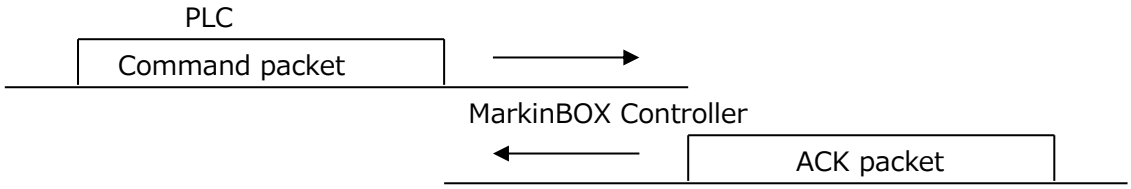
RS232C Serial Communication

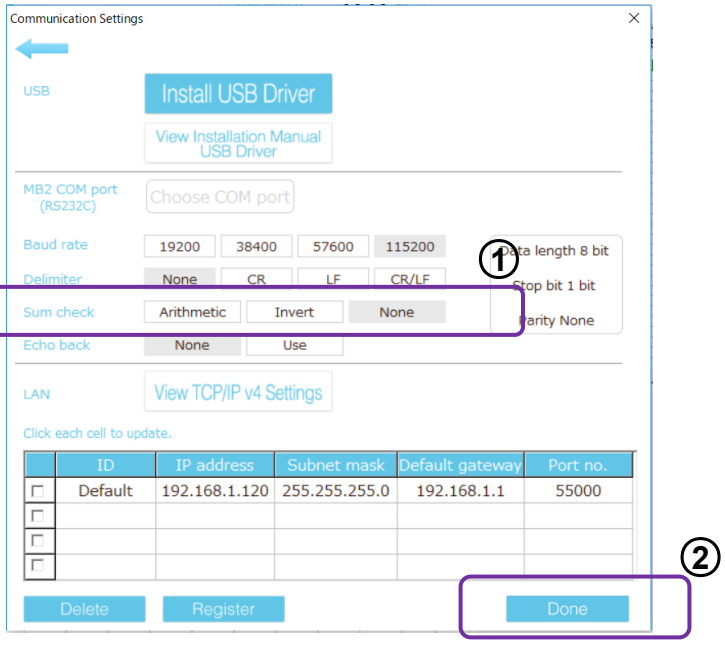
Index

Refer to the following protocol for serial communication specifications.

Serial communication	RS232C（IEE232）
Baud rate	115200 bits/sec
Bit length	8 bit
Parity	Non
Stop Bit	1 bit
No software flow control by Xon/Xoff, etc.	
Transmission data format Commands from PLC are received by MarkinBOX Controller  	

*Figures not satisfying the required digits can be expressed as “0” or “_” (space). However, since the commands are text-based, spaces cannot be used, such as the “0” in “01”.

*Using check sum is always recommended, but if you would like to communicate without check sum, it is also possible to make. Go to “**Preferences**” → **“Communication”** and show the follow window. Select “None” on the Sum Check section, and click the Done button to set up in the controller.



1. ASCII communication with combined file (Simple Communication)

Simple communication is available by sending only mark data combining with file marking. There is two ways to start marking.

【1】 Transfer only mark data by ASCII code then start marking thru ASCII code too .

【2】 Transfer only mark data by ASCII code then start marking thru D.I/O by choosing file

*It's also possible to mark with start button on marking machine after selecting the file in sketchbook 2.

