

Manual



FlexiBowl®

**SCHNEIDER MODICON M340
FLEXIBOWL PLUGIN**

TABLE OF CONTENTS

1. **Introductory Notes**
2. **Control Expert (Unity Pro)**
3. **Commands list**
4. **Alarms table**

This Plugin was created with the idea of communicating **quickly and safely with FlexiVision** through the **Schneider Modicon M340 series PLC** , using **LD/ST** language instructions.

The plug-in developed in **Control Expert (Unity Pro)** does not require an additional licence and requires the use of the **BMX NOC0401.2** card for Ethernet/IP.

FlexiBowl® Plug-In



Introductory Notes

Introduction.

The communication block with the FlexiBowl through the Schneider M340 PLC is modelled on the communication blocks of Omron and Siemens PLCs.

The aforementioned PLCs provide in their libraries the instructions for directly accessing the socket of their Ethernet ports and thus allow them to completely manage the data flow; thanks to this it was possible to set up customised communication between the PLCs and the FlexiBowl.

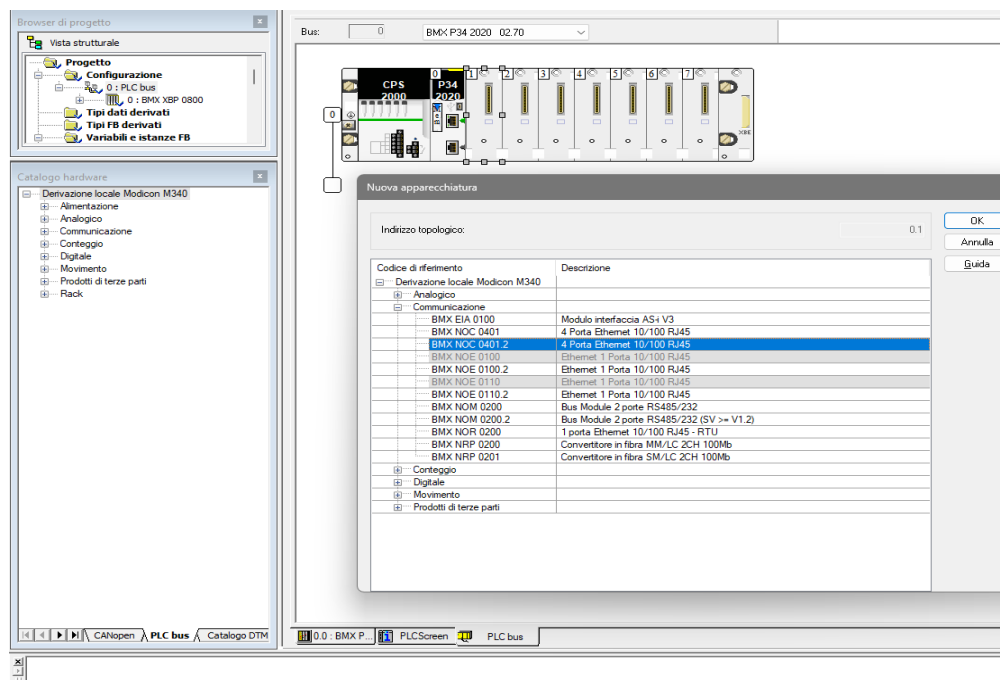
In the case of **Schneider's Modicon M340 and M580 series PLCs (Process)**, which over time replaced the Premium series, the possibility of being able to access the PLCs' Ethernet socket directly was removed, so that in order to realise a block implementing the same control logic as the FlexiBowl, it was necessary to use the industrial Ethernet/IP protocol made available by the FlexiBowl servo drive itself.

While an Ethernet card (**BMX NOC401.2**) must be installed on the PLC in order to communicate in Ethernet/IP.

In this case Ethernet/IP communication, apart from being communication at a higher level than TCP and UDP protocols, has a very precise set of instructions defined by the manufacturer of the servo drive that mounts the FlexiBowl.

