,647</td></tr>
<tr><td>Push Mode (Pulse Count)</td><td>Specifies the push mode : Stop mode : 0
Non-stop mode : 1~10,000</td><td></td><td>0</td><td>10,000</td></tr>
</table>