【Common ASCII communication both 1 and 2】

1. Create a text field in sketchbook 2 and save the file into MB2S controller as n (No.001 ~ 255).
2. Communicate with the following protocol

Starting code (2 bytes)	Packet starts @stx fixed		40h 02h (fixed)
Packet no. (2 bytes)	Two characters showing synchronization between the transmission and receiving packets. The two characters arbitrarily controlled by the transmitter. The receiving packet will always contain the same number as the transmitting packet.		Example: 30h 30h
Command (2 bytes)	Commands are fixed 09. *Receiving packet is 10 from MarkinBOX.		30h 39h (fixed)
Data length (3 bytes)	Length of the data (number of bytes). Send the number of bytes (before ETX) of the next packet's data.		Example: 30h 31h 30h
Data	nnn (3 bytes)	File no. (001 ~ 255) Send the file No. saved at the above 1.	Example: 30h 30h 31h
	nn (2 bytes)	Field no. (01 ~ 50) Send the field No. want to change in the file.	Example: 30h 31h
	nn (2 bytes)	Number of characters (01 ~ 50) Number of characters of text to be sent	Example: 30h 33h
	data (Max 50 bytes)	Marking text Maximum: 50 characters	Example : 31h 32h 33h
Ending code (1 byte)	etx fixed.		03h (fixed)
Check sum (2 bytes)	Hexadecimal form in two characters expressing the lower 8-bit figure derived from the sum encompassing from "Packet no." to "Data" above. *@STX and ETX are not included. If you choose "no use checksum", do not need addition.		

<Transmission example : it is no using check sum >

40h 02h 30h 30h 30h 39h 30h 31h 30h 30h 30h 31h 30h 31h 30h 33h 31h 32h 33h 03h

@STX Packet no. Command 09 Data length 10 File no. 001 Field no. 01 Number of characters 3 Text 123 ETX

【1】In case of the start marking thru ASCII communication

Starting code (2 bytes)	Packet starts @stx fixed	40h 02h (fixed)
Packet no. (2 bytes)	Two characters showing synchronization between the transmission and receiving packets. The two characters arbitrarily controlled by the transmitter. The receiving packet will always contain the same number as the transmitting packet.	Example: 30h 30h
Command (2 bytes)	Commands are fixed 11. *Receiving packet is 12 from MarkinBOX.	31h 31h (fixed)
Data length (3 bytes)	Length of the data (number of bytes). 003 is fixed.	30h 30h 33h (fixed)
Execution command (3 bytes)	Send an executed file no. from 001 to 255. It is same as saved file no. the above 1.	Example: 30h 30h 31h
Ending code (1 byte)	etx fixed.	03h
Check sum (2 bytes)	Hexadecimal form in two characters expressing the lower 8-bit figure derived from the sum encompassing from “Packet no.” to “Data” above. *@STX and ETX are not included. If you choose “no use checksum”, do not need addition.	

<Transmission example : it is no using check sum >

40h 02h 30h 30h 31h 31h 30h 30h 33h 30h 30h 31h 03h

@STX Packet no. Command 11 Data length 003 File no. 001 ETX

【2】In case of the start marking thru D.I/O.

1. Choose the mark file No. thru D.I/O.
2. Start marking by D.I/O.

*It is also possible to start marking from the switch button on the marking head.

2. Standard Communication (Full communication)

Packet format

2 bytes	2 bytes	2 bytes	3 bytes	(0-999) bytes	1byte	2 bytes
Starting code	Packet no.	Command	Data length	Data	Ending code	Check sum

*Need from the Stating code(@STX) to the ending code(ETX) only, if you selected no use check sum.