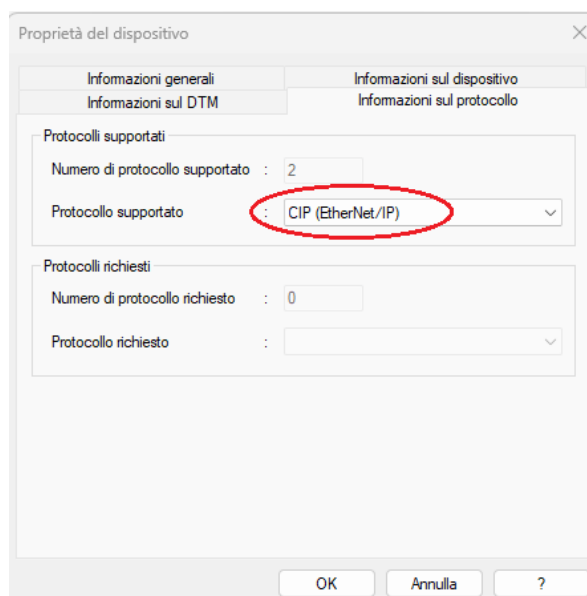
The set of instructions available with this communication block is provided from page 10 onwards of this manual, and in order to be able to send a correct command sequence, as also mentioned on page 10 of this manual, it is recommended to refer to the manual "**MOONS Host Command Reference REV J of 2014**".

Below are the steps necessary for importing and configuring the communication and control block with the FlexiBowl for the first time.



Step 1.

First of all, the **Ethernet/IP BMX NOC0401.2 Ethernet/IP** card must be inserted into the PLC hardware configuration.

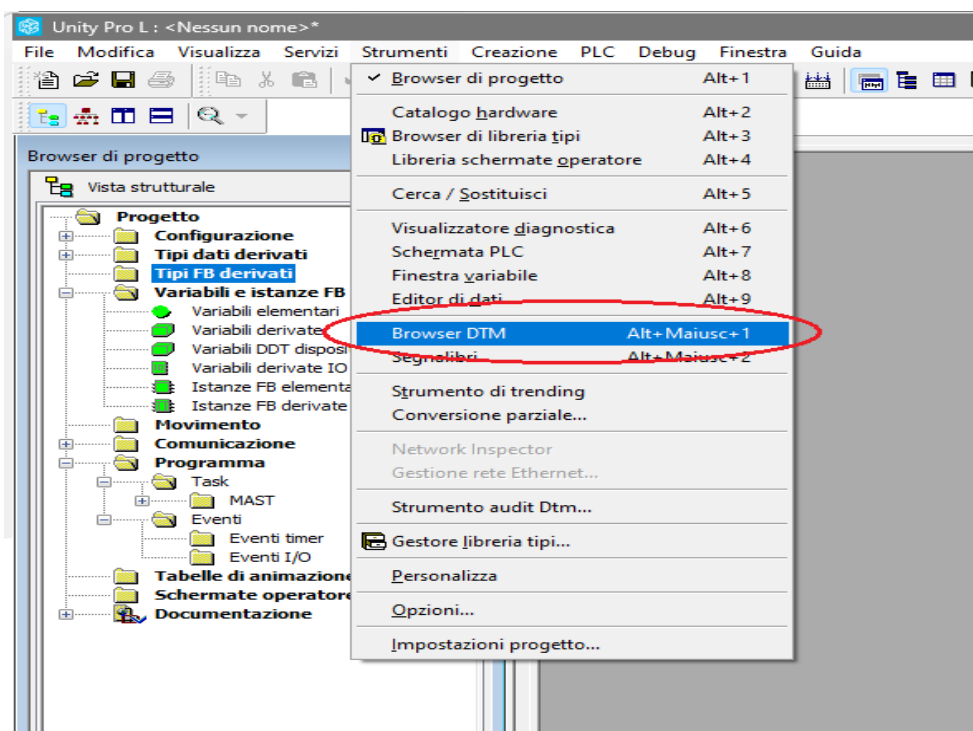


Step 2.