Starting code (2 bytes)	Packet starts @stx fixed			40h 02h (fixed)
Packet no. (2 bytes)	Two characters showing synchronization between the transmission and receiving packets. The two characters arbitrarily controlled by the transmitter. The receiving packet will always contain the same number as the transmitting packet.			Example: 30h 30h
Command (2 bytes)	Commands are fixed at two characters.			
	01	Send marking data to MarkinBOX	PLC → MarkinBOX	30h 31h
	02	Receiving packet: Command 01	MarkinBOX → PLC	30h 32h
	03	Execution command to MarkinBOX	PLC → MarkinBOX	30h 33h
	04	Receiving packet: Command 03	MarkinBOX → PLC	30h 34h
	05	Status request to MarkinBOX	PLC → MarkinBOX	30h 35h
	06	Receiving packet: Command 05	MarkinBOX → PLC	30h 36h
	07	Send move XY data to MarkinBOX	PLC → MarkinBOX	30h 37h
	08	Receiving packet : Command 08	MarkinBOX → PLC	30h 38h
	09	Send text data to MarkinBOX	PLC → MarkinBOX	30h 39h
	10	Receiving packet : Command 09	MarkinBox → PLC	31h 30h
	11	Execution command to MarkinBOX	PLC → MarkinBOX	31h 31h
	12	Receiving packet : Command 11	MarkinBOX → PLC	31h 32h
Data length (3 bytes)	Length of the data (number of bytes). Send the number of bytes (before ETX) of the next packet's data.			Example: 30h 33h 39h
Data (0-999 bytes)	Maximum data is 999, expressed in 999 characters. The data will differ according to the command. Please refer to below.			Refer to below
Ending code (1 byte)	etx fixed.			03h
Check sum (2 bytes)	Hexadecimal form in two characters expressing the lower 8-bit figure derived from the sum encompassing from "Packet no." to "Data" above. *@STX and ETX are not included. If you choose "no use checksum", do not need addition.			

The receiving time is 500 ms maximum. Retry will be controlled by the transmitter.

Data specifications according to command

I. Command 01: Send marking data [PLC → MarkinBOX]

[Header data] and [field data] is necessary.

*Please note that the character height, character pitch, and matrix size are specified up to 99.9mm.
If you need to specify the above to a length of 100 mm or more, please consider creating and sending an marking data file in advance using the following method: "1.ASCII communication with a combined file (Simple Communication)" to call upon that file and use it for marking.

The header data comprises “Force,” “speed,” and “number of fields,” which are common to all fields.

Header data (total bytes: 8)		ASCII examples
nn +0,1 (2 bytes)	Force 01-99	35h 30h (at marking force 50)
nn +2,3 (2 bytes)	Speed 01-99	35h 30h (at marking speed 50)
n +4 (1 byte)	Serial setting 0 (no use serial marking) *Send “0” usually.	30h (no use serial marking)
n +5 (1 byte)	Home position 0 (Back to home after marking) 1 (Pause after marking) *Send “0” usually.	30h (back to home after marking as standard)
nn +6,7 (2 bytes)	Number of fields sent 01 -11	30h 31h (number of fields is 1)

Field data are individual data including marking text.

Field data for text data and logo (maximum 82 bytes)		ASCII examples
nn +0,1 (2 bytes)	Field no. 01-50	30h 31h (field no. is 1)
n +2 (1 byte)	Data format 0 : fixed characters, 3 : logo, 4 : vertical on Y axis, 5 : vertical on X axis, 6 : outer arc, 7 : inner arc	30h (using fixed characters)
n +3 (1 byte)	Marking direction 0 : Standard direction, 2 : Reverse direction	30h (standard direction)
nn.n +4,5,6,7 (4 bytes)	Character height mm	30h 33h 2Eh 30h (at 03.0mm)
nnn +8,9,10 (3 bytes)	Character width percentage %	30h 36h 30h (at 060%)

nnnn +11,12,13,14 (4 bytes)	Angle Degree	2Dh 30h 34h 35h (at -045 deg)
nn.n +15,16,17,18 (4 bytes)	Character pitch mm	30h 32h 2Eh 35h (at 02.5mm)
nn.n or nnnn +19,20,21,22 (4 bytes)	Starting position X mm	30h 31h 2Eh 30h (at 01.0mm) or 31h 35h 30h 30h (at 150.0mm)
nn.n or nnnn +23,24,25,26 (4 bytes)	Starting position Y mm	30h 33h 2Eh 30h (at 03.0mm) or 31h 30h 30h 35h (at 100.5mm)
nn +27,28 (2 bytes)	Number of characters (bytes) Maximum: 50 bytes	30h 35h (at 5 bytes)
data +29-78 (Maximum:50 bytes)	Marking data Maximum: 50 characters *When logo marking, logo numbers will be transmitted registered in the controller. @L[..] should be added in the front and back like 01 ~ 31.	41h 42h 43h 44h 45h (ABCDE) 40h 4Ch 5Bh 30h 31h 5Dh (logo no. is 1)
nnn + (3 bytes)	Radius of Arc Not necessary input except ARC: mm	30h 31h 30h (010mm)

Field data for 2D code (maximum: 79 bytes)		ASCII examples
nn +0,1 (1 byte)	Field no. 01-50	30h 31h (field no. is 1)
n +2 (1 byte)	Data format 8 : fixed characters	38h (fixed characters)
n +3 (1 byte)	Barcode types 1:QR; 2:Date matrix	31h (QR code)
nn +4,5 (2 bytes)	Force 01-99	33h 30h (at marking force 30)
nn +6,7 (2 bytes)	Barcode marking speed 01-99	32h 30h (at marking speed 20)
nn +8,9 (2 bytes)	Dimensions: Data matrix only (QR:00) (10,12,14,16,18,20,22,24,26,32,36,40)	30h 30h (at 00 when QR code)
n +10 (1 byte)	Direction p : Two-way; q : One-way	70h (Two-way)

nnnn +11,12,13,14 (4 bytes)	Angle deg	30h 30h 30h 30h (at 0000 deg)
nn.n +15,16,17,18 (4 bytes)	Matrix size mm	30h 35h 2Eh 30h (at 05.0mm)
nn.n or nnnn +19,20,21,22 (4 bytes)	Starting position X mm	30h 30h 2Eh 31h (at 00.1mm) or 31h 35h 30h 30h (at 150.0mm)
nn.n or nnnn +23,24,25,26 (4 bytes)	Starting position Y mm	30h 35h 2Eh 35h (at 05.5mm) or 31h 30h 30h 35h (at 100.5mm)
nn +27,28 (2 bytes)	Number of characters (bytes) Maximum: 50 bytes	30h 35h (at 5 bytes)
data +29-78 (Maximum:50 bytes)	Data Maximum: 50 characters	41h 42h 43h 44h 45h (ABCDE)

II. Command 03: Execution command to MarkinBOX [PLC → MarkinBOX controller]

Execution command		ASCII example
n +0 (1 byte)	1: Start marking 2: Pause 3: Stop 4: Alarm reset 5: Return to origin	31h (when marking started)

Comm. example : 40h 02h 32h 32h 30h 33h 30h 30h 31h 31h 03h SS

@STX Packet no. Command no.03 Data length Start marking ETX Check sum SS

*Need from the Stating code(@STX) to the ending code(ETX) only, if you selected no use check sum.

III. Command 05: Status request to MarkinBOX [PLC → MarkinBOX controller]

Status request		ASCII example
0 bytes	No data	None

Comm. example : 40h 02h 33h 33h 30h 35h 30h 30h 30h 03h SS

@STX Packet no. Command no.05 Data length ETX Check sum SS

*Need from the Stating code(@STX) to the ending code(ETX) only, if you selected no use check sum.

IV. Command 07: Move XY position to MarkinBOX [PLC → MarkinBOX controller]

Move XY position		ASCII example
nn +0,1, (2 bytes)	Motion speed 01-10 (general setting 00)	30h 30h (general setting)
nn.n or nnnn +2,3,4,5 (4 bytes)	X coordinate X mm	30h 35h 2Eh 30h (at 05.0mm) or 31h 35h 30h 30h (at 150.0mm)
nn,n or nnnn +6,7,8,9 (4 bytes)	Y coordinate Y mm	31h 30h 2Eh 30h (at 10.0mm) or 31h 30h 30h 35h (at 100.5mm)

Comm. example : 40h 02h 34h 34h 30h 37h 30h 31h 30h 30h 30h 30h 35h 2Eh 30h 31h 30h 2Eh 30h 03h SS

@STX Packet no. Command no. Data length Motion speed X coordinate Y coordinate ETX Check sum SS

*Need from the Stating code(@STX) to the ending code(ETX) only, if you selected no use check sum.