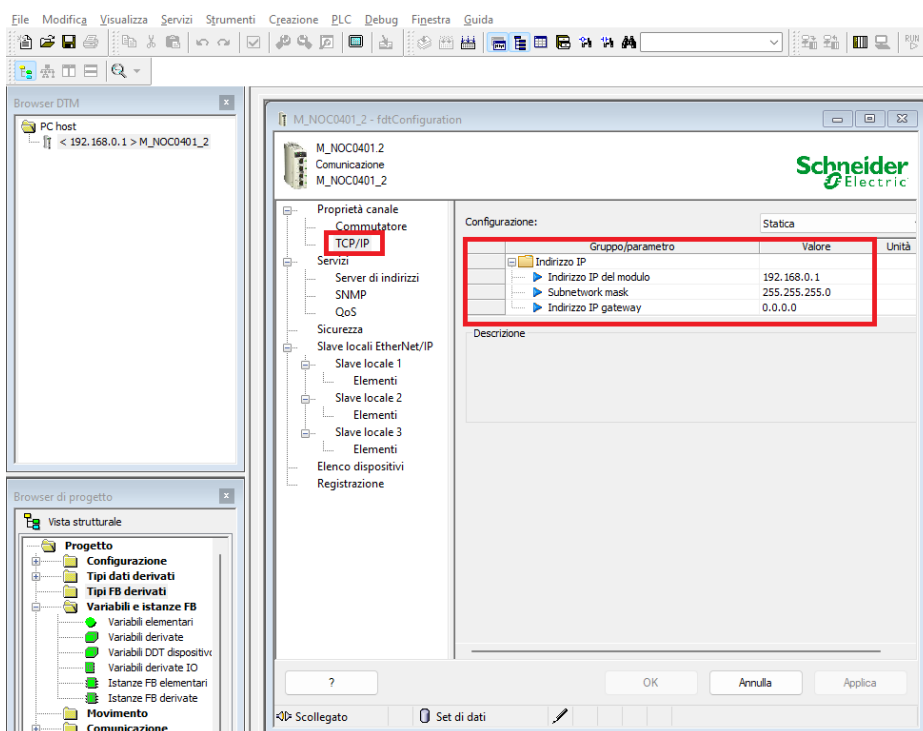
Before confirming the insertion of the card, ensure that the '**CIP Ethernet/IP**' protocol is selected, then click **OK**.

Then exit the PLC hardware configuration and save the changes.

Control Expert (Unity Pro)

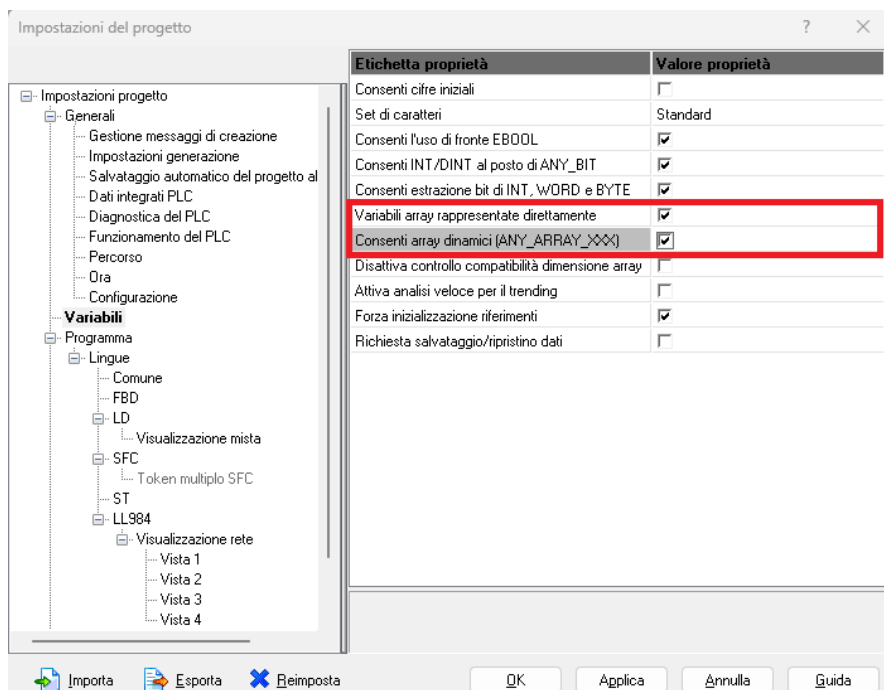


Step 3. Now open the **DTM** management section



Step 4. Go to the IP address setting section of the card, and set the IP address, subnet mask and gateway, making sure they are in the same subnet as the FlexiBowl, or vice versa.