V. Command 02, 04, 08, 10, 12 receiving packet [PLC ← MarkinBOX controller]

Receiving packet : ACK		ASCII example
n +0 (1 byte)	ACK	06h

Comm. example : 40h 02h 31h 31h 30h 32h 20h 20h 31h 06h 03h SS

@STX Packet no. Command no. Data length ACK ETX Check sum SS

*Need from the Stating code(@STX) to the ending code(ETX) only, if you selected no use check sum.

Receiving packet : NACK ASCII example		ASCII example
n +0 (1 byte)	NACK	15h

Comm. example : 40h 02h 31h 31h 30h 32h 20h 20h 33h 15h N..N 03h SS

@STX Packet no. Command no. Data length NACK ETX Check sum SS

*Need from the Stating code(@STX) to the ending code(ETX) only, if you selected no use check sum.

In case of NACK, the following N..N will be included after 15h.

In response to;	**Command (N..N)**
General command	01 : Bad command, 02 : Abnormal data size, 03 : Error in etx position, 4ssSS : Check sum error (ss: correct check sum; SS: received check sum)
start of marking	32 : Alarming, 33 : In operation and cannot execute, 34 : No marking data

stop command	35 : Not in operation, or halting
return to origin command	36 : Returning to origin
move XY position	51 : Alarming, 52 : In operation, 54 : Abnormal motion speed parameter
file relations command	81 : Abnormal file no., 82 : Abnormal field no., 83 : Abnormal text size
file marking command	61 : No existing file, 62 : Abnormal file reading
Others	30 : Abnormal data format, 31 : Bad command number

VI. Command 06: Response to status request (MarkinBOX to PLC)

Receiving status request		ASCII example
nn +0,1 (2 bytes)	'99':Alarming ' 0':Standby ' 1':Marking ' 2':Halting ' 3':Returning to origin ' 5':Operating : Other	30h 31h (Marking)

Comm. example : 40h 02h 33h 33h 30h 36h 20h 20h 32h 20h 33h 03h SS

@STX Packet no. Command no. Data length Returning to origin ETX Check sum SS

*Need from the Stating code(@STX) to the ending code(ETX) only, if you selected no use check sum.

<Transmission example : it is no using check sum >

40h 02h 30h 31h 30h 31h 30h 37h 36h 35h 30h 35h 30h 30h 30h 32h 30h 31h

@STX Packet no. Command 01 Data length 76 Force 50 Speed 50 No Serial Home position Number of fields 2 Field No.1

30h 30h 30h 33h 2Eh 30h 30h 36h 30h 30h 30h 30h 30h 32h 2Eh 35h 30h 30h 2Eh 31h

Format 0 standard 0 Chara.height 3mm Chara.width 60% Angle 0 Chara.pitch 2.5mm X axis 0.1mm

30h 33h 2Eh 35h 30h 35h 41h 42h 43h 44h 45h 30h 32h 30h 30h 30h 33h 2Eh 30h

Y axis 3.5mm Number of chara.5 bytes Text ABCDE Field No.2 Format 0 standard 0 Chara.height 3mm