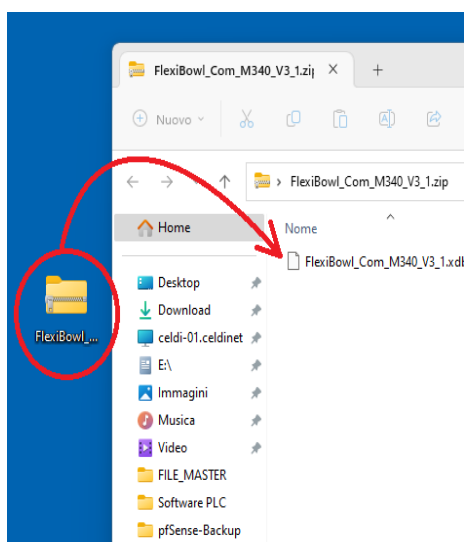
Once configuration is complete, click **OK**.



Step 5.

Before continuing, check that the following items have been selected in the Control Expert project settings:

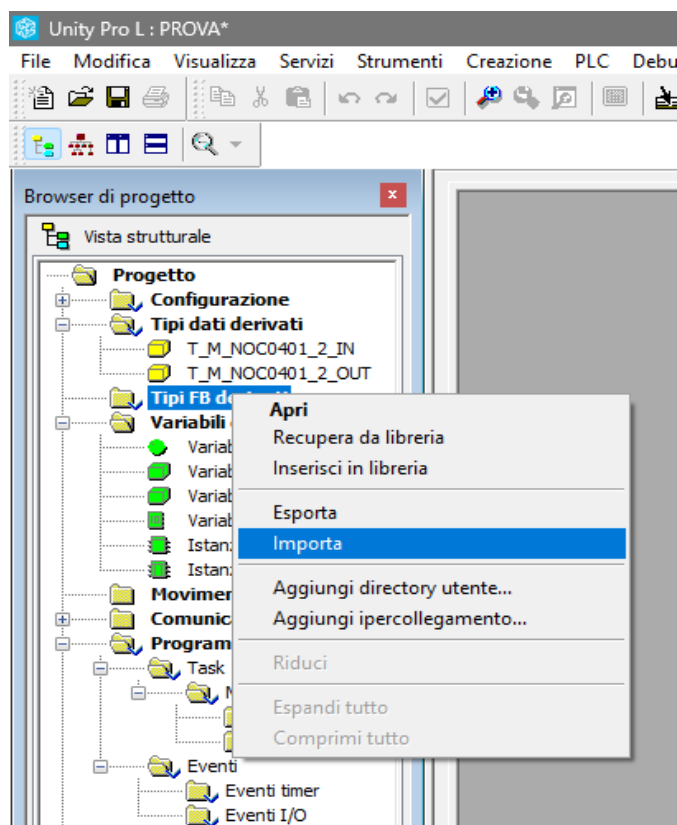
- Directly represented **array variables**;
- **Allow dynamic arrays (ANY_ARRAY_XXX)**;



Step 6.

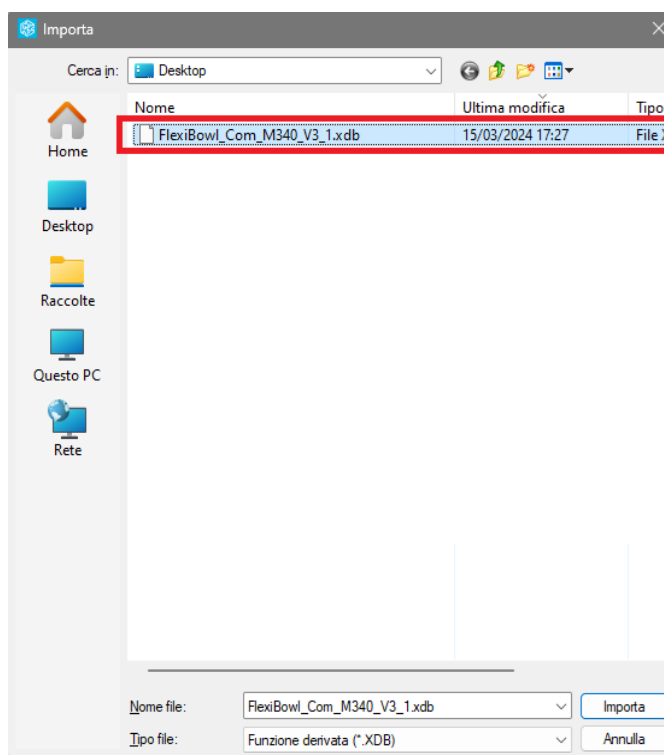
Unzip the folder
"FlexiBowl_Com_M340_V3_1.zip"
on your desktop so that you can
access the file
"FlexiBowl_Com_M340_V3_1.xdb"
from Control Expert

Control Expert (Unity Pro)



Step 7.

Right-click on the '**FB Derived Types**' folder, then select Import

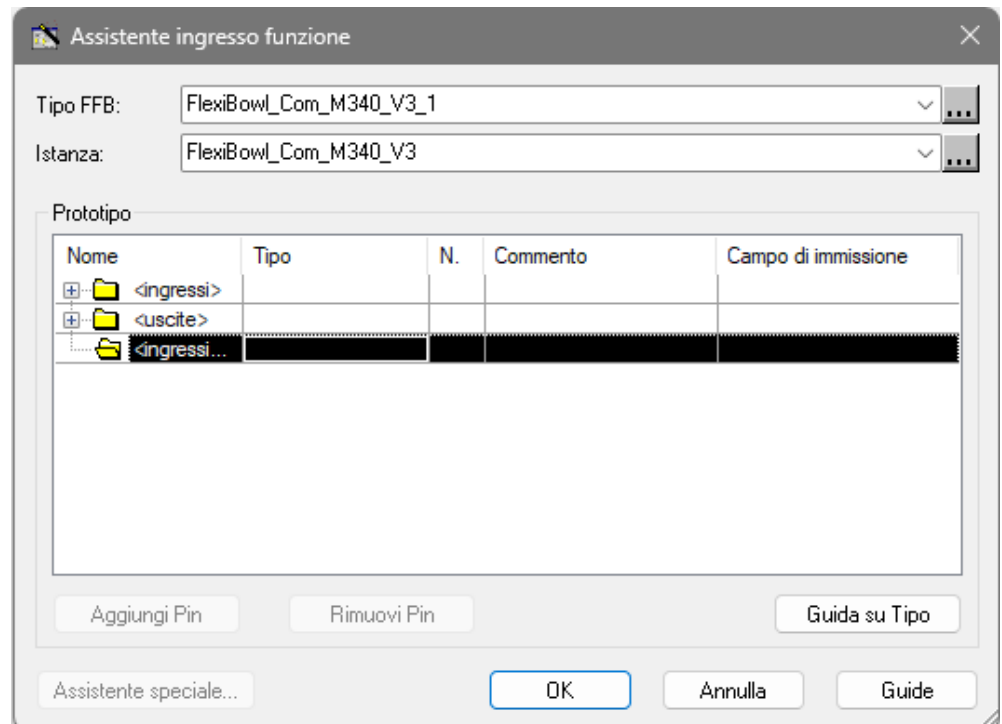


Step 8.

Then select the file "**FlexiBowl_Com_M340_V3_1.xdb**" previously unzipped on the Desktop. Press **OK** to confirm file selection.

Control Expert (Unity Pro)

Step 9.