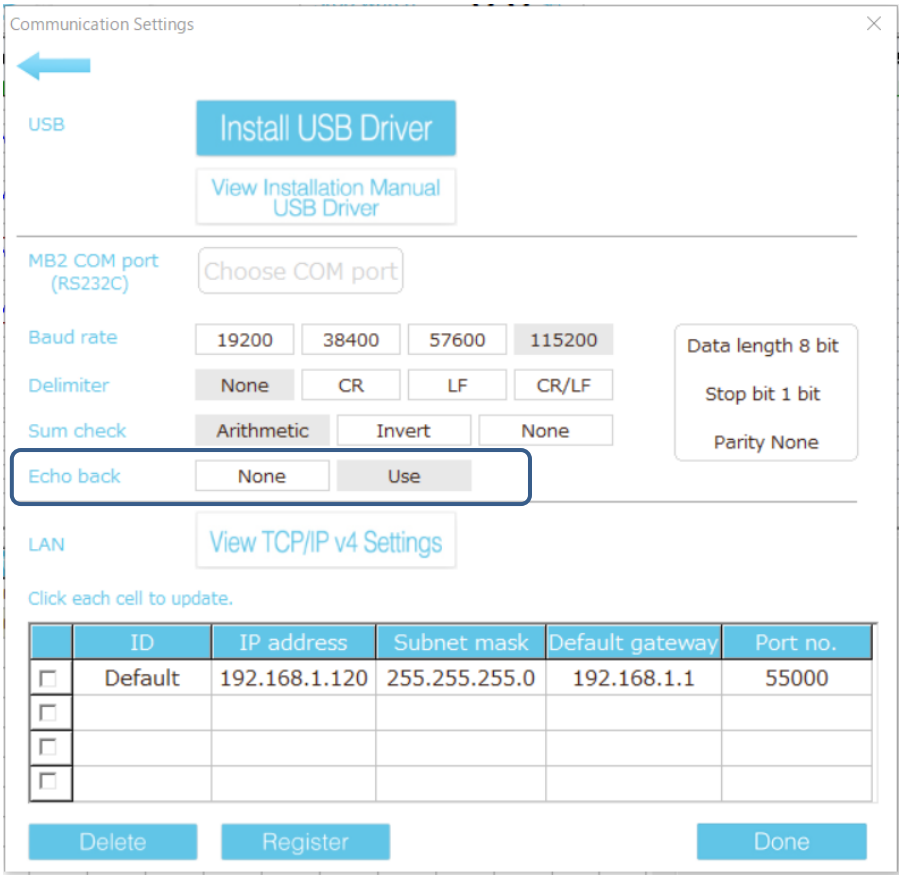
30h 36h 30h 30h 30h 30h 30h 32h 2Eh 35h 30h 30h 2Eh 31h 30h 37h 2Eh 30h 30h 35h 30h 30h 30h 30h 31h 03h

Chara.width 60% Angle 0 Chara.pitch 2.5mm X axis 0.1mm Y axis 7mm Number of chara.5 bytes Text 00001 ETX

*Need from the Stating code(@STX) to the ending code(ETX) only, if you selected no use check sum.