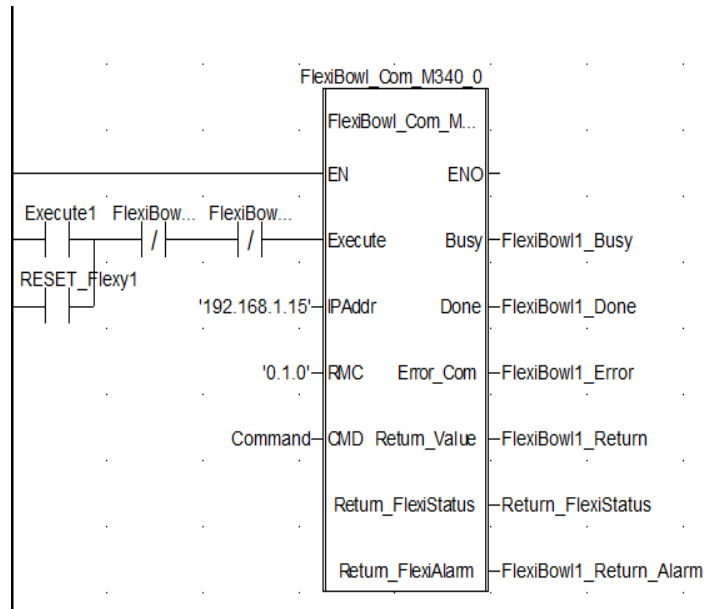
Once the block is imported, it will be available from the assistant for input into the development environment, then it can be imported.

N.B: ONLY ONE BLOCK CAN BE USED PER FLEXIBOWL CONNECTED TO THE PLC, MANAGE THE INSTRUCTIONS TO BE SENT AS PER THE EXAMPLE SOFTWARE.

Control Expert (Unity Pro)

Step 10.

Now fill the block inputs and outputs according to your needs:



EN	IN: Bool	Enabling the block. It must always be enabled because there are instructions therein that maintain the Ethernet/IP connection active
Execute	IN: Bool	Send this command to the "CMD" input. The block only considers this input for one PLC scan cycle and if communication is established.
IPAddr	IN: String	IP address of the Flexibowl. Specify the IP address of the Flexibowl. The communication instructions are high level and managed by the PLC, therefore even if the FlexiBowl is in another subnet and there is a reference gateway set in the hardware configuration of the PLC, the final FlexiBowl IP address can be specified directly.
RMC	IN: String	The RMC variable indicates the physical address of the Ethernet port to be used within the PLC. In the example project, the BMX NOC 0401.2 card is installed on the CPU side, both are located in the main rack (rack 0), from this information it can be deduced that the RMC address will be '0.1.0'. The port address is indicated by the last digit, i.e. C (Connector) in the case of BMX NOC 0401.2 it will always be 0, as all the Ethernet ports installed on board are in switch mode.

Control Expert (Unity Pro)

TAG	Type	Description
ENO	OUT: Bool	Successful block execution without any software error
Busy	OUT: Bool	Execution in progress by the FlexiBowl, therefore, no other command can be sent.
Done	OUT: Bool	Last successful command execution. For every communication error, this bit is set to FALSE even if a command is not being executed. Otherwise it always remains set to TRUE following a successfully completed instruction until the next request.
Error_Com	OUT: Bool	This bit is set to TRUE for every communication error and reset every time communication is restored.
Return_Value	OUT: String	Return string from the FlexiBowl. Return values from Ethernet/IP, for full interpretation see 'MOONS Host Command Reference REV J of 2014' manual If executing a QX, the 'QXRunning' value will be returned until completion. Once the QX instruction has been completed, the value of the last status request to the FlexiBowl will be found until the next execution. In the case of any movement required of the FlexiBowl outside QX, e.g. 'FL' command will be returned with the 'Mooving' value. In the event of a communication error, the 'Communication Error. Check the flexiBowl Con' value will be returned
Return_FlexiStatus	OUT: Word	Word containing the result of the 'SC' instruction, which is constantly sent to the FlexiBowl during any movement or every 20 sec. in idle state, i.e. if no FlexiBowl is requested by the block. The example project shows how to interpret this word to know whether the FlexiBowl is moving or performing a QX. For a complete interpretation refer to the manual 'MOONS Host Command Reference REV J of 2014'
Return_FlexiAlarm	OUT: Word	Word containing any error code read by the FlexiBowl driver. See alarms table

Commands list

Before going on to explain the list of commands, it is important to know that with the Ethernet/IP protocol, it is not possible to send 100% of the commands in the **"MOONS Host Command Reference REV J of 2014"** manual, as operation with Ethernet/IP involves a reinterpretation of the commands converted into hexadecimal bytes, specifically refer to the above-mentioned manual to be able to understand in detail, at a low level, how this protocol works.

Commands	Description
MOTION COMMANDS	
AC	P to P Acceleration
AM	Max Acceleration
AX	Alarm Reset
CJ	Start Jogging
DC	Set Change Distance
DE	P to P Deceleration
DI	Distance or Position
EF	Encoder Function
EG	Electronic Gearing
EP	Encoder Position
FC	P to P Change
FD	Feed to Double Sensor
FE	Follow Encoder
FM	Feed to Sensor with Mask Dist
FO	Feed to Length & Set Output
FP	Feed to Position
FS	Feed to Sensor
FY	Feed to Sensor with Safety Dist
HW	Hand Wheel
JA	Jog Accel/Decel rate
JD	Jog Disable
JE	Jog Enable
JL	Jog Decel rate
JS	Jog Speed
MD	Motor Disable
ME	Motor Enable
MT	Multi Tasking
SH	Seek Home
SM	Stop the Move
SP	Set Absolute Position

Commands list

Commands	Description
MOTION COMMANDS	
VC	Velocity for Speed Change (FC)
VE	Velocity Setting (For Feed Commands)
WI	Wait For Input
WM	Wait on Move
WP	Wait on Position
CONFIGURATION COMMANDS	
AD	Analog Deadband
AS	Analog Scaling
BD	Brake Disengage Delay time
BE	Brake Engage Delay time
CC	Running Current
CD	Idle Current Delay
CI	Change Idle Current
CM	Control mode
ER	Encoder or Resolution
FI	Filter Input
FX	Filter Selected Inputs
HG	Harmonic Smoothing Gain
HP	Harmonic Filter Phase
PF	Position Fault
PM	Operation Mode
SF	Step Filter Frequency
I/O COMMANDS	
AF	Analog Filter
AG	Analog Velocity Gain
AI	Alarm Input usage
AO	Alarm Output usage
AP	Analog Position Gain
AT	Analog Threshold
AV	Analog Offset
AZ	Analog Zero (Auto Zero)
BO	Brake Output usage
DL	Define Limits
MO	Motion Output
OI	On Input
SI	Enable Input usage
SO	Set Output
TI	Test Input
REGISTER COMMANDS	
CR	Compare Register
R+	Register Addition
R-	Register Subtraction
R*	Register Multiplication
R/	Register Division
R&	Register Logical AND

Commands	Description
REGISTER COMMANDS	
R	Register Logical OR
RC	Register Counter
RD	Register Decrement
RI	Register Increment
RM	Register Move
RR	Register Read
RW	Register Write
RX	Register Load
TR	Test Register
TS	Time Stamp read
Q PROGRAM COMMANDS	
QC	Queue Call
QG	Queue Goto
QJ	Queue Jump
QR	Queue Repeat
QX	Queue Load & Execute
WD	Wait Delay Register
WT	Wait Time

Commands	Description
QX2	Move
QX3	Move-Flip
QX4	Move-Flip-Blow
QX5	Move-Blow
QX6	Shake
QX7	Light on
QX8	Light off
QX9	Flip
QX10	Blow
QX11	Quick Emptying Option
QX12	Reset Alarm

Alarm list

Hexadecimal value	Description
16#0001	Position limit
16#0002	CCW Limit
16#0004	CW Limit
16#0008	Over Temp
16#0010	Excess Regen
16#0020	Over Voltage
16#0040	Under Voltage
16#0080	Over Current
16#0100	Open Motor Winding
16#0200	Bad Encoder
16#0400	Comm Error
16#0800	Bad Flash
16#1000	No Move
16#2000	Motor Resistance Out of Range
16#4000	Blank Q Segment
16#8000	(Not Used)