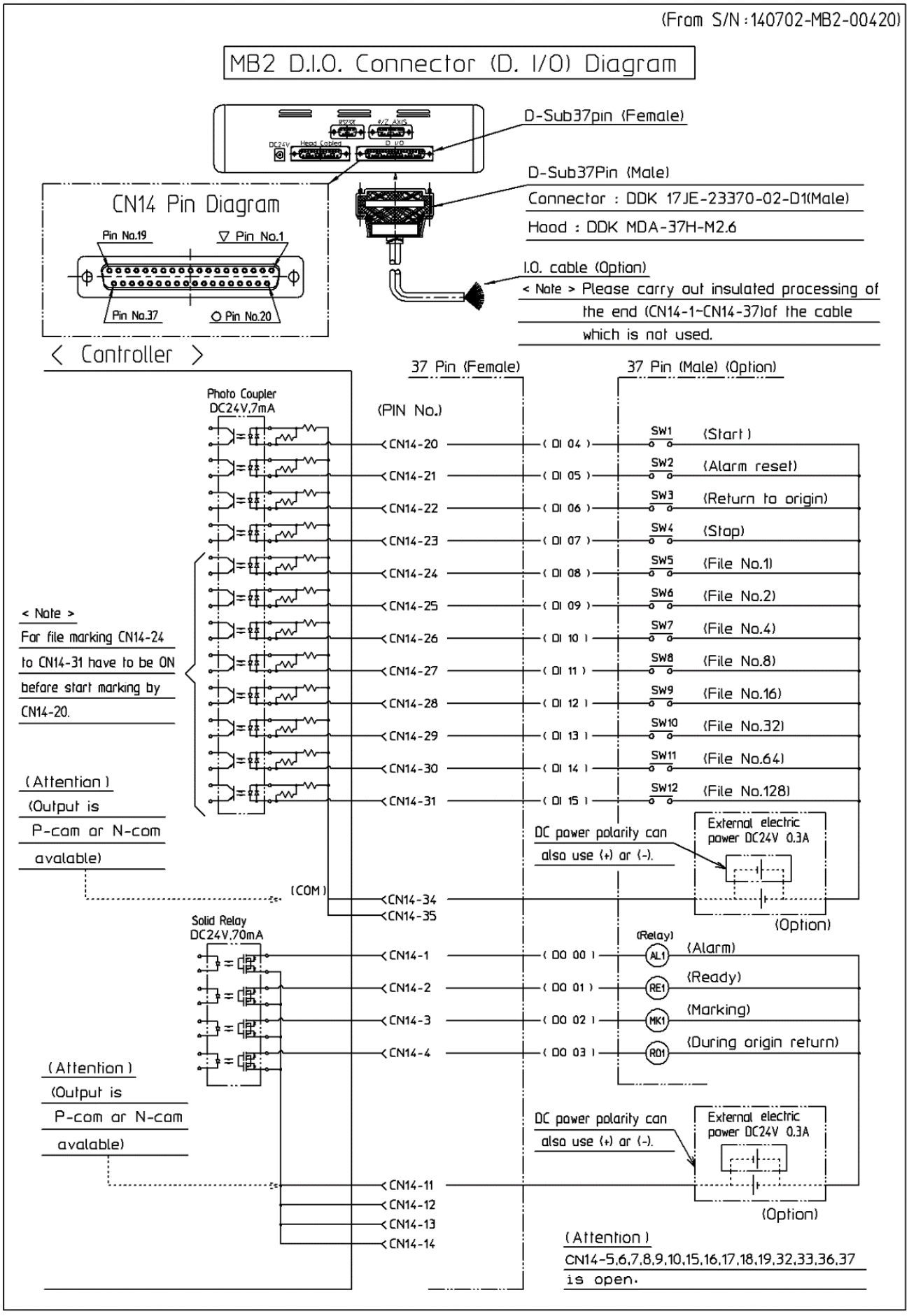
3. Echo Back function

Communication MB2 and PLC via RS232C serial is selectable the echo back function, which means MB2 enable to respond the command back to PLC. When MB2 receives @stx command, MB2 sends the echo command back. Enable to setup in the **"Preferences" → "Communication"**.



PLC (send)	MB2	PLC (receive)
Command 01	→	Command 01 and ACK/NACK
Command 03		Command 03 and ACK/NACK
Command 05		Command 05 and response status
Command 09		Command 09 and ACK/NACK
Command 11		Command 11 and ACK/NACK

4. Connectivity Diagram



Wiring Drawing For D-Sub 9 pin Connector (RS232C)

D-Sub 9pin Connector (Female) — To MB Connector

Pin No.	Signal Name	Content
1		
2	TXD	Transmit Data
3	RXD	Receive Data
4	DSR	Data Set Ready
5	GND	Ground
6	DTR	Data Terminal Ready
7	CTS	Clear to Send
8	RTS	Request to Send
9		

D-Sub 9pin Connector (Male) — To PC

Pin No.	Signal Name	Content
1		
2	RXD	Receive Data
3	TXD	Transmit Data
4	DTR	Data Terminal Ready
5	GND	Ground
6	DSR	Data Set Ready
7	RTS	Request to Send
8	CTS	Clear to Send
9